



# Player Population Patterns in Digital Games: A Data Analytics and Machine Learning Approach

**Author:**

Wannigamage, Dulakshi

**Publication Date:**

2021

**DOI:**

<https://doi.org/10.26190/unsworks/22413>

**License:**

<https://creativecommons.org/licenses/by-nc-nd/3.0/au/>

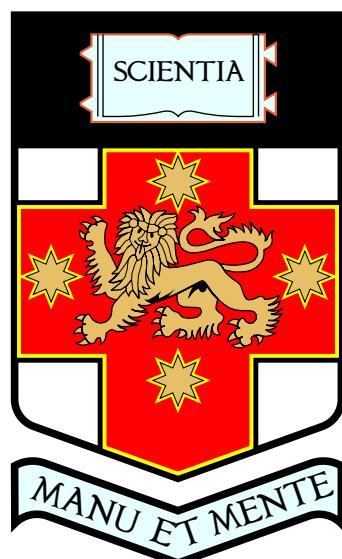
Link to license to see what you are allowed to do with this resource.

Downloaded from <http://hdl.handle.net/1959.4/70718> in <https://unsworks.unsw.edu.au> on 2022-08-04

# Player Population Patterns in Digital Games: A Data Analytics and Machine Learning Approach

Dulakshi Vihanga Wannigamage

A thesis submitted in fulfilment  
of the requirements of the degree of  
Doctor of Philosophy



School of Engineering and Information Technology  
University of New South Wales, Canberra  
Australia  
March 2021



# Thesis Title and Abstract

## Thesis Title

Player Population Patterns in Digital Games: A Data Analytics and Machine Learning Approach

## Thesis Abstract

Game data analytics have gained increased attention in the digital games industry to comprehend player behaviour. However, there is a limited understanding of game player populations across games, especially, with respect to the player population changes in the presence of various external factors. This thesis investigates the fluctuations of game player populations that occur in the presence of various external factors to generate insights and predictions about population fluctuation patterns. Specifically, temporal factors, namely, time of the day, day of the week, time since game release and events, namely, sale events and pandemics are considered. Player population data for around 2000 popular games on the Steam platform collected over two years and eight months in five minutes and hourly frequency are used in the study. First, the thesis investigates the existence of short-term seasonality in population fluctuations of games. Analysing the population time series revealed that the majority of games display daily and weekly population patterns. Moreover, nine archetypes of weekly player population patterns were identified by a dynamic time warping based clustering approach. Secondly, the thesis focuses on the population changes occurring since the release of a game to identify game life cycle patterns. To this end, a piecewise linear trend extraction algorithm capable of extracting the life cycle shape of a game is introduced. Four archetypal life cycle shapes based on the player population fluctuations were identified by a clustering approach. Next, considering the population fluctuations occurring during sale events of games, the thesis introduces an approach based on artificial neural network to forecast the maximum player population of a game during a sale event. The model uses the past sale event history of all games and game life cycle shapes. Finally, the thesis focuses on gameplay patterns during the onset of the COVID-19 crisis. Insights about the player population changes that occurred in games during the pandemic were generated. Moreover, investigations were conducted to determine how well the game preferences during the pandemic can be predicted through classification models. In essence, the thesis provides an understanding of player population fluctuations of games in the presence of a variety of external factors.



# Declarations

## ORIGINALITY STATEMENT

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

## COPYRIGHT STATEMENT

I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

## AUTHENTICITY STATEMENT

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.



# Inclusion of Publications Statement

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

- The candidate contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. they were responsible primarily for the planning, execution and preparation of the work for publication.
- The candidate has obtained approval to include the publication in their thesis in lieu of a Chapter from their Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement.**

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

Parts of the work reported in Chapter 4 have been published in 'Dulakshi Vihanga, Michael Barlow, Erandi Lakshika and Kathryn Kasmarik, "Weekly Seasonal Player Population Patterns in Online Games: A Time Series Clustering Approach," 2019 in IEEE Conference on Games (CoG), London, United Kingdom, 2019, pp. 1-8'.

Parts of the work reported in Chapter 7 have been published in 'Dulakshi Wannigamage, Michael Barlow, Erandi Lakshika and Kathryn Kasmarik, "Analysis and Prediction of Player Population Changes in Digital Games during the COVID-19 Pandemic," 2020 in Australasian Joint Conference on Artificial Intelligence, Canberra, Australia, 2020, Springer, Cham.

These research papers are mentioned at the beginning of each corresponding chapter.

## Candidate's Declaration

I declare that I have complied with the Thesis Examination Procedure.



# Acknowledgement

There are numerous people who have immensely supported me throughout my PhD journey without whom this thesis would not have been possible. First and foremost my deep appreciation goes to my supervisor, A/Prof. Michael (Spike) Barlow for his continuous guidance and motivation and being a great mentor. Thank you for the insightful discussions, for being understanding, for sharing valuable career and life lessons, for showing new perspectives. Thank you for making my PhD journey an enjoyable and memorable one. Next, I am grateful for my co-supervisor, Dr. Erandi Lakshika because of whom I got to know about this valuable PhD opportunity. Thank you for your valuable feedback, ideas and encouragements and providing me support throughout this journey. Also, I greatly admire my co-supervisor, A/Prof.Kathryn Kasmarik for her valuable advices, constructive suggestions, guidance and motivation provided to support me in achieving my goals.

I wish to acknowledge my university UNSW Canberra, for providing me a scholarship to continue my studies without financial hardships and for the various events and opportunities provided to PhD students. Also, I am thankful for my annual review panel members for their support. Next, I want to thank all my officemates in room 112 and all members of Virtual Environments and Simulations Lab for being great companions making my journey an enjoyable one.

Next, my thanks go to all the lecturers at the University of Colombo School of Computing for helping me to take my first step in computer science. Also, I am sincerely grateful for all the teachers I met throughout my life. You all made a huge impact on me and I am grateful to you for shaping me to reach new heights.

Most importantly, I am sincerely thankful for my family for being kind, loving and supportive pillars throughout my life. I am grateful for my beloved parents for believing in me and for supporting me in every possible way to achieve my dreams. I am thankful for my dearest husband Dilip for his immense love, support and encouragement throughout this journey; without your support this journey would have been very difficult.

Thank you all for this wonderful ride.



# Abstract

Digital games industry has become a leading sector in the entertainment market generating multi billions in revenue. Within the games industry, game data analytics have gained an increased attention due to the immense support it provides in decision making. A key motive of game analytics is to understand player behaviour; players are the most important asset that drives the game industry. While much work has been done in this area, there are still some aspects overlooked due to a lack of access to game data. As such, most studies are limited to single or a few games; limiting the understanding about game player populations spread across multiple games. Further to that, the changes of player population size of various games occurring in the presence of various external factors are also understudied.

Thus, this thesis investigates the fluctuations of game player populations that occurred in the presence of various external factors to generate insights and predictions about population fluctuation patterns. Specifically, temporal factors, namely, time of the day, day of the week, time since a game is released and events, namely, sale events and pandemics are considered. The Steam game platform is used as the data source for the thesis and player population time series of 2000 games collected over 2 years and 8 months are used as the main data set.

This thesis first investigates short term seasonality in player population fluctuations. It is conducted with the aim of identifying daily and weekly recurring patterns in player population fluctuations of games. It is performed by utilizing Dynamic Time Warping, time series seasonality detection and clustering techniques. Furthermore, longitudinal player population fluctuations of games are investigated to understand how population change throughout the lifetime of a game. It is conducted through a clustering approach while introducing a piecewise linear trend extraction method for life stage detection. Moreover, this thesis explores how accurately player population during sale events of games can be predicted and proposes an artificial neural network approach that utilizes past population and sale event information of games to predict maximum population during sale events. Also, player population fluctuations during the onset of the COVID-19 pandemic are scrutinized to perceive the player population changes occurring during such pandemics. Moreover, the thesis investigates how well game preferences during the pandemic, which

is a rare world event, can be predicted through a classification approach.

Results indicate the existence of short term seasonality in player population fluctuations and nine archetypal weekly player population changing patterns. Results also reveal the existence of four different life cycle profiles with respect to long term population changes. Furthermore, it was identified that population during sale events of games can be predicted utilizing machine learning approaches using past population and sale event related information of all games. The proposed approaches outperform the general population prediction models that are trained to predict population during non-sale periods using past population, in predicting population during sale events. Results also indicate an increase of player population of games and changes in daily and weekly population changing patterns at the onset of the COVID-19 pandemic. It was also identified that games that become popular during the pandemic can be predicted utilizing machine learning classification approaches with a 69% accuracy.

# List of Publications

1. **Dulakshi Vihanga**, Michael Barlow, Erandi Lakshika and Kathryn Kasmarik, “Weekly Seasonal Player Population Patterns in Online Games: A Time Series Clustering Approach,” 2019 in *IEEE Conference on Games (CoG)*, London, United Kingdom, 2019, pp. 1-8.
2. **Dulakshi Wannigamage**, Michael Barlow, Erandi Lakshika and Kathryn Kasmarik, “Analysis and Prediction of Player Population Changes in Digital Games during the COVID-19 Pandemic,” 2020 in *Australasian Joint Conference on Artificial Intelligence*, Canberra, Australia, 2020, Springer, Cham.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Motivation . . . . .	2
1.3	Research Questions . . . . .	6
1.4	Original Contribution . . . . .	9
1.5	Organisation of the Thesis . . . . .	13
<b>2</b>	<b>Literature Review</b>	<b>16</b>
2.1	Game Data Analytics . . . . .	16
2.2	Many-Game Studies . . . . .	19
2.2.1	Approaches for Analysis of Player Behaviour and Games . . . . .	20
2.2.2	Approaches for other purposes . . . . .	23
2.2.3	Online tools displaying data and statistics of games . . . . .	24
2.3	Game Metrics . . . . .	25
2.4	Research Gap . . . . .	26
2.5	External Factors . . . . .	29
2.5.1	Studies on Short-Term Temporal Factors . . . . .	29
2.5.2	Studies on Product Life Cycles of Games . . . . .	31
2.5.3	Studies on Sale Events . . . . .	34
2.5.4	Studies on the COVID-19 Pandemic . . . . .	37
2.6	Background . . . . .	39
2.6.1	Time Series Analysis . . . . .	40
2.6.2	Time Series Forecasting . . . . .	42
2.6.3	Time Series Clustering . . . . .	42
2.7	Conclusion . . . . .	48
<b>3</b>	<b>Data Collection</b>	<b>50</b>
3.1	Research Methodology Overview . . . . .	51
3.2	Steam Platform . . . . .	53
3.3	Data Collection Approach . . . . .	55
3.4	Data Contribution . . . . .	60

3.5	Chapter Summary . . . . .	62
<b>4</b>	<b>Short Term Seasonality in Player Population Fluctuations</b>	<b>63</b>
4.1	Research Procedure . . . . .	65
4.1.1	Data Collection and Preprocessing . . . . .	65
4.2	Existence of Short Term Seasonality in Player Population Fluctuations	67
4.2.1	Autocorrelation based Seasonality Detection . . . . .	67
4.2.2	Trend Removal . . . . .	76
4.2.3	Outcomes . . . . .	80
4.3	Weekly Population Fluctuation Patterns Discovery . . . . .	83
4.3.1	Time Series Clustering . . . . .	83
4.3.2	Representative Weekly Patterns Generation . . . . .	91
4.4	Results and Discussion . . . . .	94
4.4.1	Extraction of Game Characteristics . . . . .	98
4.4.2	Discussion . . . . .	100
4.5	Conclusion . . . . .	103
<b>5</b>	<b>Player Population based Life Cycle Patterns of Games</b>	<b>108</b>
5.1	Research Procedure . . . . .	110
5.1.1	Overview . . . . .	110
5.1.2	Data Collection and Pre-Processing . . . . .	113
5.1.3	Piecewise Linear Regression for Life Cycle shape extraction . . . . .	117
5.1.4	Life Cycle Archetype Discovery through Clustering . . . . .	127
5.2	Results and Discussion . . . . .	129
5.2.1	Archetypes within 1 year after game release . . . . .	129
5.2.2	Archetypes within 3 years after game release . . . . .	136
5.2.3	Archetype Comparison . . . . .	140
5.2.4	Analysis of Game Characteristics . . . . .	144
5.3	Conclusion . . . . .	151
<b>6</b>	<b>Forecasting Player Population of Games during Sale Events</b>	<b>155</b>
6.1	Methodology . . . . .	157
6.1.1	Data Collection and Preprocessing . . . . .	161
6.1.2	Sale events and non-sale periods Extraction . . . . .	162
6.2	General Population Prediction Models for non-sale periods . . . . .	165
6.2.1	Prediction Models . . . . .	165
6.2.2	Evaluation Procedure . . . . .	168
6.2.3	Outcomes . . . . .	170
6.3	Sale event specific Population Prediction Models for sale event periods . . . . .	175
6.3.1	Prediction Models . . . . .	175

6.3.2	Evaluation Procedure . . . . .	186
6.4	Results and Discussion . . . . .	187
6.5	Conclusion . . . . .	196
<b>7</b>	<b>Player Population Fluctuations during the Novel Coronavirus (COVID-19) Pandemic</b>	<b>200</b>
7.1	Empirical Analysis of Player Populations during the early period of the COVID-19 pandemic . . . . .	202
7.1.1	Data Collection . . . . .	202
7.1.2	Changes in Player Population Size of games . . . . .	203
7.1.3	Weekly Patterns . . . . .	209
7.1.4	Daily Patterns . . . . .	211
7.2	Predicting the popularity of games during the onset of the pandemic	218
7.2.1	Data Collection . . . . .	219
7.2.2	Feature Extraction . . . . .	220
7.2.3	Classification Models . . . . .	222
7.2.4	Evaluation . . . . .	222
7.2.5	Results . . . . .	224
7.3	Conclusion . . . . .	228
<b>8</b>	<b>Conclusion</b>	<b>233</b>
8.1	General Discussion . . . . .	233
8.2	Limitations . . . . .	243
8.3	Future Work . . . . .	244
8.4	Concluding Remarks . . . . .	247
<b>A</b>	<b>Analysis of Game Characteristics of the Life Cycle Archetypes</b>	<b>249</b>
A.1	Introduction . . . . .	249
A.2	Characteristics of games displaying archetypes of first year ( $1Yclust$ )	249
A.3	Characteristics of games displaying archetypes of first three years ( $3Yclust$ ) . . . . .	261

# List of Figures

2.1	Product Life Cycle; Source: [1] . . . . .	31
2.2	Game Genre Life Cycle; Source: Daniel Cook [2] . . . . .	32
2.3	Game Life Cycles; Source: <i>HoneyTracks</i> [3] . . . . .	34
2.4	Monthly Airline Passengers in the United States [4] . . . . .	41
2.5	Distance Matrix of two time series; X and Y. The green color path depicts the optimal global alignment between the two series identified by Dynamic Time Warping . . . . .	44
2.6	Before and After DTW alignment of Two Time series X and Y (Series are presented in continuous form rather than point form to better visualize the distance between the series) . . . . .	46
3.1	Research Methodology Overview; Data is categorized as common and specific in the figure to depict data that are specific to each study and commonly used by the four studies. . . . .	52
4.1	Research Procedure Overview . . . . .	66
4.2	Population series and Autocorrelation function plot of <i>DOTA 2</i> game; blue arrows in the ACF plot points to the lags that are multiples of a day's lag. Green arrows points to the lags that are multiples of a week's lag. . . . .	69
4.3	Zoomed Autocorrelation function plot of <i>DOTA 2</i> game . . . . .	70
4.4	Population series and Autocorrelation function plot of the <i>CS2D</i> game . . . . .	70
4.5	Population series and Autocorrelation function plot of the <i>Worms Armageddon</i> game . . . . .	71
4.6	Population series and Autocorrelation function plot of <i>Sleeping Dogs: Definitive Edition</i> game . . . . .	71
4.7	Percentage of Games displaying Daily Seasonality under different lag ranges . . . . .	73
4.8	Percentage of Games displaying Weekly Seasonality under different lag ranges . . . . .	75
4.9	Before and after Linear Trend Removal of the game <i>CS2D</i> . . . . .	77

4.10	Before and after Order-8 Polynomial Trend Removal of the game <i>Elite Dangerous</i> . . . . .	78
4.11	Before and after Piecewise Trend Removal of the game <i>theHunter: Call of the Wild</i> . . . . .	79
4.12	The average daily player population pattern of games . . . . .	82
4.13	Peaks and piecewise trend of population plot of the game <i>observer_</i> . . . . .	85
4.14	Trend removed population plot of the game <i>observer_</i> . . . . .	86
4.15	DTW Alignment of weekly player population of two games . . . . .	88
4.16	Dendrogram(1508Games):Clustering of weekly patterns based on the DTW distance by Average Linkage . . . . .	90
4.17	Example: Weights used in Representative Pattern generation for a cluster . . . . .	93
4.18	Representative Weekly Player Population patterns of Clusters: Represents how player population changes from Monday to Sunday. Each peak corresponds to a day of the week . . . . .	96
4.19	Age based Class Distribution of Games in Clusters:- E: Everyone, E10+ : Everyone10+ (Ages 10 and up), T: Teens (Ages 13 and up), MA17+: Mature17+ (Ages 17 and up) . . . . .	100
4.20	Range of Mean Population of games in Clusters: Per each cluster, mean population of each game over 6 months is calcluated. The minumum, mean and maximum of those values are converted to $\log_{10}$ and presented . . . . .	100
5.1	Research Procedure . . . . .	111
5.2	Trend of the population series of the <i>TERA</i> game extracted by removal of weekly seasonality through moving average based data smoothing . . . . .	115
5.3	Incrementing Windows for series . . . . .	118
5.4	RMSE of iterations and final piecewise linear fit for <i>Monster Hunter:World</i> game . . . . .	122
5.5	RMSE of iterations and final piecewise linear fit for <i>SCP: Secret Laboratory</i> game . . . . .	123
5.6	Distribution of RMSE values and No of pieces of the piecewise linear fits when threshold is chosen as $t=0.2, 0.5, 0.8$ in Algorithm 5.1 . . . . .	125
5.7	Dendrogram for hierarchical clustering; Piecewise linear trend extracted using Algorithm 5.1 from the population data of first year after game release . . . . .	130
5.8	Archetypal life cycle patterns of games within the first year after release; Piecewise linear trend extracted by Algorithm 5.1 . . . . .	131

5.9	Dendrogram for hierarchical clustering; Piecewise linear trend extracted using Algorithm 5.2 from the population data of first year after game release . . . . .	134
5.10	Archetypal life cycle patterns of games within the first year after release; Piecewise linear trend extracted by Algorithm 5.2 . . . . .	135
5.11	Dendrogram for hierarchical clustering of life cycles of first three years after game release . . . . .	138
5.12	Archetypal life cycle patterns of games within the first three years after release . . . . .	139
5.13	Difference between the one year and three years life cycle patterns within the first year after release . . . . .	141
6.1	Methodology Overview . . . . .	158
6.2	Boxplot of the relative increase percentage of mean population during sale events . . . . .	164
6.3	Architecture of Non-linear autoregressive model ( <i>NARmodel</i> ); The population of the past seven days is used as the 7 input variables. The output variable is the predicted population. . . . .	167
6.4	Iterative cross-validation: Non-sale period instances dataset is split into train, validation and test sets iteratively considering temporal placement of non-sale periods for forecasting and split percentages;test set size is constant and test sets in each iteration never overlaps . . . . .	169
6.5	Regression plot of actual and predicted values resulting from <i>nsAllModel</i> when the model is trained to predict population during non-sale periods and tested on the same. (The <i>Non-sale</i> scenario) The figure depicts the outcomes for test set 3 of the non-sale period dataset. It can be seen that the model correlation between the predicted and actual values are high as represented by the R value of 0.97 . . . . .	173
6.6	Regression plot of actual and predicted values resulting from <i>nsAllModel</i> when the model is trained to predict population during non-sale periods and tested on predicting population during sale events. (The <i>Sale using non-sale</i> scenario)The figure depicts the outcomes for the sale event dataset used as the test set. It can be seen that the model has mostly under-predicted. It can be expected as the model was trained to predict population during non-sale periods where high population increases are not common compared to sale event periods. . . . .	175

6.7	Backward feature elimination outcome: Backward elimination is performed as per Algorithm 6.1. Order of eliminated features: Days since last Event, Final Price, month, Start Date, End Date, Pre event count, duration, Initial Price, Final Price Percentage, Weekend Count, discount, Days since Release . . . . .	180
6.8	<i>sAllModel</i> Overview . . . . .	183
6.9	<i>sClustModel</i> : Life cycle shape cluster based Multi Layer Perceptron model overview: To predict the maximum population of a given sale event, first the cluster to which the game of the sale event belongs is identified, then the corresponding <i>sAllModel</i> is used to generate the prediction . . . . .	184
6.10	Architecture of Nonlinear autoregressive exogenous model ( <i>NARX-model</i> ) . . . . .	186
6.11	Regression plots of actual and predicted values resulting from <i>sAllModel</i> for test set 3: First plot depicts the regression plot of training dataset which has a R value of 0.74. Next plot depicts the regression plot for Validation dataset which has a R value of 0.85. It has the highest regression correlation coefficient out of the three datasets(training, validation and test). The next plot depicts the regression plot for test set which has a R value of 0.70. More under-predicted instances can be observed compared to over-predicted instances in each of the training, validation and test set regression plots. The final plot depicts the regression plot of all sets; training, validation and test, in a single plot which has a R value of 0.76 . . . . .	191
6.12	Regression plot of actual and predicted values resulting from <i>sClust-Model</i> for test set 3; Cluster 1, 2 and 3 plots are presented while Cluster 4 is not presented due to the unavailability of Cluster 4 games in the data set. Each sub figure depicts the regression plots for training, validation, test set and all those three sets together . . . . .	193
7.1	Data series of the daily aggregate player population, Global COVID-19 cases and US COVID-19 cases. 16th March is the date US president enforced restrictions on gatherings for up to 15 days. The Pearson correlation between the population trend and the global COVID-19 cases is 0.87. The Pearson correlation between the population trend and the US COVID-19 cases is 0.79. . . . .	205
7.2	Distribution of correlation between daily global COVID-19 cases and population of individual games . . . . .	206

7.3	Histogram of mean population change percentage of games during COVID-19 . . . . .	207
7.4	<i>Garry’s Mod-</i> Normalized Player Population during and prior to COVID-19 . . . . .	211
7.5	<i>Rocket League-</i> Normalized Player Population during and prior to COVID-19 . . . . .	211
7.6	<i>Garry’s Mod-</i> Player Population during and prior to COVID-19 . . . . .	212
7.7	<i>Rocket League-</i> Player Population during and prior to COVID-19 . . . . .	212
7.8	Daily Mean Population Pattern of <i>DOTA 2</i> and <i>Rocket League</i> during the onset of COVID-19 before and after realignment using Algorithm 7.1 . . . . .	214
7.9	Aggregate Daily Player Population Pattern during and prior to COVID-19 . . . . .	216
7.10	Mean normalized Daily Player Population Pattern of all games during and prior to COVID-19 . . . . .	217
7.11	Confusion Matrix of the Random Forest model: The confusion matrix of an evaluation run of the random forest model depicting true and predicted classes. The row-wise percentages represent the percentage of correctly and incorrectly classified instances for each true class out of the total instances of the respective true class. The column-wise percentages represent the percentage of correctly and incorrectly classified instances for each predicted class out of the total instances of the respective predicted class. . . . .	225
7.12	Prominent tags of the highly popular games; For each tag the percentage difference between the highly popular games class and not highly popular games class is presented. Only the tags in which the difference is higher than 0.5% are presented in the chart . . . . .	227
A.1	Common tags among clusters; first year archetypes . . . . .	250
A.2	Partially common tags among clusters; first year archetypes . . . . .	251
A.3	Common tags and their percentage difference between clusters and dataset ; first year archetypes . . . . .	252
A.4	Partially common tags and their percentage difference between clusters and dataset ; first year archetypes . . . . .	253
A.5	Unique Tags and their percentage difference between clusters and dataset; first year archetypes . . . . .	254
A.6	Release year distribution of games displaying first year archetypes . . . . .	255
A.7	Release month distribution of games displaying first year archetypes . . . . .	256
A.8	Mean population statistics of games displaying first year archetypes . . . . .	257

A.9	Price distribution of games displaying first year archetypes . . . . .	259
A.10	Positive review percentage distribution of games displaying first year archetypes . . . . .	260
A.11	Common tags among clusters; first three years archetypes . . . . .	262
A.12	Partially common tags among clusters; first three years archetypes . .	263
A.13	Release year distribution of games displaying first three years archetypes	264
A.14	Release month distribution of games displaying first three years archetypes	265
A.15	Mean population statistics of games displaying first three years archetypes	266
A.16	Price distribution of games displaying first three years archetypes . .	268
A.17	Positive review percentage distribution of games displaying first year archetypes . . . . .	269

# List of Tables

3.1	Entertainment Software Rating Board (ESBR) Game Ratings . . . . .	58
3.2	Properties of <i>Gameset1</i> and <i>Gameset2</i> . . . . .	61
4.1	Percentage and Number of Games displaying Daily Patterns . . . . .	80
4.2	Percentage and Number of Games displaying Weekly Patterns . . . . .	80
4.3	Cophenetic Correlation Coefficient for different Linkage methods . . .	89
4.4	Characteristics of the 9 weekly patterns . . . . .	97
4.5	Top 10 Tags of each cluster sorted by the weighted percentage of games	99
4.6	Characteristics of the games in Clusters . . . . .	104
5.1	Cophenetic Correlation Coefficient for different linkage methods; Piecewise linear trend extracted using Algorithm 5.1 from the population data of first year after game release . . . . .	129
5.2	Cophenetic Correlation Coefficient for different linkage methods; Piecewise linear trend extracted using Algorithm 5.2 from the population data of first year after game release . . . . .	133
5.3	Cophenetic Correlation Coefficient for different linkage methods; clustering life cycle shapes of three years . . . . .	137
5.4	Game transition percentages from first year archetypes to three year archetypes . . . . .	143
6.1	Boundary dates that provided the required split percentage of training, validation and test sets in each iteration of the cross-validation process; $s_d$ represents the start date of a selected non-sale period and $e_d$ represents the end date of a selected non-sale period . . . . .	170
6.2	RMSE results from <i>nsAllModel</i> and <i>nsClustModel</i> for the three evaluation approaches; Non-Sale: Model trained on Non-Sale period data and tested on the same, Sale: Model trained on Sale event period data and tested on the same, Sale using non-sale: Model trained on Non-Sale period data and tested on Sale event period data . . . . .	171

6.3	RMSE results from <i>NARMModel</i> ; Non-Sale: RMSE for predicting maximum population during non-sale periods, Sale: RMSE for predicting maximum population during Sale periods . . . . .	171
6.4	Boundary dates that provided the required split percentage of training, validation and test sets in each iteration of the cross-validation process; $s_d$ represents start date of a sale event and $e_d$ represents end date of a sale event . . . . .	172
6.5	Initially identified features for the prediction model; After the feature selection process Initial Price and Start Date features are removed . .	176
6.6	RMSE results from all prediction models for each test set in cross-validation; Set number represent the iteration number in cross-validation	189
6.7	RMSE results from all prediction models for all test sets combined: Baseline model results are the outcomes of predicting maximum player population during sale events of the sale event dataset using general population prediction models that were trained to predict maximum population during non-sale periods . . . . .	189
7.1	Features used in the classification models . . . . .	221
7.2	Performance of Classification Models . . . . .	224
A.1	Free to Play games percentage of games displaying first year archetypes	258
A.2	Free to Play games percentage of games displaying first three year archetypes . . . . .	267

# Chapter 1

## Introduction

### 1.1 Overview

The digital gaming industry has become one of the most prominent entertainment mediums attracting younger and older generations alike. According to Newzoo market research [5], the number of game players across the world in 2019 is more than 2.5 billion. It is expected that the number of game players will rise to over 3 billion by the year 2023 [6]. Due to this ever-increasing number of players, the digital gaming industry has great revenue growth. The industry has generated \$120.1 billion in income in the year 2019 alone as reported by the SuperData market research [7].

The number of game players across the world continues to increase every year [6]. Deciphering the complex behaviour of these game players is not only beneficial but also vital for the advancement of gaming as it helps to better understand the game players and provide a better gaming experience. Hence, game data analytics has started to gain attention lately [8]. Game data analytics is recognized as the process of revealing and communicating patterns from data applicable for game development and research, which includes generating insights regarding games and players [8]. Game data analytics use various sources of data such as player-related data and game-related data to generate insights. In this context, player-related data is considered as data related to players generated by player activity. Game-related data are considered to be data generated due to various processes related to

the game but not generated directly from the game, such as sales data and game reviews. There are various circumstances in which game data analytics can aid. Assisting in better understanding players with respect to their play patterns and preferences is one major instance. However, due to limited access to player-related data, player behavioural studies have been limited to single or a few games [9] which in turn has constrained the knowledge about player populations across games. Furthermore, the number of players could ebb and flow in games in the presence of various factors such as game features, time since release, day of the week, sale events, and other factors. However, there is a lack of understanding of player population fluctuation patterns that emerge in the presence of such external factors, where the external factors are defined in this context as factors that are not inherent features of the game product/software. But recently data sources that are essential for understanding this phenomenon have been becoming publicly accessible namely, the data that can be collected from the Steam platform. It is providing hitherto unprecedented opportunities for exploring player population fluctuations of games in the presence of such external factors. This thesis presents a study conducted to understand player population fluctuation patterns of games in the presence of external factors, namely, time of the day, day of the week, time since game release, sale events and pandemics. Investigating such factors, this thesis not only generates insights about daily and weekly seasonality of player population fluctuations, life cycle patterns of games based on player population fluctuations, sale events and pandemic related population changes of various games but also provides predictions related to population changes in the presence of such external factors. Ultimately, this thesis aids in advancing the knowledge about player populations and games.

## 1.2 Motivation

Game data analytics have gained increasing attention as a consequence of the massive growth of the digital game industry. Analytics is the process of discovery, interpretation and communication of principal patterns observable in the data [10].

Game data analytics uses various sources of data to provide aid in the game design process, business decision process, marketing process and in all the other aspects of the decision-making process [11]. Understanding players and recognizing their behavioural patterns are one of the main use cases of game data analytics. Understanding player behavioural patterns could aid in player retention, churn prediction [12], monetization [13] and especially in enhancing the game design [14]. In general, game data analytics plays a major role across the life cycle of a game. Moreover, as the number of games introduced to the market increases every day, game data analytics are becoming vital for a game to get ahead of the competition and for new games to further improve.

Despite the profound importance of game data analytics, various constraints have impacted the full capability of game data analytics in the past. Among many factors, inadequate access to player-related data has played a major part in the limitations in insights generated from game data analytics [9]. Specifically, most game studies have been limited to a single or a small number of games [15]. For instance, some studies are limited to a few Massively Multiplayer games [16] [17] [18]. These studies have contributed towards understanding player behaviour of various games. Furthermore, individual game analytics continue to nurture games by supporting the analysis of in-game player behaviour. However, limitations associated with data have limited the full potential of game data analytics.

In order to further strengthen the understanding concerning game players and games, *many-game* analysis studies are important. Many-game studies can be defined as studies that are not limited to a single game and its players but studies that investigate player-related data and game-related data of multiple games across the same time periods. Such studies could aid in understanding game player behaviour across games which could, in turn, assist the digital game industry to learn about general player behaviour without being restricted to a handful of games [9]. Limited access to data generated in games and the proprietary nature of player behavioural data and analytics techniques could be the main reason for the lack of many-game

studies in the past [9]. However, with the emergence of big data technologies and various online game distribution platforms, such as Steam, access to data that can be used to analyse player behaviour across games is no longer fully limited. Especially, data related to games and players made available by Steam, such as playtime of players in various games and games owned by players has already assisted several many-game analysis studies. These studies reveal various patterns of players with respect to game ownership [9], time and money spent across games, and social interactions [19] providing a better comprehension of player behaviour across games. Hence, it can be anticipated that game data analytics could now assist the digital game industry more than ever with many-game research.

Data collected from games are often transformed into interpretable measures, which are known as game metrics [20], and used in game data analytics. As game players are important for the success of a game, some of the widely used game metrics are focused on measuring player population [21]. Some of those game metrics are daily active users, monthly active users, and peak concurrent users [21]. Keeping track of player population is important to understand the overall popularity of a game [21]. The number of users is tracked and analysed not only in games but also in other applications, such as Facebook and Instagram [22].

Player population in games fluctuate in the presence of various factors such as time of the day, social media influence and marketing events. Analysing the changes in player population of games using game data analytics could help the business decision making process by generating insights related to the time of the day most players are active [23], effectiveness of marketing and promotion campaigns [24], user engagement and monetization [25] and various other use cases. Pittman and GauthierDickey have investigated the daily player population patterns of two games based on changes in player population [17]. Zhuang et al., have also identified a time-of-day effect in player population changes in the *World of Warcraft* game [26]. Such a time-of-day effect at the population level is understandable as it can be observed at an individual player level as well. For instance, a survey conducted by Triberti

et al., [27] using game-playing individuals has identified that young individuals tend to play more during the afternoon. The study of King and Hera has identified that *Fortnite* game players tend to play more influenced by the Youtube streamers of *Fortnite* game [28] indicating a change in player population in the presence of social media influence. Moreover, Johnson and Woodcock mention that the streaming platform Twitch is used for marketing games which results in a change in player population of games extending their lifespan [29]. They point out that the game *Rocket League* became popular in Twitch and had an increase of players soon after launch [29]. Choi et al., have identified that the discount pricing strategy has an effect on the sales of video games [30] indicating a change in the player population. Thus it can be understood that player population of games fluctuate in the presence of various factors such as time of the day, social media influence and marketing events.

As mentioned earlier, player population of games are often analysed using game metrics to keep track of how the game is doing in general [20] and in the presence of various factors such as day of the week and marketing campaigns. These analyses are currently performed by the game's developer or the game team using the data collected in the game. Various online tools ([www.GameAnalytics.com](http://www.GameAnalytics.com)) available support the game developers in the analysis process. Flunger et al., explains that it is important for small and medium-sized game developers to use game analytic tools [31]. However, insights generated from game analytic studies have not been made widely available as those contain business intelligence some game companies are reluctant to share [8]. Due to this per-game approach of analysis, currently few many-game studies regarding changes in player population of games are available with the only partial exception being the study of Chambers et al., [32] that analyses player population of a collection of games for a game server workload investigation identifying daily population changing patterns. Thus, there is a gap of knowledge regarding how player population of games change in the presence of various factors such as time of the day, sale events and world events although these are studied at

a per-game level as explained previously. Recently there has been an increase in many-game studies that have generated valuable insights useful for game developers on player behaviour including game ownership and playtime [19] [9] game updates and reviews [33] that are not limited to a single game, utilizing data that have been made available in the Steam platform. The study of Prathama et al., have used data from Steam including the population size of games to predict the future popularity of games [34]. It is important for game developers, especially indie developers who do not have data as large game companies do, to be aware of the current game market [34]. Many-game studies are helpful for this purpose. Hence, there is a potential for many-game studies focused on player population changes of games to enhance the current knowledge related to games. It could aid game developers to understand the common player population changing patterns in various games in the presence of various factors and apply that knowledge appropriately to their games. Thus this thesis aims to advance the knowledge about player population fluctuation patterns in the presence of various external factors.

### **1.3 Research Questions**

As discussed previously, studying player population changes in games is important and it is already being conducted in games utilizing population-related game metrics [21]. The player population of games change in the presence of various external factors such as time of the day [17] and social media influence [28]. However, as previously explained analyses regarding player population changes are conducted in a per-game approach which limits the applicability of the outcomes across other games. Also in general, data and insights generated by game companies are not widely shared [8]. Hence there is a lack of knowledge regarding player population changing patterns in games. However, such studies are important for game developers to become aware of the current digital game industry trends, especially for indie developers who do not have their player-related data to conduct investigations [34]. However, Steam has now paved the way to many-game studies by providing data.

### *1.3. Research Questions*

---

Several many-game studies have already been conducted which are focused on player behaviour including game ownership and playtime [19] [9] generating valuable insights regarding games and game players without being limited to a single game. Hence, considering the importance of analysing player population changes in games and many-game studies for game developers, this thesis is focused on investigating player population fluctuation patterns among games based on various external factors so as to generate insightful patterns and predictions about player populations and games. To this end, the main research question of this thesis can be articulated as follows:

**How does player population of games change in the presence of various external <sup>1</sup> factors?**

The external factors considered are temporal factors, namely, time of the day, day of the week and time since the release of the game and event-related factors, namely, sale events and the onset of the COVID-19 pandemic. These factors were chosen as the literature indicates that changes in the number of players of games or the number of consumers of products can be observed in the presence of these factors. The study of Pittman and GauthierDickey presents how player population of two games change based on the time of the day and day of the week [17]. Chambers et al., have identified player population changes based on the day of the week [32]. Gazecki has identified two shapes of game life cycles based on the player population changes since game release [3]. Choi et al., have investigated the sales of games on Steam and have identified that the sales are positively influenced by discounts [30] indicating a possible change of population after sale events offering discounts. Several studies have reported an increase in sales of video games and time spent playing during the COVID-19 pandemic [35] [36]. In this thesis, studies are conducted to discern patterns of player population fluctuations in the presence of the mentioned external factors. Furthermore, several sub research questions are formulated to investigate the main research question.

---

<sup>1</sup>The term *external* is defined in this context as not inherent features of the game product/software. For instance, time of the day and sale events

- **How does player population of games fluctuate during a day and a week?**

The amount of time people have for recreational activities depends on other activities people have to attend to in their day-to-day life, such as job-related activities [27]. Hence, the time of the day and day of the week could impact their playing habits. Games tend to attract various players that may have similar preferences in games. Hence, this research question is focused on investigating whether game player population fluctuations display short term seasonality<sup>2</sup> such as daily<sup>3</sup> or weekly<sup>4</sup> patterns. Furthermore, investigations are conducted to identify archetypal population fluctuation patterns games display with respect to weekly seasonality.

- **How does player population of games fluctuate during the first year and first three years after game release displaying life cycle shape approximations?**

Like any other product, games also have a product life cycle of their own. Once a game is released, the shape of the life cycle of a game can be determined based on player population fluctuations. Player population of games could change through time based on the time since a game was released. However, the patterns in which population change throughout the lifetime of a game could be diverse among various games and more importantly among types of games. Thus, this research question investigates the means to extract and interpret the life cycle archetypes of games represented by long term player population fluctuations of games. It is focusing on the population fluctuations during the first year and first three years after the game release.

- **How accurately can we forecast player population of games during sale events?**

---

<sup>2</sup>*Short term seasonality* : Recurrence of same changes at known short time intervals such as day and week

<sup>3</sup>*Daily patterns*: Recurrence of same pattern of player population changes every day

<sup>4</sup>*Weekly patterns*: Recurrence of same pattern of player population changes every week

Sale events are special periods where players buy and play new games getting economical advantages of discounts provided. During sale events, player population of games could vary rapidly. However, an improved understanding of player population fluctuation patterns could enhance the capability of population forecasting even during such special periods. Thus, this research question is focused on identifying an approach to accurately predict changes in player population size that can be expected during sale events.

- **How does player population of games fluctuate during pandemics?**

Lifestyles of people change tremendously during world crises such as the COVID-19 pandemic. Changes in the working environment, staying at home, social gathering restrictions and various other changes introduced during this period impact the normal day-to-day lifestyle of individuals. Hence, game playing patterns could also be influenced. This research question is focused on understanding how player population changes in various games during the early period of the COVID-19<sup>5</sup> pandemic. Furthermore, the question also attempts to predict the games that are likely to have an increase in population during the onset of the COVID-19 pandemic which would help in further understanding player preferences during such special times.

**Thesis Statement:** Analysis and predictions on player population changes of digital games in the presence of external factors, namely, time of the day, day of the week and time since the release of the game, sale events and the onset of the COVID-19 pandemic in a many-game approach can provide insights and suggestions that are important for game developers.

## 1.4 Original Contribution

This section presents the main contributions of this thesis.

---

<sup>5</sup>Since the COVID-19 pandemic is still ongoing at the time of writing, this work is considering the pandemic period up until the 16th of April 2020 only

- Insights generated regarding short term seasonality in player population fluctuations and the overall methodology behind the process (Chapter 4).

Studies in the literature that have analysed the recurring daily and weekly patterns of player population and play session counts in games are limited to a single or a small number of games [17] [26], with the only exception being Chambers et al., [32] which is also limited to 50 games. Hence, the applicability of the analysis methodology and generated insights of the studies across multiple various games is limited. Thus, there is less understanding of short term seasonality in player population fluctuations. While addressing this gap in the literature, this thesis provides a comprehensive study on short term seasonality in player population fluctuations using a large scale player population dataset of 1963 games that stretch over 6 months. Key insights indicate that 68% of games display daily patterns and 77% of games display weekly patterns in player population fluctuations. Moreover, 9 different archetypal weekly player population fluctuation patterns have been revealed along with the frequency of the patterns. Furthermore, game characteristics, namely, tags, age requirements and overall population size of the games representing each archetypal pattern were analysed. This contributes to further enhance the current knowledge about daily and weekly patterns of player population fluctuations among a wide range of games. Additionally, it was demonstrated that a methodology that employs autocorrelation, trend removal and DTW-based (Dynamic Time Warping) time-series clustering methods is appropriate for the analysis of the influence of temporal factors, namely, time of the day and day of the week on player populations.

- Archetypal life cycle shape approximations of games. These archetypal shapes depict the long term player population changes of games during the first year and first three years after game release (Chap-

ter 5).

Knowledge about life cycle shapes of games depicting long term player population changes can provide an understanding of the life expectancy of games and the life stages various types of games go through. However, related studies in the literature have only focused on genre-based life cycles [2] and retention-based life cycles [3] [9]. In this thesis, archetypal life cycle shapes of games based on player population changes are revealed through a clustering approach. It utilizes a novel piecewise linear regression based algorithm introduced to identify life cycle stages. Four archetypal life cycle patterns were identified, using a dataset of 683 games containing their player population data since the release date of each game. These patterns depict the long term player population changes within the first year and first three years after a game is released. Game characteristics, namely, tags, release year, release month, population size, publishers, developers, price and reviews were also analysed to provide a comprehensive description of the revealed archetypal patterns. It is expected that the contribution of the game life cycle stage identification algorithm and the archetypal life cycle shapes generated using it will provide an extended understanding about long term player population changes of games considering the age of a game.

- **Sale event related player population forecasting models and related findings (Chapter 6).**

Game player populations fluctuate during sale events in which games are sold for a discounted price. In this thesis, three prediction approaches were explored to determine how accurately the maximum player population during sale events of games can be predicted. The three approaches are, a Multi Layer Perceptron (MLP) model that uses past sale event information of all games, a life cycle archetype based MLP model which is a variant of the introduced

MLP model that uses past sale event information of the games that display similar life cycle shapes, and a non-linear autoregressive exogenous model that uses the game's own historical population and sales data. Furthermore, three general population prediction models that use past population to predict population during non-sale periods were also used in the study for comparison. The study revealed that it is challenging to accurately predict the maximum population during sale events compared to predicting the maximum population during non-sale periods. However, it was identified that more accurate predictions on the maximum population during sale events can be predicted by using sale event related features along with past population in prediction approaches that use sale event information of other games as well. This indicated the importance of using sale event specific prediction models rather than general prediction models trained to predict population during non-sale periods.

- **A comprehensive analysis of player population fluctuations during the onset of the COVID-19 pandemic and the related classification models predicting games that are likely to be popular during pandemic periods (Chapter 7).**

COVID-19 is a new form of the coronavirus that causes respiratory infections which resulted in a global pandemic. Analysis of player population changes in games during this period revealed that player population of games have significantly increased at the onset of the COVID-19 pandemic compared to the population during the same month in the previous year and during popular Steam sale events. A 33% increase of the total player population of games was observed after the 16th of March 2020 when social restriction rules were enforced in the US. Changes in daily and weekly player population patterns were also revealed. Furthermore, investigations were conducted to utilize machine learning classification models to predict games that would become highly

popular during the pandemic. Decision tree, tree bagging, tree boosting, random forest and support vector machine models were trained. However, only a 69% of prediction accuracy was achieved. Furthermore, it was identified that Adventure, Racing, Multiplayer and Boardgames are popular during the pandemic.

- The final contribution of this thesis is the dataset<sup>6</sup> of player population of 2000 games available in Steam collected over two years and eight months in short intervals, namely, 5 minutes (one subset) and 1 hour (another subset) (Chapter 3).

Player population data is collected through two sources to reduce the amount of missing data. Currently, there are no publicly accessible game player population datasets of such temporal precision. The dataset also contains price history of the games and game-related information, namely, release date, genre, tags, publishers, developers and supported languages. Hence, this dataset would provide the opportunity for researchers to study game player populations. It is useful for time series analysis and forecasting studies.

## **1.5 Organisation of the Thesis**

The rest of the thesis is organized as follows.

Chapter 2 provides a general background for the work presented in this thesis by reviewing existing literature. It presents an introduction to game data analytics. Also, many-game studies are introduced to describe its importance for the digital game industry and the current state of research in that area. Specifically, many-game studies related to player behaviour and game analysis and other purposes are presented. Furthermore, existing online tools displaying data and statistics of games are also described. Moreover, the literature related to the external factors investi-

---

<sup>6</sup><https://data.mendeley.com/datasets/ycty3sy3vj2/1>

gated in this thesis are presented. Finally, an introduction to time series analysis, forecasting and clustering is provided to introduce time series related concepts used throughout the thesis.

Chapter 3 presents the data collection methodology of the thesis. Initially, the Steam platform is introduced which is the data source of the thesis. Next, the data collection procedure related to player population data, game characteristics such as genre, tags, age restrictions, release date and game price information is explained.

Chapter 4 presents the study investigating short term seasonality in player population fluctuations. It first identifies games that display daily and weekly player population fluctuation patterns using an autocorrelation based approach utilizing several trend removal techniques. Next, archetypal weekly player population fluctuation patterns of games are revealed through a time series clustering approach.

Chapter 5 details the study focused on identifying the life cycle archetypes of games based on long term player population patterns. It is aimed at understanding the influence of the time since a game is released on its player population. It first presents two algorithms that identify life cycle stage of a game based on a piecewise liner trend extraction approach. Next, archetypal life cycle patterns during the first year and first three years are revealed through a time series clustering approach.

Chapter 6 presents a study investigating player population fluctuations during sale events. It presents three prediction approaches that predict the maximum player population during sale events. The approaches are focused on generating a single prediction model for all games, prediction models per each cluster of games that display similar life cycle shape and prediction model per each game. Furthermore, population prediction models that use past population to predict population during non-sale periods are also investigated to understand their capability in predicting population during sale events.

Chapter 7 presents a study focused on player population fluctuations during the onset of the COVID-19 pandemic. It first provides a detailed analysis on player population changes observed during the pandemic period revealing changes in player

### *1.5. Organisation of the Thesis*

---

patterns and preferences. It then presents an investigation related to predicting games that are likely to observe a significant increase of players during pandemic periods through a machine learning based classification approach.

Chapter 8 is the final chapter of the thesis and provides the conclusion. It presents a general discussion on how the research questions were addressed, the findings and their implications, limitations and future directions arising from this thesis.

# **Chapter 2**

## **Literature Review**

This chapter provides an overview of the game research studies in the literature to provide a background and reviews the current limitations of related studies that motivates the work conducted in this thesis. First, the domain of game data analytics and its prominent use cases are introduced. Subsequently, many-game studies are presented focused on analysing games and player behaviour across games. Thereafter, the current limitations in the literature are identified. Next, studies related to the external factors considered in this thesis are reviewed. Specifically, short term temporal factors, namely, time of the day and day of the week, product life cycles related to the time since release factor and sale event and COVID-19 pandemic related studies are presented. Finally, a theoretical background to the work conducted in this thesis is provided through an introduction to the time series analysis, forecasting and clustering.

### **2.1 Game Data Analytics**

Game data analytics is the process of analysing player-related data and game-related data with the purpose of generating insights that could aid in the decision making related to game design, game development, marketing and other business decisions. Game analytics field is a combination of statistics, machine learning and data mining approaches applied to the context of games [10]. Even though game analytics stems

## *2.1. Game Data Analytics*

---

from such well-established research disciplines it is a relatively new domain [8]. However, within a short time, it has become a fundamental aspect of the digital game industry. This is due to the steady growth of the game player community that generates massive amounts of data yielding a wealth of information. The types of data that can be used in game data analytics is quite substantial. This includes play time, game activity, purchasing history, game reviews and much more. Utilizing such data in game analytics can aid in providing a better gaming experience to users through potential enhancements in the game. Furthermore, game analytics are also used to further improve player retention, monetization, fraud detection, player profiling and many other purposes. Also, several specialist game analytic service providers already exist and provide analytics facilities to the game developers [37] [38].

Game analytics is commonly utilized to deepen the understanding of players. To this end, studies have been conducted commonly focusing on player retention, player churn, player profiling and monetization use cases. Player retention refers to keeping players interested in a game for a longer period of time. Studies have been conducted to analyse the factors related to player retention. Some of these studies have focused on investigating the design features useful for player retention [39], identifying factors that influence player retention on MMORPGs [40], analysing player retention approaches in *World of Warcraft* [41], dynamically adapting a game to increase player retention based on game analytics [42]. Studies have also been conducted to generate predictions regarding player retention [43] [44]. Player churn refers to the situation where a player leaves a game and probably never returns. Several player churn prediction models have been proposed in the literature using game-specific and universal characteristics. Some of these studies have proposed churn prediction models based on hidden markov model based approach [45], based on features that are generic to games such as playtime and session length [46] [47], based on survival analysis [48], based on both player engagement and social influence of other churners [18]. Player profiling and segmentation is an essential mechanism

## *2.1. Game Data Analytics*

---

for understanding players. It is used to model individual player behaviour and identify groups of players based on similar characteristics. There are various studies in the literature that have focused on player profiling to identify behavioural profiles of players based on character related and game related features [49] [50], based on play style changes [51] and game play statistics [52]. Monetization is identified as the process that generates revenue from games. Monetization related studies have been conducted to identify purchasing patterns of players [53] [54] and to make predictions regarding future purchases [13] [55]. Each of these use cases is often connected and goes hand in hand. For instance, predicting player churn events could help with initializing action encouraging the player to stay, increasing retention [12]. Additionally, player segmentation based on in game player behaviour could help enhance player retention by designing customized player retention strategies for player segments [56].

Game data analytic studies have contributed to broadening the knowledge about games and players. However, publicly available studies on player behavior have been limited to a single or a few games rather than many games [9]. Key reasons for the lack of many-game studies could be the reluctance of game companies in sharing their confidential player-related data and the lack of publicly available datasets in the past [57] [9]. Digital game distribution platforms, such as Steam, have provided more opportunities for developers to distribute their games [58]. Hence, game players also have more opportunities in accessing games, owning multiple games, and migrating between games easily [58]. Although more games are accessible to game players and the same player can own many games, the limiting of game analytic studies to individual games constraints the understanding of player behaviour across games [9]. The lack of many-game studies places several limitations upon obtaining more benefits from game analytics useful for the video game industry and research. For instance, many-game studies can be helpful to model a player’s behaviour based on the player’s current games that can then be utilized to build models in a game the player newly joins aiding the cold-start problem [59] [60]. Also, many-game studies

can be used to better understand the purchasing behaviour of game players and improve the game recommendation process [57]. Furthermore, many-game studies could provide insights regarding the playtime distribution of players across games to learn more about players and their preferences [9] [19]. Studying various games and their players could provide deeper knowledge about player preferences and behaviour in different types of games as it supports the study of player behaviour across games rather than being limited to a single game. In essence, there is limited understanding of how the game analytic studies that are limited to a single or a few games can be applied across games [9]. However, as players have more access to games and the opportunity to move between games it is becoming useful to understand game player behaviour across games without being limited to a few games to enhance the knowledge about games and players. Hence, many-game studies are becoming important.

## **2.2 Many-Game Studies**

Many of the game studies have been limited to a single or few games in the past with limited knowledge regarding how the outcomes of the studies can be applied across games [9]. For example, Ducheneaut et al., have explored the social dynamics including play patterns and grouping patterns of the game *World of Warcraft* [16]. The study by Pedersen et al., have generated player experience models for a platform game using gameplay features and controllable features [61]. Kawale et al., have investigated player churn in *EverQuest II* based on both player engagement and social influence of other churners [18]. The study of Mahlmann et al. has focused on predicting aspects of playing behaviour in *TombRaider: Underworld* [62]. However, as players have more access to games with the popularity of online game distribution platforms, many-game studies have been emerging [9] [19].

As explained in Chapter 1, many-game studies can be interpreted as studies conducted based on multiple games to generate insights or predictions using the data from all games collaboratively. Currently, several many-game studies do exist in the

literature and those are reviewed in this section. These studies have been conducted using data the researchers have collected or using data provided by some game companies. Most of the studies have used Steam, the digital distribution platform for video games, as the chief data source [9] [19] [63] [64]. Currently existing many-game studies have focused on further understanding player behaviour in various games and also on further understanding games and their differences through player behaviour. Furthermore, there are other purposes associated with some such studies. This section presents the many-game studies in the literature and their purposes and outcomes. Moreover, details about some existing online visual tools for many-game analysis are also presented.

### **2.2.1 Approaches for Analysis of Player Behaviour and Games**

Game data analytics can aid in analysing massive amounts of game telemetry spread across games to enhance understanding of player behavioural patterns and games. In this section, studies that have analysed the player-related data and game-related data regarding the actions of large groups of players across many games focused on revealing insights regarding player behaviour and games are reviewed. Specifically, studies that investigate player behavioural patterns that are not limited to individual games such as how players distribute their playtime across games, game ownership patterns, player migration between games and also game community related social structures of players and cheater behaviour are chosen. Also, studies that reveal insights and predictions regarding games based on player behaviour are selected which helps to analyse games with respect to their differences, similarities and player preferences. These studies have conducted research focusing on examining the popularity of games, the predictability of player population in games, game ownership preferences, genre preferences, playtime based retention profiles of games and the predictability of success of a video game. The studies are further explained as follows.

The study of O’neil et al. has investigated various aspects of player behaviour

using data about 108.7 million players of the Steam platform [19]. The study has shown that on Steam, game ownership and games played follow a long-tailed distribution. The study shows that the Action genre is the most popular genre of owned games [19]. It is significantly higher than the Strategy and Indie genres. Also, as per the same study, playtime distribution of players on Steam shows a heavy-tail distribution and follows the 80-20 rule in which the top 20% of Steam players are responsible for 82.4% of the total playtime of players across games [19]. The study has also identified playtime based genre preferences. As per the study, playtime data of Steam players have revealed that multi-player games are more popular compared to single-player games. Furthermore, the Action genre has been identified as the most played game genre. The study has also investigated social structures including friends and groups of players on Steam.

The study of Baumann et al., [64] identified six clusters of hardcore game player behaviour using a subset of the Steam dataset of the previous study of O’neil et al. [19]. Hardcore game players are defined in their study as game players that play more than 20 hours per week. The six types of hardcore game players identified are First Person Shooters, Team Fortress 2 player, Action game player, DOTA 2 Player, Strategy and Action Combiner and Genre-switching player. Their analysis of playtime of hardcore game players reveals that the Action genre games are the most played genre which is higher than Indie, RPG, Simulation, and Strategy. A similar result regarding Action genre is also revealed in the study of O’neil et al. [19]. Furthermore, the hardcore game player clusters that have been identified also displayed that sports games and racing games are less played.

The study of Sifa et al. have also revealed various insights regarding players and games using playtime data of 6 million players in Steam covering 3000 games [9]. They have initially identified five player clusters based on their playtime distribution in which four clusters are representing players of individual games while only one cluster, containing nearly half of the players in the dataset, is representing players that play multiple games. Furthermore, their extended cluster solution has revealed

11 clusters of players. Interestingly, there are some similarities among the clusters identified in this approach and in the hardcore game player clusters by Baumann et al., [64]. Specifically, the FPS shooters, Team Fortress 2 players, DOTA 2 players and Genre-switching players which match with Active steam players cluster. Moreover, the study of Sifa et al., has also analysed the game ownership patterns in Steam using frequent itemset mining and association rule mining [9]. They have identified several association rules revealing sets of games most frequently played per player. Furthermore, out of the identified most frequently played games the majority of the games have appeared to be flagship games from the Valve company. Different retention profiles displayed by games have also been identified in the study of Sifa et al., based on aggregate playtime data of Steam games [9]. The conducted archetypal analysis has revealed four patterns; games with short playtime, games in which playtime peaks at 4 hours and quickly drops soon after, games with slow decaying playtimes and games with slow decaying playtime but dominated by AAA games. To better understand these patterns they have also analysed the games in each of the clusters. Also, they have further identified that the player interest in games is usually limited to 30-35 hours and for the majority of games, player churn can be observed within a few hours of a player’s gameplay.

Churn and migration behaviour of players have been investigated in the literature. Players moving from one game to another is considered as player migration. Three groups of player migration behaviour have been identified in the study of Sifa et al. [65]. The study has indicated that most of the players prefer to stay within the AAA games group and the players in the Platformer-group tend to stay within that genre. However, when it comes to the free-to-play and indie game players group, it has been identified that they tend to migrate to the AAA and platformer groups rather than migrating within the same game group.

Moreover, “cheater behaviour” of players in the Steam community has been investigated in the study of Blackburn et al., using data of 12 million players [63]. They have identified that cheater behaviour can be spread through friends.

The popularity of games is investigated in the work of Chambers et al., which is one of the earliest many-game analysis studies in the literature [32]. It is understood from the study that the distribution of game popularity follows a power law distribution indicating that choices of players are affected by the popularity of games. Chambers et al., have also investigated the short term and long term predictability of game server workloads based on player population of games [32]. The study shows that although the population of games is predictable over the short term the population trends show unpredictable long term fluctuations.

Baukhage et al., have also analysed playtime data of five games to understand how players lose interest in a game [66].

Trneny has investigated the factors affecting the success of a video game using game related data from 4600 games on Steam [67]. They have also attempted to predict a game's success based on descriptive game related data prior to release. The study identifies price, graphics, storage requirements, tags and multi-player support as some factors that affect the success of a game.

### **2.2.2 Approaches for other purposes**

Although a large number of many-game studies have focused on understanding player behaviour and games as presented earlier, several other studies have been conducted focusing on aspects that are also important to the digital game industry and related to player behaviour and games. The studies have investigated the possibility of game recommendation based on cross game preferences, the tag similarities of games, game reviews, game updates and early access games. More details on these studies are presented here.

The game recommendation process can be improved by understanding game preferences of players and game similarities. Sifa et al., have proposed two approaches of archetypal analysis for generating game recommendations based on the players' past playtime data across games [57]. Pathak et al., have proposed a personalized game bundle recommendation approach [68]. Studying games and their tags can

reveal similarities associated with tags. Yee has generated a tag similarity map to help visualize how tags are related using 2129 games in Steam [69]. Windleharth et al., have also analysed tags applied to games in Steam in order to categorize them through a conceptual analysis [70]. Game reviews have also been studied in the literature. Lin et al have analysed game reviews of 8025 games on the Steam platform to generate various insights [71]. Kang et al., have also analysed game reviews in Steam to analyse what factors influence the usefulness of a game review [72]. Lin et al., have analysed update information of 50 popular games on Steam [73]. They have identified that most games do not have a consistent update cycle. Lin et al., have studied early access games in Steam to generate insights [74]. The study reveals that developers update games more frequently while they are in the early access stage and game players tend to be more tolerant towards games in the early access stage. The study of Becker et al. have investigated the evolution of the Steam gaming community exploring the growth of the number players, the number of games and groups of the players [75].

### **2.2.3 Online tools displaying data and statistics of games**

Many-game research studies related to player behaviour analysis, game analysis and other purposes were presented in the previous subsections. Some online tools that display information about games can also be used for many-game analysis. There are a few online tools that display various data and statistics about games using the Steam platform as their data source. Specifically, these tools present information about games, their price, their population and various other data about games. These tools can be used as a visual aid to compare games and to keep up to date about games. These tools are introduced in this section as the study conducted in this thesis is also using Steam as the data source.

- **SteamSpy<sup>1</sup>:** SteamSpy is a platform that provides various data about games in Steam. The data and statistics provided in SteamSpy are based on data

---

<sup>1</sup><https://steamspy.com/>

continuously being collected from a random sample of Steam user profiles. SteamSpy provides estimations and charts about number of owners of a game as well as about total playtime of a game within the past 2 weeks and during the longer history. A geographical based breakdown of players and owners of a game, playtime distribution charts and user review charts of games are also provided. Meta data about games such as release date, developer, publisher and languages are also displayed. However, most of these information and visualizations are only available to registered SteamSpy users [76].

- **SteamCharts<sup>2</sup>:** SteamCharts is a website that displays charts of concurrent players of Steam games. It is purely focused on displaying the fluctuations of concurrent player counts of games to help anyone visualize trends in games [77].
- **SteamDB<sup>3</sup>:** SteamDB is another third-party tool providing insights about games in Steam [78]. SteamDB displays a variety of information about games including basic game related data such as genre, tags and release date, date of the last update, packages and bundles that include the game, downloadable content (DLC), achievements and statistics of the game. Moreover, it contains time series charts presenting price history, player counts, twitch viewers of each game. Apart from game related data much other information is provided.

## 2.3 Game Metrics

Game metrics are interpretable measures of anything related to games [20]. They can assist in the decision-making process related to games within the digital game industry. Drachen et al. have categorized game metrics to three types, namely, performance metrics, process metrics, and user metrics, expanding the categorization of Mellon [79]. Performance metrics are metrics used to measure how well the technical system of the game works which is related to evaluating the performance

---

<sup>2</sup><https://steamcharts.com/>

<sup>3</sup><https://steamdb.info/>

#### *2.4. Research Gap*

---

of the software. Some such metrics are the number of bugs per day and game server stability. Process metrics are the metrics used to monitor the game development process such as average turnaround time for new content development. User metrics are metrics related to the people who play games. These are categorized as gameplay metrics, community metrics and customer metrics [20]. Gameplay metrics are the metrics used to measure the activity of players within the game such as weapon use. Community metrics are metrics related to the interactions of the user within the game community. The customer metrics consist of all metrics that are related to the customer perspective of a user, which include metrics such as Daily Active Users and Average Revenue Per User. Some of the widely used metrics in the video game industry can be listed under these customer metrics. These metrics are used to measure monetization and to measure the player population [21]. The metrics used for monetization measures are Conversion Rate (players who convert to a paying customer), Average Revenue Per User (ARPU), Average Revenue Per Paying User (ARPPU), User Acquisition Cost (UAC)(the cost to attract new players) and Life Time Value (LTV)(the total revenue earned from a player throughout their lifetime). The metrics used to measure player population are Daily Active Users(DAU), Month Active Users (MAU) and Peak Concurrent Users (PCU). Keeping track of the number of players in games using these metrics are useful to understand the overall trends and popularity of the game [21]. In general, conducting game analytics with the help of metrics aids communicating in the decision-making process related to the game [80].

## **2.4 Research Gap**

Many-game studies can be considered as an instance of leveraging game data analytics to better understand players and games. As seen earlier, many-game studies generate insights to understand various aspects of player behaviour that are not limited to a specific game. Furthermore, more knowledge about games is also generated with such studies. Game analytic studies have been limited to a single or a

## *2.4. Research Gap*

---

few games limiting their applicability to better understand player behaviour across multiple games [9]. With the introduction of digital distribution platforms for video games, players have gained more access to games making it easier for the same player to join multiple games [58]. Hence, game analytic studies that are not limited to individual games are important to understand player behaviour across games [9]. It should be noted that many-game studies are not a replacement for individual game based game analytics. Individual game analytics are still important for the developers to analyse the data of their own games to make decisions. However, many-game studies are important to understand the common trends and player behavioural patterns observable among the wide gaming community. As observed earlier, current research studies in this area have focused on various objectives including player analysis such as game ownership, playtime, spending patterns, migration between games and even social structures. Furthermore, studies have also generated insights about games, such as what games are most popular and different player behaviour associated with different game genres including retention profiles. Hence many-game studies are helpful to understand common player behavioural patterns that could aid in better understanding player preferences and behaviour in various types of games. That knowledge can then be applied to the developer's own game and for new developers to better understand the general player community.

Due to the novelty of the domain of many-game studies there are still some aspects overlooked. Specifically, the player population behaviour of games in the presence of various external factors are not well investigated in the current literature. The number of players playing a given game changes throughout time as players leave the game temporarily or permanently, which is regarded as churn, and as new players join the game. Over the life time of a game such player population fluctuations can be observed. There are game metrics that are widely used in the digital game industry focused on the player population changes in games. Some of these are Daily Active Users (DAU), Monthly Active Users (MAU) and Peak Concurrent Users (PCU) [21]. Such metrics are useful to understand the overall

## *2.4. Research Gap*

---

trends and popularity of a game [21]. As the population of games are analysed on a per-game level using these metrics and as game companies rarely share their metric data publicly [80], currently there is limited knowledge regarding player population changes of games at a many-game level, the only exception being the study of Chambers et al. [32]. Hence, it is useful to investigate player population fluctuations of games to reveal insights regarding common population changing patterns as those can be used by game developers to understand common trends of player population changes. Player population could change in the presence of various external factors such as time of the day, day of the week, age of the game, special sale events or world crisis. Understanding how the player population change in the presence of such external factors in various games could help in better understanding the player community and their behaviour in games. Furthermore, it could provide insights regarding common player population changing patterns in the presence of common external factors that can be useful for game developers. However, currently such investigations are overlooked in the literature. Exceptions include the study of Chambers et al. [32]. The study depicts the importance of analysing player population changes at a many-game level for planning game hosting, especially when hosting games in a shared and on-demand infrastructure, and provides insights useful for game publishers and infrastructure providers. Analysing player population of 50 popular games the study reveals player population is predictable on a daily and weekly basis due to the existence of daily and weekly patterns which would be helpful for demand prediction. However, it has only identified that the population of the chosen games change in a similar pattern within a week, no thorough investigation about those patterns are conducted. Also, as only 50 popular games are used for that analysis its applicability across other games is not clear. Analysing player population traces of four games a population increase has been observed during the Christmas season and a population drop has been observed during the Sobig virus. This indicates the player population changes in the presence of external factors. However, no thorough investigation regarding this has been conducted in the study.

Hence, in this thesis player population fluctuations of games in the presence of external factors are investigated.

The next section presents a review of the literature directly related to the research problem addressed in this thesis. Specifically, the literature related to the external factors considered in this thesis are explored.

## 2.5 External Factors

As discussed in the previous section, not much attention has been given in the literature to investigate the game player population changes in the presence of external factors. Hence, in this thesis studies are conducted to explore the changes in game player populations in the presence of various external factors. Specifically, attention is given to short term temporal factors, namely, time of the day and day of the week, age of the game, and event-related factors such as sale events and the COVID-19 pandemic. These factors were chosen as the literature indicated that population or customer changes can be expected in the presence of such factors. The study of Pittman and GauthierDickey reveal that there is a time of day and day of the week effect on the population of 2 games [17]. SteamChart.com displays that player population of games fluctuate across time since the game release. Doucet has observed an increase in visitors to their game during the Steam Halloween sale event [81]. It has also been identified that time spent playing has been increased during the COVID-19 pandemic [36]. This section provides an introduction to these external factors and presents studies conducted related to those external factors in the game domain.

### 2.5.1 Studies on Short-Term Temporal Factors

This section provides an overview of the studies focused on short term temporal factors, namely, the time of the day and day of the week in the video games domain.

Game playing activities can be influenced by temporal factors. In fact, studies

from the literature report such observations. Pittman and GauthierDickey have investigated the daily player population cycles of the *World of Warcraft* and *Warhammer Online* game servers [17]. By analysing overlapped hourly population line charts for the seven days of the week, they have identified that players usually play more during evenings peaking at around 7 pm. Also, a higher population has been observed earlier in the day on weekends. Zhuang et al., have also observed a daily pattern in the player population of the *World of Warcraft* game [26]. Although such a time of the day effect has been identified they report that no significant pattern related to the day of the week could be identified in the game, which is different from the finding of Pittman and GauthierDickey [17]. Mahmassani et al., have also identified daily patterns in the *Everquest II* game [82]. It has been performed by investigating play session information of 552 players across four servers of the game whose geographical locations are suitable for timezone adjustments required for the analysis. The identified daily pattern has indicated a low number of sessions during the first half of the day until noon and a sharp increase afterwards followed by a sharp decrease after midnight. Furthermore, they have identified that the sharp rise within the day occurs earlier on Friday, Saturday and Sunday compared to the other days. Chambers et al., have also identified daily and weekly player population patterns by analysing the population data of 50 popular games collected from the GameSpy server [32]. Fast Fourier Transformation charts generated using population data for a one year period of games have been used. Peaks occurring at 24-hour cycles have indicated the existence of daily patterns in games. For some games, peaks have occurred at 1-week cycles indicating weekly patterns of population. Moreover, analysing the playtime dataset of 5 games, Bauckhage et al., [66] have revealed that, higher player activity occurs towards the weekend in two single-player games (*Just Cause 2* and *Tomb Raider: Underworld*). Especially in *Just Cause 2*, it is higher on Saturday. However, they claim that in general, the average playing activity of players seems to be similarly distributed during all days of the week among all 5 games they have studied.

These studies suggest that game player populations change in the presence of temporal factors such as time of the day and day of the week. However, most of the studies have only analysed a few games and the largest study among them has only used 50 games [32]. Hence, the implication of these findings across games is not yet clear. Moreover, when it comes to weekly patterns, studies only indicate that an increase in playing is observable over the weekend. However, if it is true for all games or if there are other patterns is something that can not be inferred from the current research. Hence, a thorough investigation on the daily and weekly pattern of games are conducted in this thesis in Chapter 4.

### 2.5.2 Studies on Product Life Cycles of Games

This thesis examines the player population change of games as the age of a game changes. In any product, changes in the number of customers(player population in games) or sales can be observed when the age of the product changes. The classic model of product life cycle represents four life stages a product goes through in its lifetime as depicted in Figure 2.1. This section presents prior work related to product life cycles in video games.

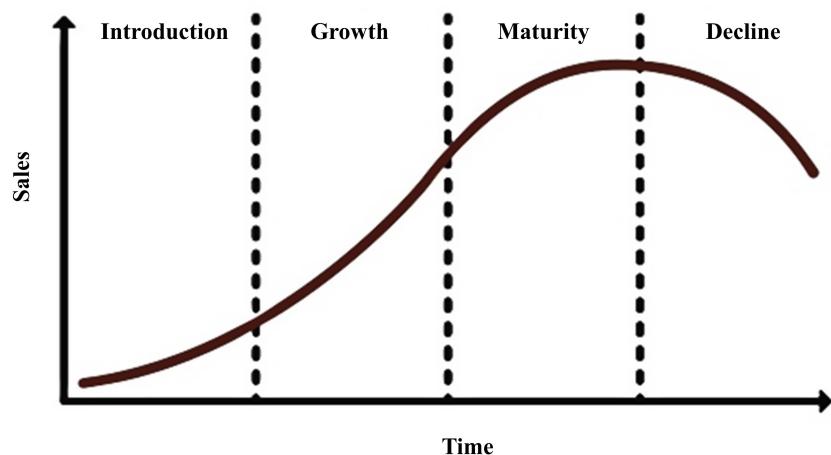


Figure 2.1: Product Life Cycle; Source: [1]

A digital game is also a product that has its own product life cycle. However, only a few research works have been performed related to product life cycle of games.

One key study that has investigated the product life cycle of games is the study of Cook [2] [83]. It has however, focused on game genre life cycle rather than individual games by considering genres as product categories. Moreover, the genre life cycle of the study is based on the number of games released, while the classical product life cycle is based on sales. As per the article, the shape of the genre life cycle of games is quite similar to the classical product life cycle. The five stages of a genre life cycle are Introduction, Growth, Maturity, Decline and Niche as depicted in Figure 2.2. However, it is mentioned that while a majority of genres follow this life cycle there could be exceptions as well.

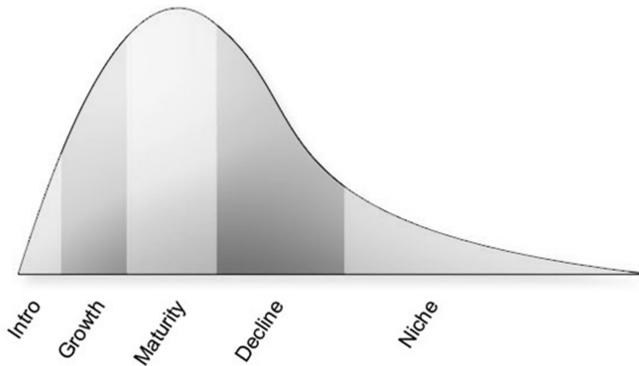


Figure 2.2: Game Genre Life Cycle; Source: Daniel Cook [2]

- **Introduction:** This is the introduction stage of a genre where the fundamental mechanics that represent the genre is introduced through the game.
- **Growth:** During the growth stage several other games could appear in the market. *Genre kings*, which are leading games of the genre could appear that draw increased attention to the genre. For instance, *Fortnite* in the Battle Royale genre.
- **Maturity:** In the maturity stage a few leading games of the genre control the market and set the standard. By this stage, game players are quite familiar with the genre. However, it is quite likely that only AAA games would survive in this stage due to the competitiveness requiring higher budgets.

- **Decline:** Decline of the genre happens when game players, on the whole, are no longer interested in the genre or due to the introduction of other newer genres. Thus, fewer games of the genre would be released during this stage.
- **Niche:** Once the genre is dead in the market it reaches the niche stage. There would only be re-releases for the existing audience or games solely released for non-profit purposes.

Gazecki proposes two shapes of game life cycles based on the number of players [3]. The shapes are introduced to explain the different metrics that can be used for game analytics during different stages of the lifetime of a game. The two proposed shapes represent viral games and games with better retention as depicted in Figure 2.3. Liu et al., have proposed a mobile application life cycle model for applications in the Apple app store by introducing daily application download ranking based milestones to the classical product life cycle model [84]. In their study it has been identified that on average mobile game applications observe 13 days of the introduction stage, 10 days of the growth stage and 25 days of the maturity stage and that entertainment applications have a shorter life cycle. Moreover, Draskovic et al., have conducted a case study using two mobile games, namely, *War Robots* and *Candy Crush Saga* to investigate the strategies of games applied during each stage of the product life cycle to increase the number of players and player retention [85]. Moreover, Sifa et al., have also identified four archetypal player retention profiles of Steam games using total playtime data of players [9]. One archetype represents games with short playtime and another shows a peak at 4 hours and a drop. The other two patterns both show a slow decreasing profile indicating the existence of players that have higher total playtimes. Although these archetypal shapes represent retention profile shapes, it is somewhat relevant to the product life cycle of games as it represents that there are games that are played for shorter periods and longer periods.

When it comes to game product life cycle, many of the previous studies have mainly focused on genre life cycle, retention related life cycle patterns and strategies

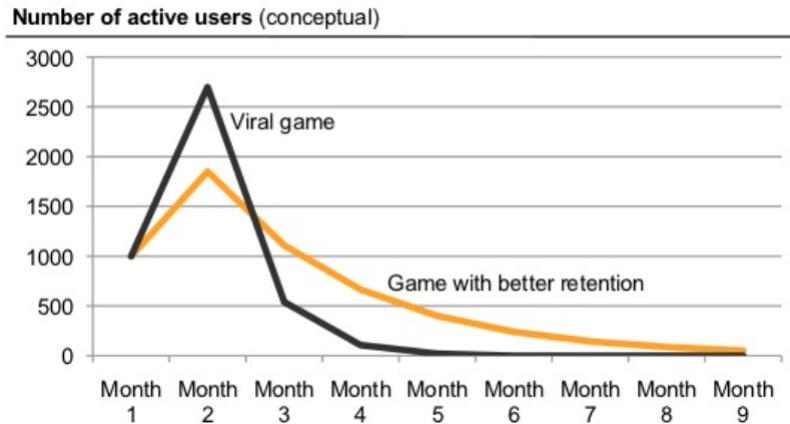


Figure 2.3: Game Life Cycles; Source: *HoneyTracks* [3]

applied in mobile games with respect to the classical product life cycle. Hence, it is evident from the literature that the shapes of game product life cycles have not been thoroughly investigated. Specifically, game product life cycle shapes could be identified based on player population time series of games. It could accurately indicate the life cycle of a game based on its popularity. However, past work has overlooked this. Furthermore, apart from the classic product life cycle shape there could be other shapes products display [86] [1]. Thus, it is important to further investigate games to identify if games also display diverse product life cycle shapes. Hence, this thesis investigates product life cycle shapes of games focusing on the age of game external factor in Chapter 5.

### 2.5.3 Studies on Sale Events

This section provides an introduction to Steam sale events and sales forecasting approaches in the literature focused on games.

#### 2.5.3.1 Steam Sale Events

Product discounts have long been regarded as important in attracting new customers and increasing sales. Hence, product discounts are offered to customers from time to time. Steam has also been offering discounts to attract new players to games in order to extend their product lifetime. Such discount offerings are considered as sale

events. Steam games can be purchased in various types of sale events. Mainly, two types of sale events are applicable to games. The first is sale events determined by the game developer/publisher. Three categories of such sale events exist, namely: Launch discounts which offers a discounted price for around seven days after game release; Custom discounts that can be given at any time or aligned with special events of the game such as updates or the anniversary; and Weeklong deals, that start on the Monday of a week and feature on Steam in the weeklong deals page and Steam news section [87]. The second type of sale events are those created by Steam itself. Steam selects the games for these sale events based on several criteria to select games that customers would be more interested in. Once selected the developers/publishers are notified. These sale events are featured on the Steam home page. Five categories of such sale events are offered, namely: Daily deals that last for 48 hours; Weekend deals from Thursday to Monday; Midweek Madness from Tuesday to Friday; Seasonal sales such as Summer sales and Winter sales; and Free weekends where temporary access is given to try the game without purchasing [87].

Moreover, due to the popularity of the Steam platform, sale events in Steam are popular among game players. Hence, participation in Steam sale events is important to attract new players for games, especially for indie games [88]. In fact the playtime of indie games are high during special seasonal sale events [89]. When it comes to sales, Steam has accounted for 86% of the sales of the game *A wizard's lizard*, which is higher compared to the percentage of sales originated from other websites that offer discounts such as Humble Bundle and Kickstarter [90]. Choi et al., have also investigated sales of games on Steam and have identified that the sales of discounted games are usually higher than the sales of non discounted games [30]. Additionally, they have revealed that both the discount percentage and the discounted price have a positive impact on the sales of a game. However, the sales of games are negatively impacted when there are more games sold on discounted price at the same time. Nonetheless, it is apparent that sale events in Steam are popular and contributes to the growth of games by attracting new players.

### 2.5.3.2 Video Game Sales Forecasting approaches

Several sales forecasting approaches related to games can be found in the literature. Ruohonen and Hyrynsalmi analysed the correlation between weekly video game sales and Google search volumes [91]. Sales data of 96 games have been collected from VGChartz and bivariate vector autoregression models have been created for each game. However, it has been revealed that, for the majority of the games, past internet search volumes have nearly no impact on the current sales of the game. Schaer et al., also investigated if Google searches, and shares of Youtube videos in social media platforms such as Facebook and Twitter can be used to improve video game sale forecasting [92]. Using weekly sales data of 78 games collected from VGChartz, they identified that univariate forecasting models based on past sales data alone perform better than linear models that use such online information. Moreover, no significant difference between the predictive power in different stages of a game's product life cycle has been identified. Rossetti et al., have shown that the SARIMA model is more reliable than exponential smoothing to forecast the sales of console games in the Italian market based on monthly sales data [93]. Guitart et al., found that the ARIMA model and generalized additive mixed models can provide higher accuracy in forecasting daily sales and total playtime of the games *Age of Ishtaria* and *Grand Sphere* while deep neural networks are also promising for the task [94]. Event details such as in-game events, in-game monetization and promotional events, marketing campaigns to attract new players, national holidays and temperature have also been used in the forecasting model either as 0 or 1 indicating the presence of the event or as a scale value when the event is associated with a value. SteamSpy<sup>4</sup> provides sales figures of Steam games by generating approximations based on the owned games information of a sample of Steam users. It is, however, an estimation of the current sales indicated by the number of owners rather than a forecast of the future.

---

The existing literature indicates that there is limited research conducted to fore-

<sup>4</sup><https://steamspy.com/>

cast sale event related player population or sales of games. Most of the presented game studies have investigated sales forecasting of games, but only one study has also used sale event related information in the forecasting process. However, that study is also limited to two games [94]. Due to the popularity of Steam sale events and frequent discounts offered in Steam, it is of great importance to investigate if the outcomes of such sale events of a game can be predicted beforehand. It could aid in planning the scheduling of sale events and discounts offered. Moreover, not all players who purchase a game at a discounted price from a sale event end up playing the game. Hence, in order to better estimate the expected players of a game due to a sale event, the focus needs to be given to the fluctuation of player population size rather than owners or sales of the game. Thus, this thesis presents a study conducted to introduce a forecasting model capable of predicting the maximum player population of a game during sale events in Chapter 6.

#### **2.5.4 Studies on the COVID-19 Pandemic**

This section provides an introduction to the COVID-19 pandemic and changes observed in the digital game industry during that period.

COVID-19 is recognized as a new form of the coronavirus that causes respiratory infections. It was first reported in China in December 2019 and it has spread around the globe since then causing a pandemic [95]. Although countries have taken various actions to control the spread of the virus, it has not been fully eradicated yet at the time of writing. This worldwide phenomenon has caused various disruptions to people's normal lifestyles [96]. The economic issues caused by the pandemic has resulted in a loss of employment. Also, in many countries, strict government regulations have made it mandatory for people to stay at home to control the virus spread, especially during the periods where the infection rates were uncontrollably high. Moreover, working from home and studying from home have become more common than ever during the pandemic.

With all the changes happening across the world during the pandemic, even the

## *2.5. External Factors*

---

digital games industry has observed changes in player behaviour. Several studies have observed an increase in sales of video games during the pandemic. Gameindustry.biz reported that game sales had increased by 63% during the peak pandemic period by analysing market data of 16 major game companies [35]. Moreover, they have observed a relatively higher increase in digital game downloads compared to the increase in physical game sales. Also, an increase in sales of gaming consoles has been observed. A survey conducted by Nielsen Games polling 3000 players reported that more than 45% of players in the US have spent more time playing video games during the first peak time of the pandemic than prior to COVID-19 [36]. Steam, the popular digital game distribution platform, reached a record of 20 million concurrent users on the 16th March 2020 during the pandemic [97]. Moreover, game streaming services have also observed a surge in viewership. Especially, in Twitch<sup>5</sup>, the popular game streaming platform, the daily viewership has doubled in the US as recognized by Nielsen [98]. Even the global viewership in Youtube Gaming<sup>6</sup> has increased [99]. Furthermore, the gaming industry has also spread awareness about social distancing during COVID-19 by contributing to the #PlayApartTogether campaign initiated by the World Health Organization [100]. Leading game companies such as Activision Blizzard, Riot Games, Unity Technologies and Twitch have joined this campaign offering special events, rewards and activities to promote people to play games and practise social distancing. Also, an educational game named *Can you save the world?* was released to enhance the awareness of children about social distancing [101]. Moreover, since gaming has increased, King et al., suggest that it is important to devise balanced approaches to game playing for long term well being [102]. Moreover, as per a survey based on *Pokemon Go* game players, intention to play location based games socially outside is reduced as the COVID-19 situation becomes severe [103]. Although the demand for digital games has positively increased during the pandemic, the industry has faced some negative impacts as well. This includes production delays of hardware, such as consoles and

---

<sup>5</sup><https://www.twitch.tv/>

<sup>6</sup><https://www.youtube.com/gaming>

also delays in releasing some games [104].

Since the COVID-19 pandemic is a very recent event it can be seen that there are only a few studies that have investigated the changes that have occurred in the gaming industry. However, understanding player behaviour during the pandemic, especially with respect to population changes is quite useful in order to be better prepared for such future crisis events. It would aid in determining resource allocation and predicting demand and preference for various games helping the gaming industry to thrive during such crises. Hence, a thorough investigation on player population changes during the onset of the COVID-19 pandemic is conducted in this thesis. It is conducted to generate insights about play patterns and to predict games that would observe a population increase during the pandemic in Chapter 7.

This section provided a review of the current literature related to the external factors considered in this thesis. Specifically, studies related to short-term temporal factors namely, time of the day and day of the week, product life cycles representing product changes related to the age of the game, sale events and COVID-19 pandemic were thoroughly reviewed discussing the current limitations. The subsequent section of this chapter provide the theoretical background for the work conducted in this thesis.

## **2.6 Background**

This section is aimed at providing a brief overview of time series analysis, forecasting and clustering as the work conducted in this thesis is using player population time series data. The trend and seasonality concepts presented in time series analysis section are used to pre-process the dataset used in the thesis and also to reveal short-term seasonal patterns in the data (Chapter 4). The time series forecasting section applies to Chapter 6 which presents a study related to predicting population. Concepts presented in time series clustering section is used in Chapter 4 and 5 to create clusters of similar population time series.

A time series is created by recording a value of a variable over a fixed interval [105]. In other words, it is a sequence of data points ordered in the time domain. For instance, daily concurrent players of a game or monthly sales of a game. Time series analysis, forecasting and clustering are core goals in time series studies. Time series analysis is focused on understanding the time series by extracting meaningful statistical properties. Time series forecasting is focused on generating models to accurately predict future observations of a time series. Time series clustering is focused on identifying groups of homogeneous time series from a time series dataset. Time series analysis, forecasting and clustering are described further in the following sections.

### **2.6.1 Time Series Analysis**

Time series analysis is the first step in tasks that involve time series. It is useful to get an initial understanding of the underlying data and its statistical properties. The two main properties of a time series are trend and seasonality [4].

#### **2.6.1.1 Trend**

A long-term increase or decrease of values in the time series would represent the existence of a trend in the series [106]. For instance, Figure 2.4 displays an increasing trend in airline passengers. Extracting the trend of a time series could aid in clearly visualizing a simplified view of the series to understand any overall increases or decreases in the trend. Several approaches can be used for trend extraction. Among them, moving average and polynomial fitting are primarily used.

**Moving Average:** Moving average is a non-parametric trend estimation approach that is sometimes used as a data smoothing technique. Here a value at a certain point of the series is approximated by calculating the mean of values within a chosen window surrounding that data point. Equation 2.1 presents this where  $m$  is the window size and  $m = 2k + 1$  and  $t$  is the index of the data point which is being replaced by moving average [106]. The chosen window size will determine the

smoothness of the extracted trend. A window size of 12 is used in Figure 2.4.

$$T_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j} \quad (2.1)$$

**Polynomial Fitting:** Polynomial fitting is a parametric approach for trend extraction. It is a least squares method that attempts to fit a parametric function minimizing the sum of squares of residuals [105]. Any order of polynomial function can be estimated to represent the trend. Polynomial of order 1 is used in Figure 2.4.

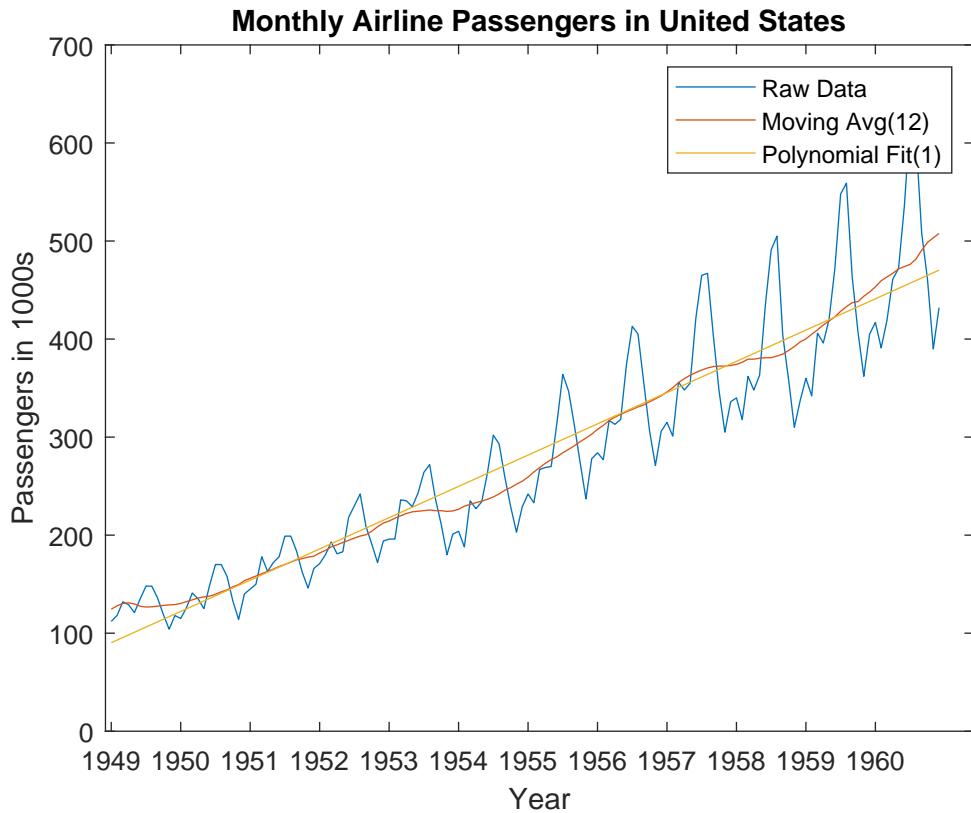


Figure 2.4: Monthly Airline Passengers in the United States [4]

### 2.6.1.2 Seasonality

A repeating pattern of fixed known periods can be identified as existence of seasonality in a time series [4]. Usually, seasonal patterns appear in time series if the series is impacted by seasonal factors such as day of the week or time of the year [106]. Moreover, seasonality generally refers to known fixed time periods such as weeks and years. For instance, the monthly airline passengers time series displays a yearly

seasonality as depicted in Figure 2.4.

### **2.6.2 Time Series Forecasting**

Time series forecasting is the process of modeling time series data to generate future prediction values. Traditionally, time series forecasting has been performed using statistical and mathematical approaches. Some of the most prominent approaches are Exponential smoothing, ARIMA models (Autoregressive Integrated Moving Average) and Regression models [106]. While these approaches are still prevailing neural network based approaches have also been proven to perform well in time series forecasting recently [107] [108]. Some of the commonly used neural network approaches for time series forecasting are as follows.

- **Long Short Term Memory (LSTM):** LSTM network is a type of Recurrent Neural Network (RNN) [109]. However, one key feature of LSTM is its capability to identify long term dependencies in the series compared to usual RNNs, hence the name long short term memory.
- **Nonlinear Autoregressive Neural Network (NAR) [110]:** NAR model forecasts the future values of a series based on the past values of a series. The neural network models the nonlinear relationship.
- **Nonlinear Autoregressive Exogenous Neural Network (NARX) [111]:** In NARX models future values of a time series are forecast using the past values of the same series and current and past values of the external series that have an influence over the considered series. The function that determines the connection between these variables is a nonlinear function which would be a neural network.

### **2.6.3 Time Series Clustering**

Clustering is a technique used in data mining to reveal patterns and groups hidden in data [112]. Time series clustering can be regarded as a specialized area in clustering

that is focused on the clustering of time series data. It is widely used in exploratory data analysis to find distinct patterns hidden in time series data. Generally, time series clustering uses the conventional clustering algorithms built for static data such as hierarchical methods and partitioning methods by either transforming time series data to static data form suitable for such algorithms or using distance measures suitable for time series data [113].

#### 2.6.3.1 Distance Measures

Distance measures are used to calculate the similarity/difference between time series. Distance measures are chosen based on the characteristics of the time series and the goal of the clustering process. Euclidean distance and Dynamic Time Warping are some of the common shape-based time series measures [114] which are also used in this thesis.

- **Euclidean Distance:** Euclidean Distance calculates point to point distance to measure the overall distance between two time series as per Equation 2.2 where  $X$  and  $Y$  are time series of length  $N$ . This measure can be used only if the time series are of similar length. Hence, it can be used when the objective of the clustering is based on the similarity of the values at each point of the time series [114].

$$EDist(X, Y) = \sqrt{\sum_{t=1}^N (x_t - y_t)^2} \quad (2.2)$$

- **Dynamic Time Warping:** Dynamic Time Warping is a commonly used algorithm to discover shape-based similarity between two time series that may vary in speeds and length [115]. DTW computes the optimal global alignment between two time series utilizing temporal warps, to aid better measure of the similarity between them. Point to point distance measures, such as Euclidean distance, would not always give the best sense of similarity between two time series that are quite similar in shape but contains slight distortions and delays

between them. However, DTW is capable of handling such time series.

DTW calculates the similarity between two time series by first identifying the best alignment between them. In order to better understand this concept, suppose there are two time series  $X$  and  $Y$ . First, a distance matrix  $D$  of size  $M \times N$  where  $M$  is the length of  $X$  and  $N$  is the length of  $Y$  has to be created. Next, this matrix is filled by calculating the distance between all possible data point pairs of the two series as in the matrix in Figure 2.5. The distance between two data points of  $X$  and  $Y$  is represented by  $d(X_i, Y_j)$  where;

$$d(X_i, Y_j) = |X_i - Y_j| \quad (2.3)$$

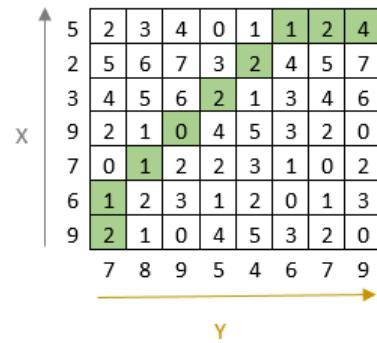


Figure 2.5: Distance Matrix of two time series;  $X$  and  $Y$ . The green color path depicts the optimal global alignment between the two series identified by Dynamic Time Warping

Finally, the best alignment of the two time series is recognized by discovering the warping path in the matrix that leads to the minimum distance between the two time series. It is the path that minimizes the warping cost given in Equation 2.4 [116].

$$DTW(X, Y) = \min\left\{ \sqrt{\sum_{k=1}^K w_k} \right\} \quad (2.4)$$

In Equation 2.4,  $w_k$  is the  $(i, j)_k$  element of matrix  $D$  that is also the  $k^{th}$  element of a warping path  $W$  within the matrix. The warping path is discovered by using the recursive function in Equation 2.5 [116].

$$\gamma(i, j) = d(X_i, Y_j) + \min \{ \gamma(i - 1, j - 1), \gamma(i - 1, j), \gamma(i, j - 1) \} \quad (2.5)$$

In Equation 2.5,  $\gamma(i, j)$  is the cumulative distance of the current cell  $(i, j)$  plus the minimum cumulative distance of the neighbouring cells. The equation also represents that the optimal alignment path can be found only by selecting the optimal next move, which is selecting between the neighbouring vertical, horizontal or diagonal cell with the minimum value and adding it to the path.

Continuing the example, the matrix in Figure 2.5 shows the optimal path identified that aligns the two time series  $X$  and  $Y$ . The data points in series  $X$  is aligned with data points in series  $Y$  using the identified optimal global alignment. For instance, the first data point of  $X$  which is 9 is aligned with the first data point of  $Y$ , which is 7. The second data point of  $X$  which is 6 is aligned with the first data point of  $Y$ , which is 7. Figure 2.6 depicts the two series before and after alignment. Furthermore from Figure 2.6, it can be seen that before DTW alignment the difference between the two time series are larger if measured with a point-to-point distance measure. On the other hand, after well aligning the two time series using DTW the point to point distance between them is lower.

### 2.6.3.2 Clustering Algorithms

As mentioned earlier, time series clustering usually uses the conventional clustering algorithms, such as partition methods and hierarchical methods, with different modifications. This thesis uses hierarchical clustering methods. Hence, it is explained as follows.

**Hierarchical Methods:** Hierarchical methods aim to assign data objects to a hierarchy of groups. These methods can be either agglomerative or divisive based on how the hierarchy is formed. The most common method is agglomerative (bottom-up), which treats each data object as separate clusters initially and merges each in an iterative fashion until all are merged to one or some other condition is met.

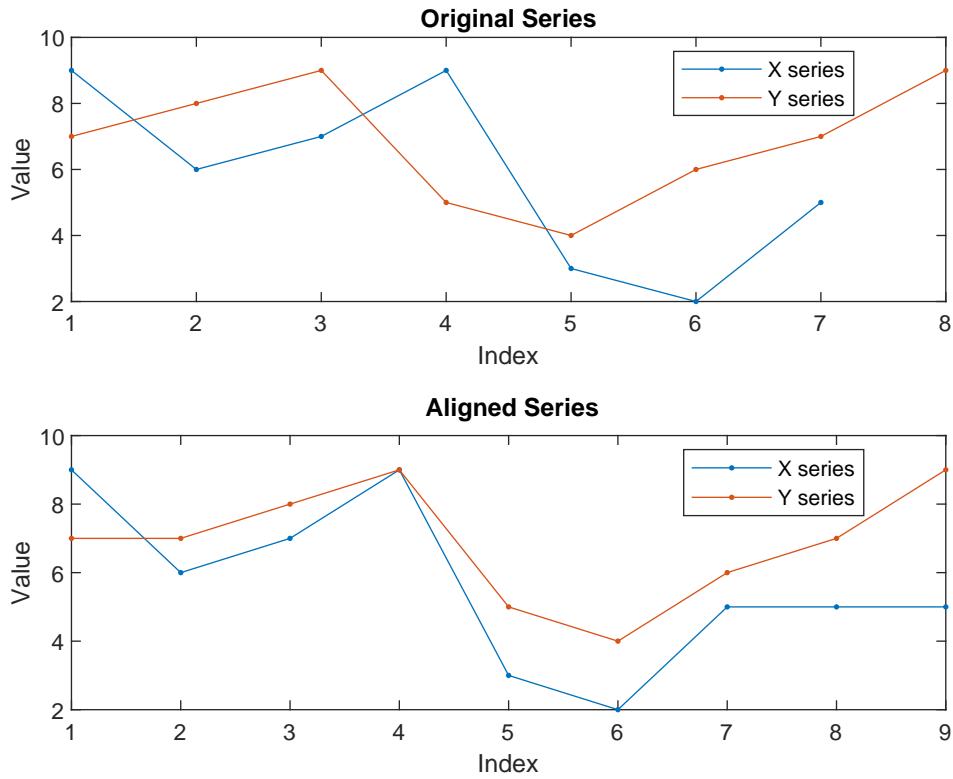


Figure 2.6: Before and After DTW alignment of Two Time series X and Y (Series are presented in continuous form rather than point form to better visualize the distance between the series)

In contrast, the divisive method (top-down) regards all data objects are in a single cluster initially and it recursively divides each cluster into smaller clusters until each object is in a single cluster or another condition is met.

Moreover, when forming clusters using the hierarchical method, linkage methods are used to determine how the clusters should be formed based on the distance between the data points in each cluster. Some of the linkage methods are as follows [117] ;

- **Single Linkage:** This is also known as the Nearest Neighbour method. The distance between two clusters is considered to be the smallest distance between any 2 data objects in the two clusters.
- **Complete Linkage:** This is also known as the Farthest neighbour method. As opposed to the single linkage, this uses the largest distance between any

two data objects in the two clusters to represent the distance between the two clusters.

- **Average Linkage:** This linkage method calculates the average distance between all pairs of data objects in the two clusters to determine the distance between two clusters.

Generally, hierarchical clustering results in a dendrogram, which is a tree-like representation of how clusters are connected hierarchically. Based on the requirements one can decide the level at which to cut the dendrogram to obtain the desired clustering of data objects [117].

### 2.6.3.3 Evaluation Measures

Cluster evaluation usually depends on the goal of clustering and hence there is no universally accepted method to determine if a certain cluster solution is good [117]. Nonetheless, several cluster evaluation approaches exist in the literature. The cluster evaluation approach used in this thesis is presented as follows.

In this thesis, cophenetic correlation coefficient is used to evaluate cluster solutions. Cophenetic correlation coefficient is a goodness-of-fit measure commonly used in hierarchical clustering to indicate how well the dendrogram represents the pairwise distances of the underlying data. Equation 2.6 presents the Cophenetic correlation coefficient. In that,  $D_{ij}$  represents the distance between  $i$  and  $j$  data objects in the distance matrix  $D$  which was used to build the dendrogram.  $\bar{D}$  is the average value in  $D$ . In the same way,  $T_{ij}$  represents the cophenetic distance in the dendrogram between  $i$  and  $j$  data objects. It is the height of the link at which these data objects are first joined together in the dendrogram.  $\bar{T}$  is the average value of  $T$  which has the heights of links connecting all pairs of data objects in the dendrogram.

$$c = \frac{\sum_{i < j} (D_{ij} - \bar{D})(T_{ij} - \bar{T})}{\sqrt{\sum_{i < j} (D_{ij} - \bar{D})^2 \sum_{i < j} (T_{ij} - \bar{T})^2}} \quad (2.6)$$

This section provided a brief introduction to time series analysis, forecasting and clustering and introduced various concepts and techniques associated with time series studies. More details about the concepts and techniques will be provided as required in the thesis chapters where those are applied.

## **2.7 Conclusion**

Game data analytics are becoming broadly used in the video game industry. Widespread use is expected as the worldwide gaming community is expanding, generating a large amount of data. Game data analytics assist in generating meaningful insights from the data related to games. However, publicly available game analytic studies that provide insights about player behaviour are limited to a single or few games which have limited the applicability of that knowledge across games [9]. With the popularity of digital game distribution platforms, such as Steam, players have more access to games and the same player owns multiple games [58]. However, the lack of many-game studies in the past has limited the understanding of the general player behaviour that is not restricted to individual games. Hence, several many-game studies have been conducted in the literature to generate insights that assist game developers to learn more about the general video game player community and their behaviour. As explained in this chapter, those studies have broadened the knowledge regarding games and players by providing insights related to game ownership, genre preferences, social structures within the game community, predictability of video game success, and playtime-based retention profiles of games. Although the existing many-game studies have provided such knowledge regarding players and games, investigations regarding player population changes in games have been overlooked in the many-game literature with an exception being the study of Chambers et al. [32]. It was identified that player population changes in games are constantly analysed and game metrics such as Daily Active Users, Monthly Active Users and Peak Concurrent Users are often used for the analysis [21]. Such analyses are helpful for the game developers and publishers to understand the popularity of the game,

how well the game retains players, forecasting server demand and determining the success of a promotion campaign in obtaining new players [21]. It was identified from the literature that changes in player population of games could be observed based on the time of the day, day of the week [17] and time since a game has been released as depicted in SteamCharts.com and also during sale events [81] and the COVID-19 pandemic [36]. However, since player population analyses are often conducted at an individual-game level by the game company or the developer, not much knowledge regarding the player population changes of games at a many-game level is available. The literature indicates that investigating player population changes at a many-game level is useful to provide insightful information to the current and upcoming game developers. For instance, to predict what type of games become popular [34] and to assist demand prediction when shared on-demand game hosting is used [32]. The study of Chambers et al. has explored the player population changes in the presence of time of the day and day of the week external factors to reveal the existence of common daily and weekly player population changes helpful for demand prediction in shared game servers [32]. Also, investigating player population changes in the presence of external factors is helpful to generate knowledge regarding the common player population patterns displayed by games, such as growth during sale events, which the game developers can then apply to their own games. Hence, this thesis explores the player population fluctuations in the games considering the influence of time of the day, day of the week, age of the game, sale events and world crisis COVID-19.

The next chapter presents the data collection process of the work presented in this thesis.

# Chapter 3

## Data Collection

The previous chapter reviewed the game data analytics literature and identified the current limitations. The research questions of this thesis were derived based on those identified limitations. This chapter presents the overview of the research methodology of the thesis and the details regarding data collection, which forms an essential part of the research conducted in this thesis.

The Steam game platform is chosen as the data source of this study as it is a leading digital distribution platform for video games widely popular among game players. The collected data includes player population data collected in different time intervals, price series and information about games such as genre, release date, tags, developer and publisher. The collected data is not only used for the study conducted in this thesis but also made publicly available in the Mendeley data repository<sup>1</sup> [118] for any researcher to use and to further advance knowledge in any related research discipline.

In this chapter, first the overview of the research methodology is presented. Then, the data source of the study, the Steam platform is introduced. Next, the specifics of the collected data and the procedures followed to harvest those are explained. Then, information regarding the public accessibility of the collected dataset for future research is provided. Finally, the chapter summary is provided.

---

<sup>1</sup><https://data.mendeley.com/datasets/yey3sy3vj2/1>

## 3.1 Research Methodology Overview

The research conducted in this thesis is focused on investigating the player population fluctuations in the presence of external factors. The external factors considered in this thesis are time of the day, day of the week, time since the game release, sale events and the COVID-19 pandemic. The research questions of the thesis, which are related to these external factors, were presented in Chapter 1. In order to address the research question of the thesis, four studies are conducted. Each study is focused on one of the external factors.

- **Study 1:** The first study of the thesis is focused on investigating the player population fluctuations in the presence of time of the day and day of the week. The study attempts to identify if player population fluctuations of games display daily or weekly recurring patterns and what patterns are displayed.
- *Study 2:* The second study is focused on investigating player population changes of games since the time of game release. It is aimed at identifying life cycle shapes of games based on the long term player population changes since game release.
- **Study 3:** The third study is focused on the player population fluctuations during sale events. It is aimed at generating a model to predict player population changes during sale events. It also uses the life cycle shapes revealed from Study 2 in model generation.
- **Study 4:** The last study is focused on the player population changes during the onset of the COVID-19 pandemic. It is aimed at revealing insights regarding population changes of games during the pandemic and predicting games that become popular during that period based on player population changes.

The overall methodology of the thesis including the data sources are depicted in Figure 3.1.

### 3.1. Research Methodology Overview

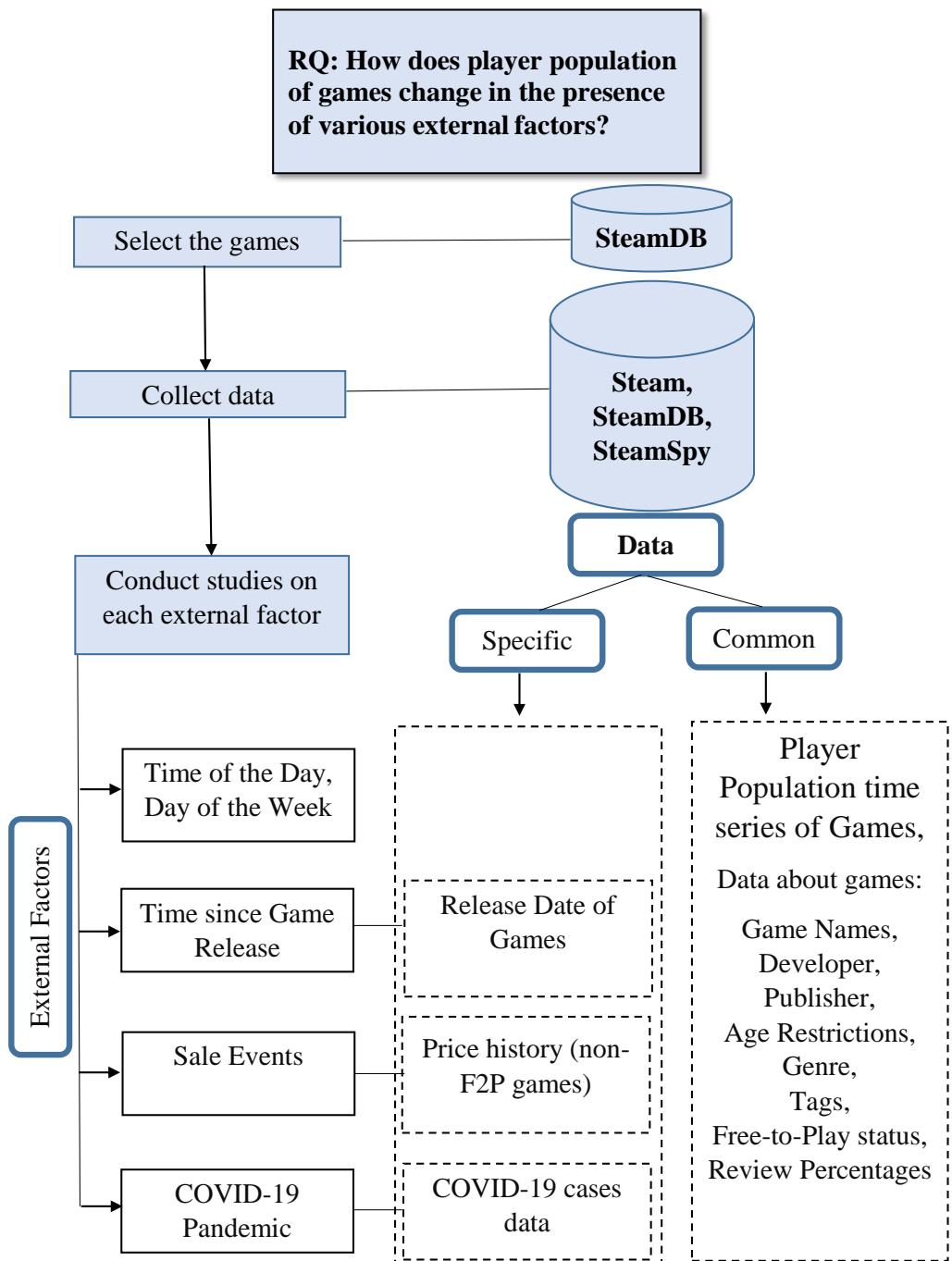


Figure 3.1: Research Methodology Overview; Data is categorized as common and specific in the figure to depict data that are specific to each study and commonly used by the four studies.

The initial step of the methodology is the selection of games. Game selection is conducted based on the player population statistics provided in SteamDB. Once

the games are selected, the data collection process is initiated. Required data and the sources to collect them is first identified in the data collection process. Whilst some data are commonly required for all four studies some data are specific to each study. These are depicted in Figure 3.1. The specifics of the data collection process, including game selection is presented in this chapter. The next step of the methodology is conducting the four studies related to the external factors. An overview of the studies was presented earlier. The detailed methodologies of each study are presented in the corresponding chapters. The studies are conducted in employing a quantitative research approach. Matlab and Python are used to conduct the data analysis, model generation and evaluations in the thesis. The rest of the chapter presents the details of the data collection process.

## **3.2 Steam Platform**

Steam is a popular digital distribution platform for video games developed by Valve Corporation. It has around 100 million<sup>2</sup> users and more than 30,000 games<sup>3</sup>. According to the 2018 year in review report by Steam [119] there have been 47 million daily active users, 90 million monthly active users and 1.6 million new purchases per month in the year 2018 alone. These numbers indicate the vast popularity of Steam in the game community. Steam provides a number of services to game developers and game players alike which has resulted in such wide popularity.

Steam is mainly a store-front for game players to purchase and download games. Games can be purchased directly from the Steam store or can be purchased from third-party sellers and activated through Steam. The Steam store contains both Free-to-Play games and non-Free-to-play games. Game players can purchase games for full price or discounted price through special offers or sale events. Steam provides Digital Rights Management (DRM) and verifies game ownership to protect against piracy. Moreover, Steam facilitates automatic updating of games. This assists the

---

<sup>2</sup><https://store.steampowered.com/about/>

<sup>3</sup><https://store.steampowered.com/search/?category1=998>

game developers to easily send updates and bug fixes of games and for the game owners to quickly get recent updates.

Steam provides various social features through the Steam Community that allows players to connect and discuss games. Some of the social features are steam chat, friend activity, player groups, group discussions and the most recent steam broadcasting to watch others play. Moreover, game hubs have been established to provide all the information about a game in a single place which includes reviews, news, discussions and videos. All these facilities are believed to provide a better gaming experience for players.

Game developers are also given various services through the Steam platform, especially through Steamworks<sup>4</sup>. Steamworks provides various tools that aid developers to further enhance their games and have a better experience in distributing their games through Steam. These include business tools to manage the games, marketing opportunities, gameplay features through the Steamworks API and several more. While providing such services to game developers and publishers Steam continues to be a major game distribution platform.

Steam is used as the data source for this thesis due to its widespread popularity. There are over 20 digital distribution platforms for PC games [120]. However, not all are widely used as Steam. Origin and GOG are also popular platforms. However, Origin by Electronic Arts had received some criticism and GOG only has retro games [120]. Hence, Steam was selected. As previously explained, Steam is widely popular among the game community including game players and developers alike. Since Steam provides thousands of games to millions of players this serves as a great platform that could be used to investigate player populations. Furthermore, Steam has provided an Application User Interfaces (API) to access various information related to games and players. Thus, acquiring the necessary data from Steam is a feasible task.

---

<sup>4</sup><https://partner.steamgames.com/>

### 3.3 Data Collection Approach

The initial step towards data collection is determining what data is required. In order to address the main research question of the thesis, which is “how does player population of games change in the presence of various external factors?” it is understood that game player population data, game-related data, and external factors related data are required. A set of games needs to be chosen for this research that is representative of the games people commonly play and is available on Steam. Moreover, in order to clearly observe player population fluctuations, the games need to be chosen based on the magnitude of the games’ player base. For this purpose, a third-party tool named SteamDB<sup>5</sup> which provides game statistics was used to retrieve a list of games ranked based on their number of players. The top 2000 games from the list on the 11th December 2017, which would be referred to as *Gameset1* and the top 1350 from the list on 9th September 2019, which would be referred to as *Gameset2* were chosen. These contain the top games based on the player population. The top 2000 and the top 1350 games were chosen because if a smaller number such as 100 is chosen only the most popular games would be included and if a larger number, such as all the games in the list provided by SteamDB, is chosen even the games that have only one player would be included. Hence, intermediate numbers were chosen making sure that the games included in the game sets have at least 50 players to observe player population fluctuations. The key difference between these two game sets is the frequency of the population data collection and the availability of population data since game release which is explained later in the chapter. Furthermore, non-games such as software, DLC (downloadable contents) and demos were removed and not used in the study. Hence, after removal *Gameset1* consisted of 1963 games and *Gameset2* consisted of 1260 games. Moreover, there are 852 games common between the two sets, which is 43% and 67% of games in *Gameset1* and *Gameset2* respectively. Each chapter uses a subset of games in these two game sets based on the respective objectives of the studies presented in the

---

<sup>5</sup><https://steamdb.info/>

chapters.

Once the games are chosen the data collection process can be started. The procedure for collecting the various data required for this research is as follows.

1. **Player Population of Games:** The player population data (in other words, the number of players) of each chosen game needs to be collected to investigate population fluctuations. As mentioned earlier, *Gameset1* was initially created using the top 2000 Steam applications list obtained from SteamDB. In order to collect player population data of games, a data collection frequency needs to be chosen. As the first study of the thesis focuses on the daily population patterns, population data should be collected in intervals less than 24 hours, such as per hour and minute. Collecting data in short intervals records more population fluctuations which is helpful to closely analyse population changes if needed but, it is resource-intensive as it takes more storage and more API calls. Furthermore, since the population data is collected in real-time, rather than obtaining historical data, it is not possible to get past population data in a higher frequency than the collected frequency in case it would be required later for analysis. Considering these concerns, it was decided to collect player population data at two different frequencies. Player population of the games in the top 1000 applications in the list used to create *Gameset1* were collected at 5 minute intervals as it contains the most popular games. This is helpful to more closely examine fluctuations of the most popular games if needed. The population of games in the next 1000 applications of the same list were collected at 1 hour intervals. This was collected in 1 hour intervals rather than 5 minute intervals due to storage limitations. In essence, the *Gameset1* consists of 1963 games (after removal of non-games from the 2000 initially selected) and player population data of the first 982 games (after removal of non-games from the first 1000) were collected in 5 minute interval and player population data of the next 981 games (after removal of non-games from the next 1000) were collected in 1 hour interval. Player population data were

### *3.3. Data Collection Approach*

---

collected using the Steam API, specifically, using the *GetNumberOfCurrentPlayers*<sup>6</sup> method passing application ID of the game. The population data harvesting process started on the 14th December 2017 and continued until 12th August 2020. Furthermore, the data collection process was duplicated in order to reduce missing data that could occur due to power failures or server unavailability. To this end, a remote server and several computers with local servers (XAMPP, a cross-platform web server solution package consisting of Apache server, MariaDB, Php, and Perl) were set up to send data requests in parallel. Then data fusion was performed to combine the data to generate the final population dataset.

Daily player population data of games in the *Gameset2* were downloaded from SteamDB. SteamDB has player population data of games since the release date. Their player population data collection process was started in 2015 and the data prior to that have been extracted from internet archives. Since player population data since the release date of games is needed to address some objectives of this research those were collected from SteamDB. The population data collection process of the thesis was started on the 14th December 2017 as previously mentioned. The API provided by Steam for collecting player population data is limited to collecting real-time player population of games rather than historical data. Hence, in order to obtain player population data of games since the release date, rather than only from 14th December 2017, it was necessary to rely on SteamDB as it has historical player population data as well. Although SteamDB has population data since release of a game, the data they have made available only contains population data on a daily basis. Hence, instead of completely relying on SteamDB our own data collection process was also set up as earlier explained to collect the population data of *Gameset1* in intervals of 5 minutes and 1 hour. Furthermore, since web scraping is prohibited in SteamDB, the daily population data files were manually

---

<sup>6</sup><https://api.steampowered.com/ISteamUserStats/GetNumberOfCurrentPlayers/v1/?appid=>

### *3.3. Data Collection Approach*

---

downloaded from each game’s page. This harvesting process resulted in daily population data of the games in *Gameset2* from their release dates until the 9th September 2019.

2. **Game Names:** Game names were extracted using the *appdetails*<sup>7</sup> method of the Storefront API<sup>8</sup>.
3. **Release Date:** Game release dates were also extracted using *appdetails* method of the Storefront API.
4. **Developer, Publisher:** Developer and Publisher names were also extracted using *appdetails* method of the Storefront API.
5. **Age Restrictions:** In order to identify the appropriate age group of a given game, age requirements were needed to be extracted. Hence, attention was given to content classification ratings. Various countries have their own content rating system to classify movies, games and others in order to provide advice to consumers about the content. This process assigns an age restriction to games based on the content as depicted in Table 3.1. Thus, it is a good source to understand the age requirements of various games. The Entertainment Software Rating Board (ESBR)<sup>9</sup> is an American organization that classifies video games and applications based on age and content. An age restriction extraction process was conducted using the ESBR rating system.

Class	Meaning
E	Everyone
E10+	Everyone10+ (Ages 10 and up)
T	Teen (Ages 13 and up)
MA17+	Mature17+ (Ages 17 and up)
Ao	Adults Only 18+ (Ages 18 and up)

Table 3.1: Entertainment Software Rating Board (ESBR) Game Ratings

6. **Genre:** Genres of games were extracted using the *appdetails* method.

---

<sup>7</sup>(<http://store.steampowered.com/api/appdetails/>)

<sup>8</sup><https://wiki.teamfortress.com/wiki/User:RJackson/StorefrontAPI>

<sup>9</sup><https://www.esrb.org/>

7. **Tags:** Apart from genres games can also be described using tags. Moreover, while genres are more generic tags can be quite descriptive. Hence, tags were also collected during the data collection process. The Steam page of each game displays a list of at most 20 tags assigned to the game by players. These can be extracted by scraping each game's page in the Steam store. However, along with the tags, the number of players who have applied each tag to the game needs to be retrieved. Hence, the SteamSpy API <sup>10</sup> was used to extract the top 20 tags of each game along with the number of players who have applied each tag.
8. **Free-to-play status:** Games in Steam are either Free-to-Play games that can be played for free or non-Free-to-Play which have to be purchased. The Free-to-Play status of games was collected by scraping each game's page in the Steam store. Furthermore, cross-checking was conducted by investigating the existence of Free-to-play tag in the list of tags of each game.
9. **Reviews:** The positive review count and negative review count of each game presented in the game's home page in Steam store as of 29th January 2020 was collected by scraping each game's store page in Steam.
10. **Price Details:** The price of non-Free-to-play games in Steam changes from time to time due to special promotions and sale events. The price details of games in *Gameset1* were collected with a daily frequency. The appdetails method of Storefront API was used with price overview filter to collect the initial price, final price and discount percentage. Price details were collected from 07th April 2019 until 12th August 2020. Furthermore, the price details of the games in *Gameset2* were manually downloaded from SteamDB to obtain historical price details. US price history of games were recorded in USD as the majority of Steam users are located in the US [121].
11. **Covid-19 Data:** Novel Coronavirus (Covid-19) cases data were directly ob-

---

<sup>10</sup><https://steamspy.com/api.php>

tained from the dataset<sup>11</sup> made publicly available by the Center for Systems Science and Engineering at John Hopkins University

Throughout this thesis subsets of these collected data are used in each chapter based on research objectives. *Gameset1* and *Gameset2* are not used together in any of the studies. The subset selection process will be explained in detail in each chapter. To provide a brief overview of the subsets, Chapter 4 will be using games in *Gameset1* and player population data from the first 6 months of those games. Chapter 5 will be using the player population data of the first three years after game release of games in *Gameset2*. Chapter 6 will be using the non-Free-to-play games of *Gameset2* and Chapter 7 would be using games in *Gameset1*.

Table 3.2 depicts the properties of the games in *Gameset1* and *Gameset2*. Note that the release date of games could be a date later than the data collection start date for some games in *Gameset1* as some games are released in early access mode and the final release date then becomes a later date.

## **3.4 Data Contribution**

The collected dataset of *Gameset1* is made publicly available to contribute to strengthening future research in gaming. Player population data, price data and game-related metadata are all available in the Mendeley Data repository<sup>12</sup> [118], which is a prominent research data repository. The game-related metadata are release date, free to play status, developers, publishers, supported languages, genre and tags. Our player population data are collected in shorter intervals over 2 years and 8 months providing rich player population time series. Furthermore, to the best of our knowledge, no public dataset exists that contains player population data sampled at such high frequency. Even SteamDB and SteamCharts, the third-party tools/websites that provide insights about Steam games, have not made their population data publicly available at such high frequency as ours. Moreover, the game

---

<sup>11</sup><https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases>

<sup>12</sup><https://data.mendeley.com/datasets/yct3sy3vzj2/1>

### 3.4. Data Contribution

---

	<i>Gameset1</i>	<i>Gameset2</i>
<b>Number of Games</b>	1963	1260
<b>Population Data Sampling Frequency</b>	5 minutes and 1 hour	daily
<b>Period Covered</b>	14th Dec 2017 to 12th Aug 2020	Since Release Date of each game until 9th Sep 2019
<b>Median of Mean Daily Population</b>	79	512
<b>Median Absolute Deviation (MAD) of Mean Daily Population</b>	64	361
<b>Median of Release Year</b>	2015	2016
<b>Number of Unique Tags</b>	339	348
<b>Top 12 Genres</b>	Action, Indie, Adventure, Strategy, RPG, Simulation, Casual, Free to Play, Massively Multiplayer, Sports, Early Access, Racing	Action, Indie, Strategy, RPG, Simulation, Adventure, Free to Play, Casual, Massively Multiplayer, Early Access, Sports, Racing
<b>Top 10 Publishers</b>	Ubisoft, SEGA, Feral Interactive, 2K, Square Enix, Bethesda Softworks, Warner Bros. Interactive Entertainment, Paradox Interactive, Valve, Activision	SEGA, Ubisoft, 2K, Square Enix, Bethesda Softworks, BANDAI NAMCO Entertainment, Paradox Interactive, Valve, Feral Interactive, Electronic Arts
<b>Top 10 Developers</b>	Feral Interactive, Valve, Aspyr, Square Enix, Koei Tecmo Games, Firaxis Games, Traveller's Tales, Creative Assembly, Relic Entertainment, Telltale Games	Feral Interactive, Valve, Square Enix, Firaxis Games, Creative Assembly, Koei Tecmo Games, Ubisoft Montreal, Relic Entertainment, Sports Interactive, Paradox Development Studio

Table 3.2: Properties of *Gameset1* and *Gameset2*

related data and price information that is made available along with population data is also useful for future research, not only in the domain of digital games but also in other areas such as time series analysis and forecasting.

## **3.5 Chapter Summary**

This chapter presented the data collection procedure of the thesis and the research methodology overview.

First, the research methodology overview of the thesis was presented. Then, the main data source, the Steam platform was introduced. Also, the data collected for this research and the procedure of collection were described for re-usability. Specifically, the collected data includes player population series, price series and data about games such as genre, tags, release date, publisher and developers. Several web APIs and web scraping approaches were used to collect these data.

Two sets of games namely, *Gameset1* and *Gameset2* were introduced. The main difference between those that is important for this research is the length of the population series and the data collection frequency. The population data of games in *Gameset1* were collected over a fixed period of 2 years and 8 months in 5 minutes and 1 hour intervals while daily player population data since the release date of each game of *Gameset2* were downloaded from SteamDB. Subsets of these game sets will be used throughout the next chapters selected based on the chapter objectives.

Furthermore, the population data, price history and other game-related information of games in *Gameset1* was made publicly available in the Mendeley data repository. Since currently, there is not any such publicly accessible dataset of player population and price time series this contribution may be a significant asset to the research community.

The next chapter provides the first technical work of the thesis which is focused on short-term seasonality of player population fluctuations. It employs the dataset resulted from the data collection procedure presented in this chapter.

# Chapter 4

## Short Term Seasonality in Player Population Fluctuations

Parts of the work reported in this chapter have been published in the following research paper;

1. **Dulakshi Vihanga**, Michael Barlow, Erandi Lakshika and Kathryn Kasmarik, “Weekly Seasonal Player Population Patterns in Online Games: A Time Series Clustering Approach,” 2019 in *IEEE Conference on Games (CoG)*, London, United Kingdom, 2019, pp. 1-8.

The previous chapter presented the data collection and the research methodology overview of this thesis. Using the collected data, this chapter presents the study conducted to investigate short term seasonality, namely, daily and weekly patterns in player population fluctuations to understand the population changes in the presence of temporal factors, such as time of the day and day of the week.

As presented in Chapter 2, the few existing studies that have focused on daily and weekly patterns of game playing activity are limited to a single [17] [26] [82] or a few games [32]. Hence, the insights generated from those studies are not suitable to be generalized across many games. Moreover, while the studies imply that a higher playing activity occurs across weekends, whether all games display that pattern or if any other weekly patterns exist is not thoroughly investigated. Hence, a thorough investigation on daily and weekly patterns of player population fluctuations of games are conducted in this chapter. Game developers can identify the daily and weekly

---

patterns the players of their game displays from the data collected in their game. However, there is less knowledge regarding daily and weekly population patterns at many-game level. Chambers et al. have demonstrated that investigating daily and weekly population patterns at a many-game level is important to determine the predictability of game hosting server workload [32]. It is important when games are hosted in shared on-demand infrastructures [32]. Furthermore, it is also useful for upcoming developers who do not have their own player-related data, to learn how player population changes in various types of games during a day and a week. Hence, this chapter conducts a study to address the question of “How does player population of games fluctuate during a day and a week?”. It is hypothesized that games display daily and weekly population patterns and there can be more than one weekly population pattern that can be identified from games.

In this chapter, an investigation of daily and weekly patterns of player population is conducted using a dataset of player population data of 1963 Steam games collected over a 6 months period, which is a subset of *Gameset1*. The existence of short term seasonality in player population fluctuations is investigated using an Auto Correlation Function to identify if all games display daily or weekly patterns. Several trend removal approaches are used to enhance the recognition of seasonality. Furthermore, a time series based cluster analysis is conducted to identify archetypal weekly patterns games display, their frequencies, and what types of games are associated with each archetype. A Dynamic Time Warping (DTW) approach is used to generate archetypal weekly patterns. Tags, age requirement and overall population size of games are investigated to identify what types of games each archetype is associated with.

The key findings of the study indicate that 68% of games display daily patterns and 77% of games display weekly patterns of player population fluctuation. Furthermore, 9 archetypal patterns of weekly player population changes were identified along with the percentage of games displaying each pattern. Moreover, several insights were generated by analysing the characteristics of games associated with each

pattern.

This chapter first provides an overview of the research procedure of the study. Thereafter, the work related to the investigation of the existence of short term seasonality in population fluctuations is presented. Subsequently, the procedure of archetypal weekly pattern extraction is presented. Finally, the chapter conclusion is provided.

## 4.1 Research Procedure

The study conducted in this chapter is aimed at investigating the existence of short term seasonality in player population fluctuation. To serve this purpose, the study consists of two main investigations. The first investigation is focused on revealing the games that display daily and weekly patterns in player population fluctuations. It is conducted by incorporating autocorrelation functions and several trend removal approaches. The methodology of this investigation and its outcomes are provided in Section 4.2. The second investigation is focused on revealing the archetypal weekly patterns games display. For this purpose, the games that were identified as displaying weekly seasonality from the first investigation are used. Archetypal patterns are revealed utilizing hierarchical clustering and dynamic time warping approaches. The methodology of this investigation is depicted in Section 4.3. An overview of the research procedure is depicted in Figure 4.1. The data collection and preprocessing methods of this study are explained as follows.

### 4.1.1 Data Collection and Preprocessing

A subset of the population data of games in *Gameset1*, introduced in Chapter 3 is used in this study. To recall, *Gameset1* consists of the 1963 games selected from the the 2000 applications with the highest player population within the last 24 hours on 11th December 2017 after excluding all non-game applications. These games were chosen as it is vital for the study to select games that have a strong player base as

#### *4.1. Research Procedure*

---

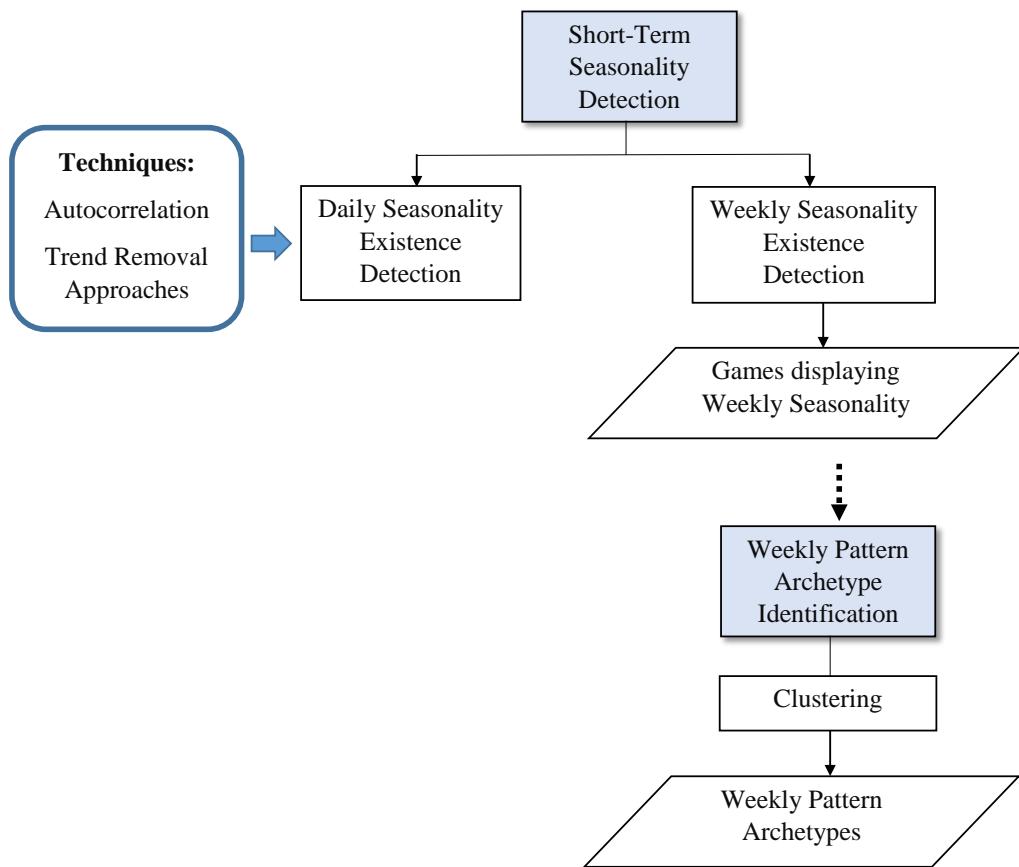


Figure 4.1: Research Procedure Overview

indicated by a high average number of players in order to determine the existence of seasonality. Player population data of the 1963 games from 14th December 2017 to 13th June 2018, which covers a time stretch of 6 months, is used in this study. Moreover, the player population dataset consists of the current player population size recorded in 5 minutes interval for 982 games (first half of the selected list of games) and in 1 hour intervals for the rest of the games.

Missing data handling and data smoothing was conducted as a data preprocessing step. Due to Steam server connectivity failures, network failures or other reasons there are missing data in the population time series. The median value of the percentage of missing data in each game over the selected 6 months duration was 1.8%. Furthermore, the median value of the percentage of the longest missing data sequence of each game was 1%. Moreover, since the study is focused on pattern identification, a smoothing process is required to emphasize the underlying

patterns. Hence, to address both missing data and data smoothing, the median filtering procedure was applied over the population dataset. Median Filtering is a data smoothing technique that replaces a data point with the median of its neighbouring data points [122]. The window size, which determines the number of neighbour data points to be used can be specified based on the requirements. A window size that is suitable for imputing missing data and that does not over-smooth data has to be selected to preserve the underlying patterns of the data. Hence, after exploring multiple values, a window size of 7 was chosen. In the rest of the chapter, the smoothed data will be addressed as follows when separate addressing to the dataset is required:

- *5mData* : Player population data of 982 games collected at 5 minutes intervals
- *60mData*: Player population data of 981 games collected at 60 minutes (1 hour) intervals

## **4.2 Existence of Short Term Seasonality in Player Population Fluctuations**

This section presents the methodology and outcomes related to identifying games that display daily and weekly seasonality in player population fluctuations.

### **4.2.1 Autocorrelation based Seasonality Detection**

Seasonal pattern identification is a sought-after preliminary step in most of the time series analysis and forecasting tasks [106]. As explained in Chapter 2, time series analysis is the process of analysing time series data, such as hourly rainfall series, to identify and explain underlying statistical properties and structures of the series such as trend and seasonality [4]. This process can be conducted using several available techniques such as observing original data plots, Fast Fourier Transformation (FFT) based methods [123] and autocorrelation, depending on the overall goal and the

nature of the dataset. However, observing original data plots is time-consuming as there are many games in the dataset used in this study. Also, the player population time series is transformed from the time domain to the frequency domain when an FFT based method is used for seasonality detection [124]. However, no such transformation is required for the autocorrelation method as the population series are already in the time domain. Hence, in order to identify short term seasonal patterns in the player population dataset the autocorrelation technique was used.

*Autocorrelation* is defined as the correlation of a variable with a lagged version of itself [4]. It is also known as *serial correlation*. The function that calculates autocorrelation is given in Equation 4.1 [125].

$$r_k = \frac{c_k}{c_0} \quad (4.1)$$

where;

$$c_k = \frac{1}{T} \sum_{t=1}^{T-k} (y_t - \bar{y})(y_{t+k} - \bar{y}) \quad (4.2)$$

In Equation 4.1,  $r_k$  measures the autocorrelation of lag  $k$ . Lag  $k$  indicates the time series  $y$  lagged by  $k$  time steps. The length of the time series is given by  $T$  and the sample variance of the time series is represented by  $c_0$ . Also,  $\bar{y}$  represents the overall mean. The value of autocorrelation is always between +1 and -1 [4]. Thus, autocorrelation could be positive as well as negative in different situations.

#### 4.2.1.1 Preliminary Analysis with Autocorrelation Function Plots

Autocorrelation Function plots (ACF plots) present the autocorrelation values of a time series up to a selected lag. It can be used to visually determine the seasonality of a time series [106]. If the data is inherently seasonal, the autocorrelation values for seasonal lags would be larger [106] in the ACF plot compared to other lags. The population time series of a representative set of games from the dataset and their corresponding autocorrelation function plots are presented in this section to further explain the autocorrelation based seasonality detection process. Autocorrelation of

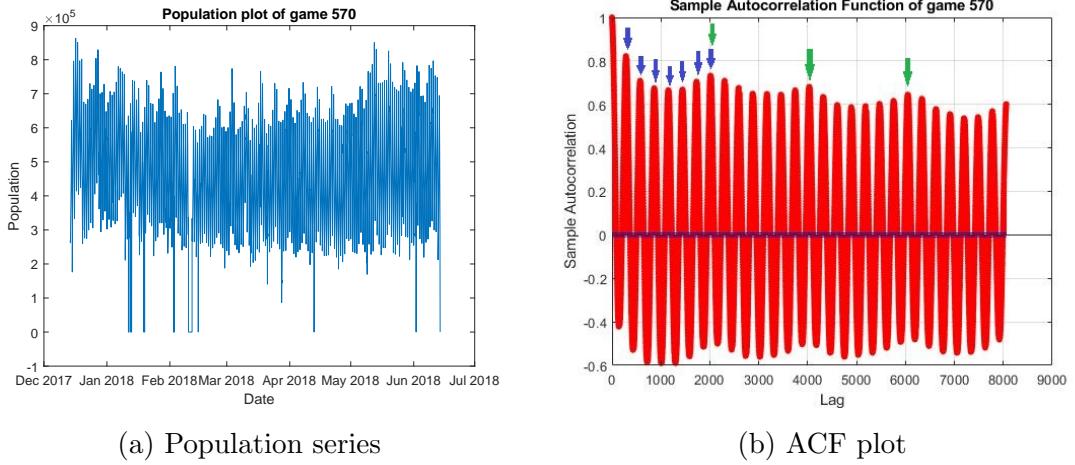


Figure 4.2: Population series and Autocorrelation function plot of *DOTA 2* game; blue arrows in the ACF plot points to the lags that are multiples of a day’s lag. Green arrows points to the lags that are multiples of a week’s lag.

up to 4 weeks of lag are presented in the plot. This means, autocorrelation for all lag until lag 8064 for *5mData* and until lag 672 for *60mData*. Furthermore, due to random variations, autocorrelation usually would not always be zero even when there is no correlation. However, the values will be closer to zero. The blue lines in ACF plots help to determine this bound where values that lie outside this confidence bound can be regarded as significantly different from zero representing significant autocorrelation [106].

Figure 4.2 presents the 6 months population time series of the *DOTA 2* game and its autocorrelation function plot. It could be observed that autocorrelation values are higher at lags that are multiples of 2016. This lag number represents a lag of a week for games in *5mData* (from which *DOTA2* is drawn). Hence, it can be understood that *DOTA 2* displays weekly seasonality. Moreover, it could also be observed that there are peaks at every 288th lag representing a day. This could indicate the existence of daily cycles. For clarity, a zoomed version of the series is presented in Figure 4.3. It can be seen in the population series that the same pattern of population variation repeats daily where the population gradually increases and decreases within the 24 hours of a day.

A long-term increase or decrease of values in the time series is identified as a trend in the series [106]. Some player population series exhibit trends which could

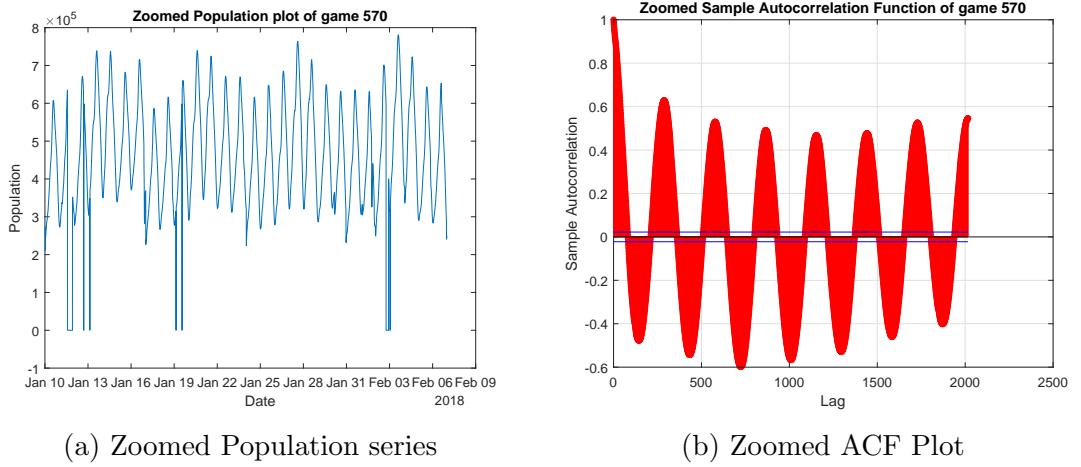


Figure 4.3: Zoomed Autocorrelation function plot of *DOTA 2* game

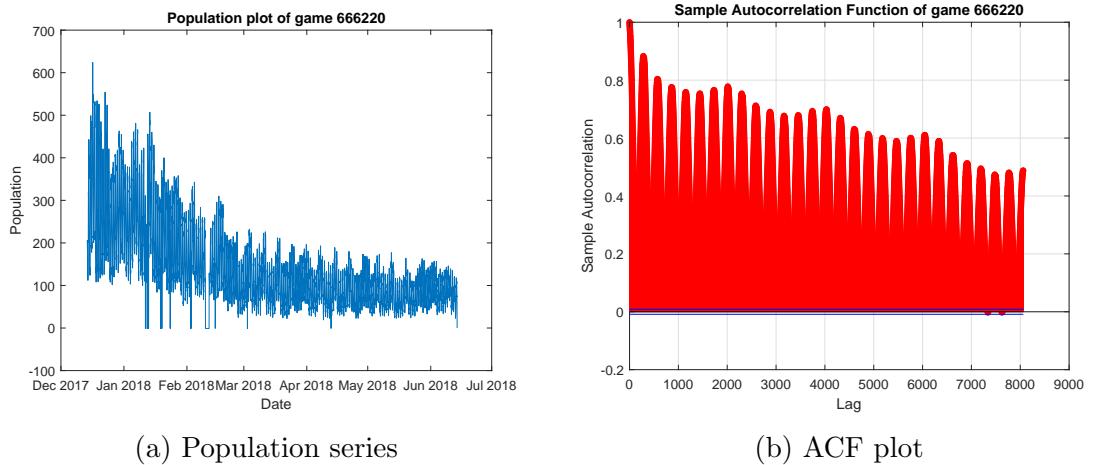


Figure 4.4: Population series and Autocorrelation function plot of the *CS2D* game

obstruct direct identification of seasonality using autocorrelation. A few such examples can be seen in Figure 4.4, 4.5 and 4.6. When the population series has trends, those appear in the ACF plots as well. Hence, for accurate seasonality detection, it is recommended in the literature to remove trends from the time series prior to seasonality detection. The trend removal process conducted in the study is presented in Section 4.2.2. The systematic approach for seasonality detection conducted in the study is presented in the next section.

#### 4.2.1.2 Seasonality Detection Procedure

The main hypothesis behind the seasonality detection procedure is that if a game displays a daily or weekly pattern, its autocorrelation value should be highest at

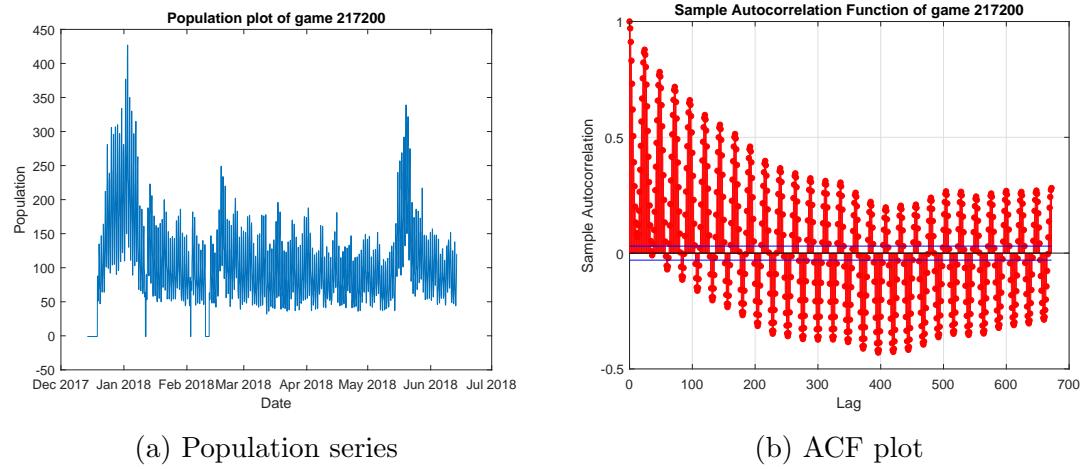


Figure 4.5: Population series and Autocorrelation function plot of the *Worms Ar-mageddon* game

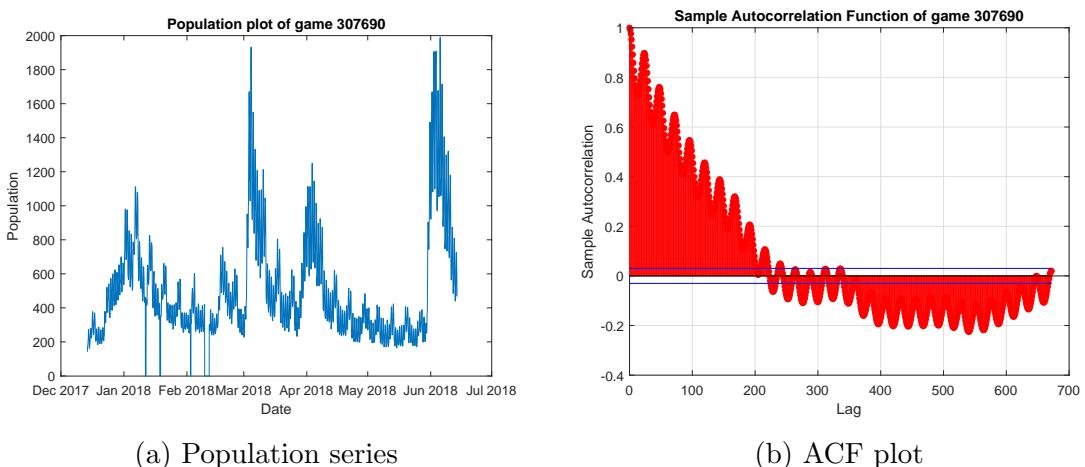


Figure 4.6: Population series and Autocorrelation function plot of *Sleeping Dogs: Definitive Edition* game

lags representing a day and a week. However, among those two the highest would be at a day's lag. Hence, as the first step, autocorrelation values for each game are calculated from lag 1 to lag 2030 and lag 1 to lag 180, for *5mData* and *60mData*, respectively. For *5mData*, lag 2016 represents a lag of a week and lag 288 represents a lag of a day. For *60mData* lag 168 represents a week and lag 24 represent a day. Thus, to include these lags, the mentioned boundary lag numbers were chosen.

### **Daily Pattern Detection**

Ideally, if a game displays daily patterns its highest autocorrelation value should appear at the lag that represents a day. Autocorrelation values are usually higher for lags that represent small time differences as population values that lie closer to each other are more correlated than values that are longer apart. This can be observed in the ACF plots presented earlier as well. Hence, for daily pattern detection, lags smaller than 6 hours are excluded as they have high autocorrelation values hindering the detection of daily seasonality. Thus, to detect the existence of daily patterns, the lag corresponding to the maximum autocorrelation value among all the lags from 72 to 300 and lags from 6 to 40 for *5mData* and *60mData* respectively is identified. This lag range represents lags from 6 hours to 1 day and 16 hours. The upper bound of the lag was selected to include a few lags beyond the 1 day mark and nothing more than 2 days. When a time series displays daily seasonality, the autocorrelation values of lags that represent multiples of a day's lag (eg: 24 hours, 48 hours lag) are usually higher compared to the other lags [106]. This can be observed from the ACF plots presented earlier in the chapter as well. An upper bound of 2 days was chosen as lags of up to 2 days are sufficient to determine the existence of daily seasonality. Choosing lags beyond that point to find the lag that represents maximum autocorrelation for daily seasonality detection is unnecessary and less efficient as multiple high autocorrelation values can be observed after 2 days lag (eg: at 48 hours, 72 hours).

As mentioned earlier, if a game exhibits daily seasonality, its lag corresponding

to the maximum autocorrelation would be the lag number representing a day. However, it is not always exactly the case due to distortions in the dataset. As games attract players from all over the world, time zone differences could result in some distortions. Thus, a game is labeled as a game displaying daily seasonality not only if its maximum autocorrelation occurs at exactly a day's lag, but also if it occurs in a lag within a certain distance to the day's lag. In order to determine the most appropriate distance, the percentage of games identified with increasing lag ranges were recorded and is depicted in Figure 4.7.

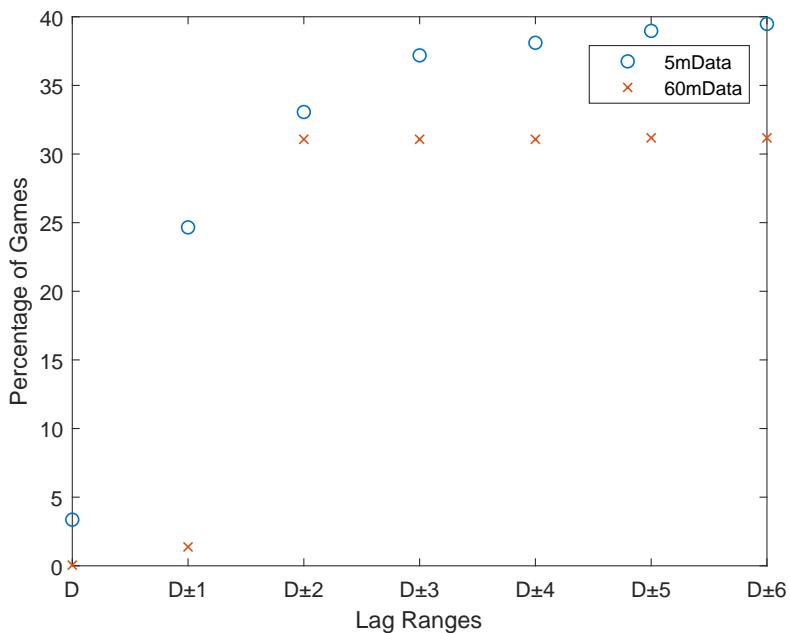


Figure 4.7: Percentage of Games displaying Daily Seasonality under different lag ranges

In Figure 4.7,  $D$  represents the lag number corresponding to a day where  $D = 288$  for  $5mData$  and  $D = 24$  for  $60mData$ . Each data point represents the percentage of games identified as displaying daily seasonality when the game's lag corresponding to maximum autocorrelation,  $m$ , lies in the range indicated by the x axis, eg:  $D - 1 \leq m \leq D + 1$ . The lag range for seasonality detection is selected by identifying a saturation point from the percentage of games series. The saturation point to select the lag range for seasonality detection is determined considering two criteria. First, if the percentage of games does not increase and has the exact same

value for at least three consecutive lag ranges, it is interpreted as saturation and the first occurring point of that sequence of points is chosen as the saturation point. Such a point cannot be detected if the percentage of games keeps on increasing even slightly. Hence a second criterion is used for such situations where the first criteria cannot be applied. If the percentage of games keeps on increasing, the difference between consecutive points are considered as it indicates the rate of percentage increase. The earliest point at which the percentage difference between the point and the next point is significantly lower compared to the percentage difference between the point and the previous point, is chosen as the saturation point. It can be seen that the percentage of identified games saturated and did not increase after the lag range  $D - 2 \leq m \leq D + 2$  for the *60mData*. Although the percentage of identified games keeps on increasing for each lag range for *5mData*, the percentage of increase keeps on declining. This can be observed by considering the difference of percentage between each consecutive data point pair of *5mData* in Figure 4.7. This difference is lower after the mentioned lag range  $D - 2 \leq m \leq D + 2$  compared to the lag ranges prior to that indicating a saturation of the percentage of identified games. Hence, a game is identified as displaying daily seasonality not only if its lag corresponding to maximum autocorrelation is exactly a day's lag ( $m = D$ ), but also if the lag yielding the maximum autocorrelation is within a range of 10 minutes for *5mData* and within a range of 2 hours for *60mData* from a day's lag on either side.

### **Weekly Pattern Detection**

Similar concepts and procedures are applied to weekly pattern detection as well where only the lag ranges are changed appropriately for weekly patterns. To detect if a game displays weekly seasonality, it is checked whether the lag corresponding to maximum autocorrelation represents a week's lag. Since autocorrelation is higher for lower numbered lags that represent closer data points as well as the first lag that represents a day, lower level lags are excluded as before. Specifically, all lags until one and half day's lag are excluded. Lags from 432 to 2030 for *5mData* and lags

from 36 to 180 for *60mData* are used to find the lag of maximum autocorrelation.

As before, a game is considered to display a weekly pattern, not only if its lag corresponding to maximum autocorrelation is a week, but also if it is within a selected distance from a week's lag. Several distance ranges were explored and the percentage of identified games in each range is depicted in Figure 4.8.

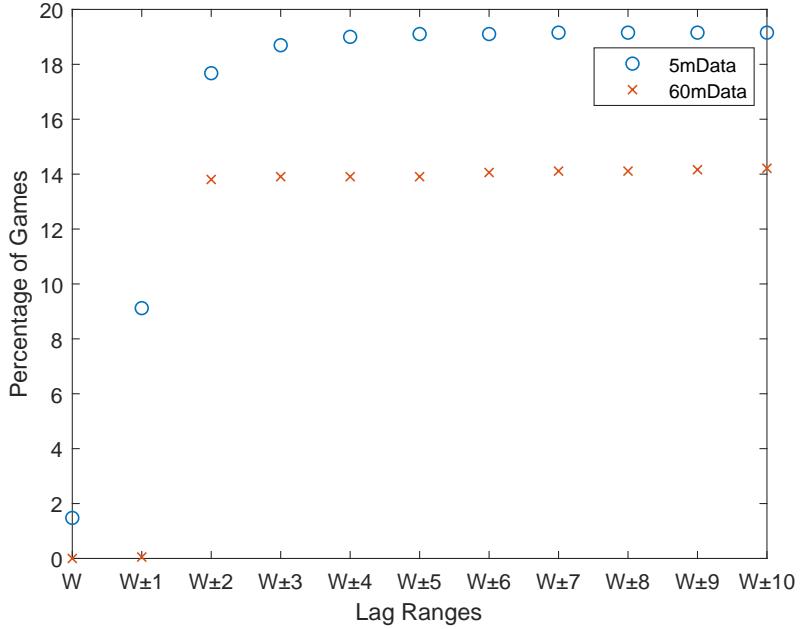


Figure 4.8: Percentage of Games displaying Weekly Seasonality under different lag ranges

In Figure 4.8,  $W$  represents the lag number corresponding to a week where  $W = 2016$  for *5mData* and  $W = 168$  for *60mData*. Each data point represents the percentage of games identified as displaying weekly seasonality when the game's lag corresponding to maximum autocorrelation,  $m$ , lies in the range given in the x axis, eg:  $W - 1 \leq m \leq W + 1$ . It can be seen that initially, the number of games being identified increases but it saturates after the lag range of  $W - 7 \leq m \leq W + 7$  for *5mData* and  $W - 3 \leq m \leq W + 3$  for *60mData*. Hence, a game is identified as displaying weekly seasonality if its lag corresponding to maximum autocorrelation falls within a range of 35 minutes from a week's lag either side for *5mData* and within a range of 3 hours for *60mData*.

The seasonality detection procedure can be summarized as follows. For daily

seasonality existence detection, autocorrelation values of lags 72 - 300 for *5mData* and 6 - 40 for *60mData* are calculated. A game is recognized as displaying daily seasonality if its lag corresponding to maximum autocorrelation is within a range of 10 minutes or 2 hours from a day's lag for *5mData* and *60mData* respectively. For weekly seasonality existence detection, autocorrelation values of lags 432 - 2030 for *5mData* and 36 - 180 for *60mData* are calculated. A game is recognized as displaying daily seasonality if its lag corresponding to maximum autocorrelation is within a range of 35 minutes or 3 hours from a day's lag for *5mData* and *60mData* respectively.

Although seasonality detection as explained can be directly calculated for some games, it is not accurate for games that exhibit a trend. Hence, a trend removal procedure is conducted prior to seasonality removal as presented in the next section.

### **4.2.2 Trend Removal**

Trends in time series can impact the reliability of seasonality detection. Hence, several trend removal procedures are explored and trends are removed prior to seasonality detection. A trend is removed by subtracting the value calculated by the fitted trend function from the original value, at a given point in the population time series. In this section, three trend removal procedures are introduced. Three procedures are introduced as it is not known beforehand which procedure is most suitable for removing the trend of the population series for seasonality detection. Hence, each of these procedures is used separately to remove the trend of the population series. Trend removed population series are then used for the seasonality detection process that was explained in the previous section. Due to the unavailability of a precise definition for a trend, various approaches are used to determine trend and remove it from a time series [126]. Hence, there is no standard approach or metric to evaluate a trend removal process and the most appropriate trend removal depends on the application. Although R-squared and RMSE can be used to evaluate how well an extracted trend fits to the underlying data series, it is not suitable to

determine if the trends are removed well for seasonality detection. Trend removal is conducted in this section to assist the mentioned autocorrelation-based seasonality detection process. Each trend removal process used is focused on removing a trend at a different level ranging from a single trend for the complete series and many mini-trends for the complete series. The three trend removal procedures are not applied together, rather applied separately to record how many games are identified as displaying short term seasonality (daily and weekly) by each approach. The number of games is recorded with respect to each trend removal because it helps in determining how well the removed trend has assisted the seasonality detection.

### 1. Linear Trend Removal

The linear trend of a population series is calculated as a least-squares fit estimated by a first order polynomial function fit that best fits the data series. The linear trend of a sample game is depicted in Figure 4.9. It can be seen that linear trend estimation can be used to remove the overall trend of a given game.

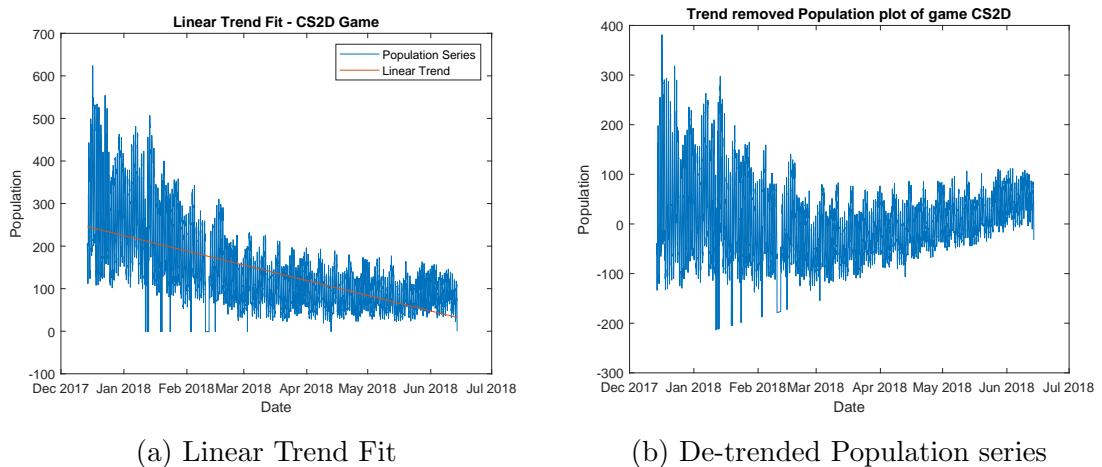


Figure 4.9: Before and after Linear Trend Removal of the game *CS2D*

### 2. High order Polynomial Trend Removal

In some games, the population could change in an irregular fashion where the trend cannot be represented by a linear fit. Hence, an order-8 Polynomial function fit is calculated by least-squares to represent the trend. For instance,

Figure 4.10 depicts the polynomial trend fitting of the game *Elite Dangerous* and the same population series after polynomial trend removal.

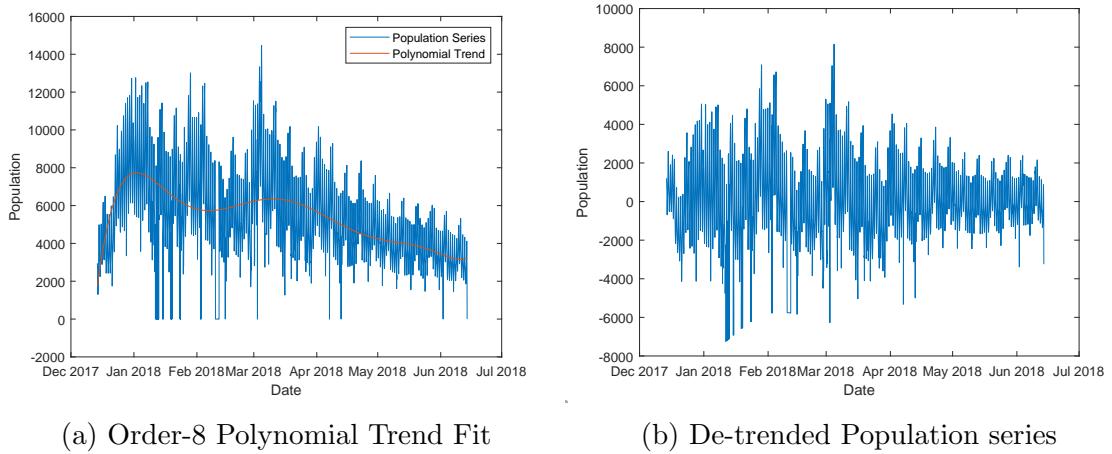


Figure 4.10: Before and after Order-8 Polynomial Trend Removal of the game *Elite Dangerous*

### 3. Piecewise Trend Removal

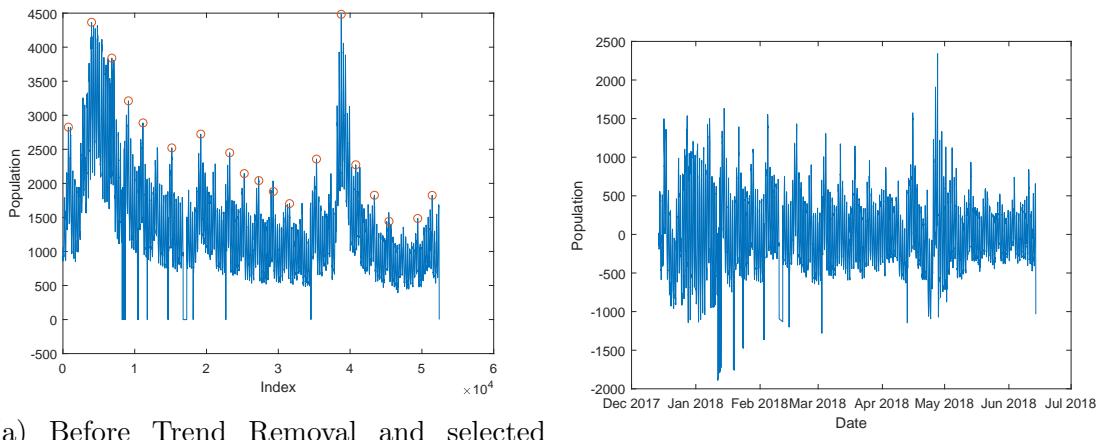
Instead of overall trends games can have mini trends as well. Thus, to further improve the trend removal process piecewise trend removal is considered. It is capable of removing the mini-trends while preserving any short term seasonal patterns available.

As the first step, a procedure to determine pieces in the time series data needs to be devised. Since the population data fluctuates differently in different games producing peaks and troughs, one way to determine pieces would be based on identifying peaks in the data. A *peak*, could be defined as a data point that is larger than its two neighbouring points. However, this definition would not only recognize significant peaks, but also local maxima making it problematic to determine boundaries of pieces. One approach to overcome this is by ignoring all peaks that occur within a given window size, while preserving important peak points that could serve as boundaries for pieces. For this purpose, *findPeaks*<sup>1</sup> implementation in Matlab was used by providing the *MinPeakDistance* as an input parameter. In essence, in *findPeaks*, all

<sup>1</sup><https://au.mathworks.com/help/signal/ref/findpeaks.html>

peak data points in a given series are first identified considering if it is larger than both of its neighbouring data points. Next, the highest peak points are selected by ignoring all peaks that occur within the chosen window size which is repeated for all identified peaks.

Appropriate window size has to be decided for the peak identification process. The ultimate goal of the peak identification process is to determine boundary points for pieces for trend removal. When selecting pieces for trend removal caution must be taken to not lose existing daily and weekly patterns and also to improve the quality of the trend removal process. Thus, the chosen window size should not be smaller than a week and not too large such that the mini trends are not corrected. Hence, 7 days was selected as the window size. The window size selected is 2016 for *5mData* and 168 for *60mData*. Once the pieces are identified, a linear fit is calculated for each piece and the trend is removed. For instance, Figure 4.11 presents the population series of the game *theHunter: Call of the Wild* before and after piecewise trend removal. It indicates how peaks serve as boundaries of identified pieces of the series.



(a) Before Trend Removal and selected peaks for piece boundaries

(b) After Piecewise Trend Removal

Figure 4.11: Before and after Piecewise Trend Removal of the game *theHunter: Call of the Wild*

This section presented the procedure conducted to identify games that display daily and weekly player population fluctuation patterns. The three trend removal techniques used to remove trend prior to seasonality detection were also presented.

In the next section, the outcomes of the seasonality detection procedure are presented.

### 4.2.3 Outcomes

This section presents the results obtained from the short term seasonality detection process. It also includes a discussion about the outcomes of the trend removal process and their strengths and weaknesses. Table 4.1 presents daily pattern results and Table 4.2 presents the weekly pattern results. In the tables *NoTrendRem* indicates results when no trend removal process was conducted prior to seasonality detection. It is used as the baseline approach for comparison.

	<i>NoTrendRem</i>	<i>Linear</i>	<i>Polynomial</i>	<i>Piecewise</i>
No of Games	1259	1259	1269	1337
Percentage (%)	64.14	64.14	64.65	68.11

Table 4.1: Percentage and Number of Games displaying Daily Patterns

	<i>NoTrendRem</i>	<i>Linear</i>	<i>Polynomial</i>	<i>Piecewise</i>
No of Games	649	668	861	1508
Percentage (%)	33.06	34.02	43.86	76.82

Table 4.2: Percentage and Number of Games displaying Weekly Patterns

When it comes to daily patterns it can be seen that *NoTrendRem*, *Linear*, *Polynomial* methods have identified almost the same number of games while the *Piecewise* method has identified a comparatively higher amount. Specifically, there is an increase of 3.5% when *Piecewise* method is used compared to *Polynomial* method. However, when compared to *NoTrendRem* there is only 0.51% increase in *Polynomial* method and 3.97% increase in *Piecewise* method. This indicates that the existence of trend in data has not highly impacted the identification of games with daily patterns.

When it comes to weekly pattern recognition, trend removal plays a major role in autocorrelation based seasonality detection. As indicated in Table 4.2, each trend removal process has resulted in identifying more games with weekly patterns. This

process was further verified through visual inspection of the trend removed series. It was conducted to verify that weekly seasonality is not an artefact introduced by trend removal. Whilst it is acknowledged that visual inspection is subjective, it was only conducted as a verification process in addition to the more systematic process of recording the number of games that display daily and weekly seasonality for each trend removal approach. Furthermore, visual inspection of time series is commonly conducted in time series analysis to observe trends and other properties of the series [127]. A sample set of games that display different types of trends, such as series with a single overall trend and series with several upwards and downward trends were chosen for the verification. Since this is a visual inspection that closely compares the series before and after trend removal for verification purposes, no precision or recall was recorded. In fact, precision or recall is not quite suitable as the verification compares the before trend removal series with after trend removal series rather than comparing the after trend removal series with some ground truth series. Comparing the percentage of games identified by *NoTrendRem* and *Linear* method, the *Linear* method has identified only a 0.96% percentage of more games compared to *NoTrendRem*. This implies that not many games in the dataset exhibit a linear trend that can be removed by *Linear*. On the other hand, *Polynomial* shows a 10.8% increase in the results compared to *NoTrendRem* indicating that more games in the dataset show fluctuations in population that can be represented by a high order polynomial. However, in most games, in fact, 77% of the games have been identified by the *Piecewise* method. Apart from indicating that most games display a weekly pattern, *Piecewise* also implies the highly irregular nature of player population fluctuations in games which makes it harder to use a linear function or a high order polynomial function to represent the trend. The piecewise linear function has an advantage over those methods as it could eliminate short term mini trends. Thus, based on the results, *Piecewise* technique was comparatively better in supporting the identification of games with weekly patterns.

Overall, the results indicate that most games, in fact 68% of games, display a

daily pattern in how the population fluctuates. This indicates that there exists a recurring daily pattern of the changes in number of players playing a game. Figure 4.12 depicts the average daily population pattern of games. To generate this pattern, the average daily pattern of each game over the 6 months period is extracted first. Each extracted daily pattern is then normalized to a 0 - 1 scale to focus on the shape alone rather than the magnitude and to avoid bias from games with higher population size on the final average pattern. Finally, the average daily population pattern of all games is generated by calculating the average of the average daily pattern of each game. The time in the daily pattern in Figure 4.12 is provided in UTC time zone which was used in data collection. It can be seen that the population starts to rise after 7am and the peak population is reached by 7pm. The population starts decreasing afterward. This depicts a 24 hour cycle in which the population increases for 12 hours and decreases for another 12 hours within a day. However, due to time zone differences the exact time of reaching peak and trough of daily population can be different between games. Nonetheless, as depicted in Figure 4.12 the average daily population pattern of games displays a shape where population gradually increases and decreases within the day.

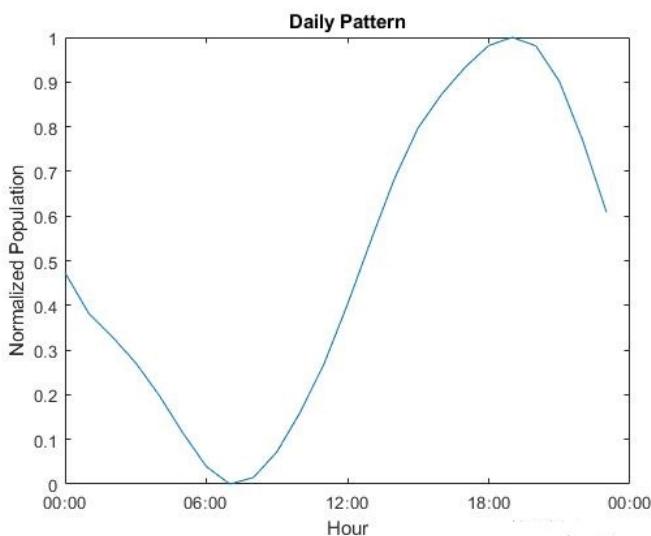


Figure 4.12: The average daily player population pattern of games

Furthermore, it was discovered that 77% of games exhibit a weekly pattern in how the player population fluctuates in a game throughout a week.

Since it was identified that the majority of the games display weekly patterns in player population fluctuations it is of interest to explore what different or similar weekly patterns games exhibit. The next section presents the details of the investigation conducted for that purpose.

## **4.3 Weekly Population Fluctuation Patterns Discovery**

This section presents the details of the study conducted to recognize the variety of weekly patterns displayed by games. A time series clustering approach is taken for this purpose. The methodology of the clustering approach is presented first, followed by the results.

### **4.3.1 Time Series Clustering**

As explained in Chapter 2, time series clustering involves partitioning a given set of time series into several distinct groups [114]. It is challenging due to the high dimensionality of the time series data and the complexity of determining a similarity measure [128]. Nonetheless, various goal-driven choices are made in time series clustering to accomplish this challenging task. The task of weekly patterns identification through time series clustering also involves making various choices related to steps involved in clustering. Specifically, the steps can be identified as data selection, data normalization, distance measure selection, clustering algorithm and related parameter selection and finally determining representative patterns of clusters. These are presented in the following sections to describe the methodology of the clustering based weekly pattern identification approach.

#### **4.3.1.1 Data Selection**

Games that were identified as displaying weekly seasonality in the previous experiment were chosen. Precisely, 1508 games out of the 1963 games were chosen.

Once the games are selected, what data or features of each game should be used in the clustering process have to be decided. The intention of this clustering process is to identify the archetypal shapes of weekly patterns games display. Hence, from each game, population data of a duration of a week have to be extracted. Since games have overall trends and mini-trends as a result of the influence from various external factors, it is unlikely that all weeks would appear the same in a certain game. Thus, averaging over the time period would not be an ideal method to extract a representative week from a game. Moreover, since the aim is to identify different weekly player population patterns that normally occur in games, it is not suitable to use weeks that behave differently due to external factors. Hence, population data that represent how population changes within a normal week has to be extracted from each game's population series. A normal week is considered as a week in which the population is not impacted by external factors. Hence, for each game, a week from the segment of the time series that has the lowest trend is extracted. The segment of a time series, whose slope of the linear trend is closest to zero (and whose mean is not different to those weeks that surround it) is recognized as the segment that has the lowest trend in the time series. Identifying a normal week based on the lowest trend closest to zero could be disadvantageous if the identified week corresponds to a situation where some servers of the game were down. However, it is not very common for servers to stay down for periods as long as a week. Another approach to find a normal week can be based on the most common trend. However, this can also be disadvantageous in situations where the game has events quite often. Then the week extracted based on the most common trend would be representing a week that is impacted by such external events. It is not suited to the definition of the normal week, which is a week not impacted or at least less impacted by external factors. Considering that the lowest trend based normal week extraction approach has lesser limitations it was selected. The process of extracting a normal week is further described as follows.

##### **Extraction of a normal week:**

1. **Piecewise Trend Calculation:** The previously described peak identification based piecewise linear trend calculation approach is used. For instance, Figure 4.13 represents the identified pieces of a game and the corresponding piecewise trend.

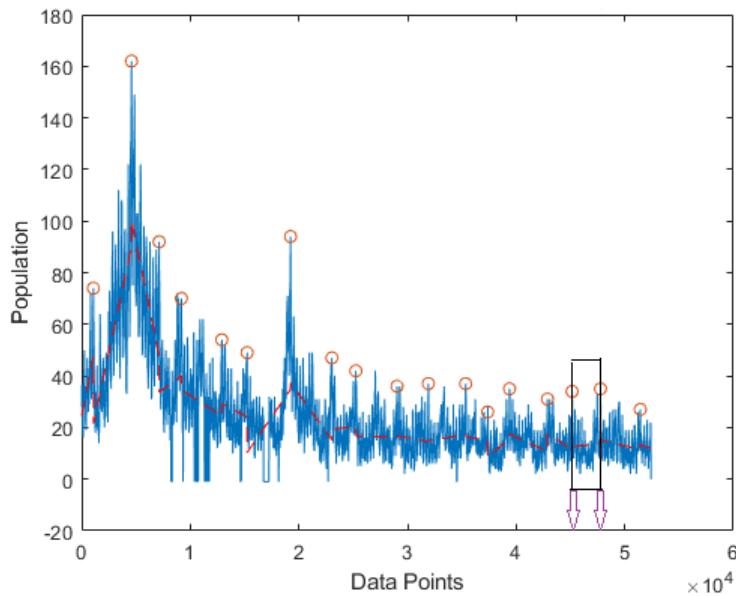


Figure 4.13: Peaks and piecewise trend of population plot of the game *observer*.

2. **Locating the piece with the lowest trend:** Once the trends of all pieces are calculated the indexes of the boundary points of the piece with the lowest trend is recorded. The piece with the lowest trend is the piece whose linear trend has a slope closest to zero. The region represented by a square in Figure 4.13 represents its piece with the lowest trend and indexes.
3. **Find indexes of a week within the chosen piece:** The aim is to extract a normal week starting from Monday and ending on Sunday for the clustering process. However, it is not certain whether the chosen piece will represent an exact week starting from Monday. Hence, a week within or from either side of the selected piece that satisfies the Monday to Sunday condition has to be extracted. For this purpose, the index of the ending boundary point of the piece is first used and the corresponding temporal information from the time series is identified. The temporal information, which is the date and time, is

used to determine the day of the week. Based on the temporal information, weeks starting from Monday and ending on Sunday that lies fully or partially within the boundaries of the piece are selected. For each selected week, the number of data points that lie outside the boundaries of the piece is calculated. The week with the smallest number of data points that lie outside the boundary is finally chosen as it has the smallest offset from the piece. Then the indexes of that week are recorded.

4. **Detrend the time series:** The trend of the time series is removed by piecewise linear trend removal using the pieces and trends previously identified.
5. **Extract the week from the trend-removed data:** The indexes of the previously recognized week is used to extract the population data of the week from the trend-removed time series. The region represented by a square in Figure 4.14 is the week extracted from trend removed time series based on the identified indexes.

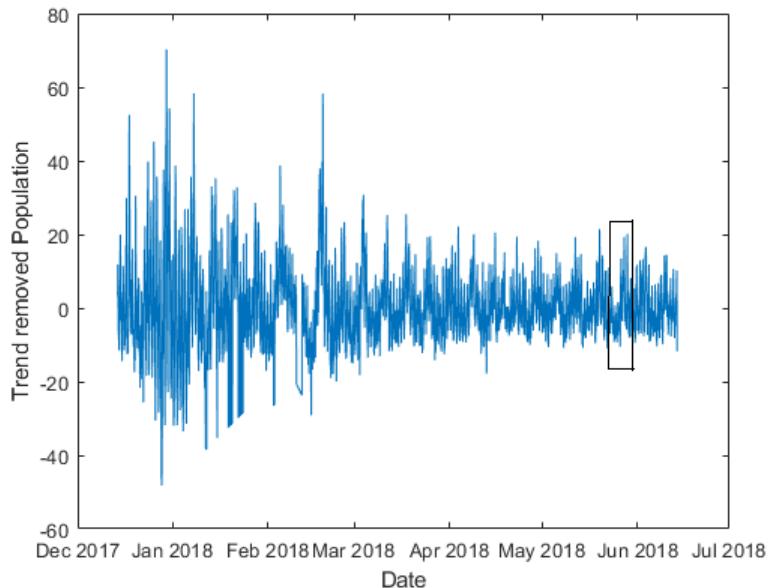


Figure 4.14: Trend removed population plot of the game *observer*.

#### 4.3.1.2 Normalization

Data normalization transforms raw data to a specific range of values, such as 0-1, in order to standardize raw data. Normalizing the extracted weekly player population data before clustering is necessary for several reasons. The overall population size of each game in the dataset is different and ranges from hundreds to tens of thousands. Since the goal for the clustering process is to discover the diverse weekly player population patterns based on the shape alone, irrespective of the size of overall player base of games, the extracted weekly population data should be normalized. Moreover, normalization is also necessary due to the choice of distance measure, which is presented later [129]. Thus, the extracted week of each game is normalized as per Equation 4.3 to scale the values to a range of 0 - 1.

$$\text{Normalized}(T_i) = \frac{T_i - T_{min}}{T_{max} - T_{min}} \quad (4.3)$$

In Equation 4.3,  $T_i$  represents the  $i^{th}$  value of time series  $T$ . The minimum value in  $T$  is represented by  $T_{min}$  while  $T_{max}$  represents the maximum value in  $T$ .

The population data of games in *5mData* is converted to an hourly frequency by calculating the hourly average. It is performed in order to use those along with *60mData*, which is already in hourly frequency, in the clustering process.

#### 4.3.1.3 Distance Measure Selection

Dynamic Time Warping (DTW) is chosen as the distance measure for the clustering process considering several factors. DTW is a shape-based measure appropriate for raw time series comparison [129]. Furthermore, it is invariant to local warps as it calculates the optimal global alignment between two time series handling, within bounds, any speed or length differences. Hence, it is appropriate for measuring the distance between weekly patterns based on shape addressing the shifts from time zone differences. More details concerning DTW were presented in Chapter 2.

Figure 4.15 depicts weekly population fluctuations of two games before and after DTW alignment. It can be seen that the slight distortions and delays in the original

series are handled to optimally align them highlighting their similarity. Thus, DTW is chosen as the distance measure in clustering.

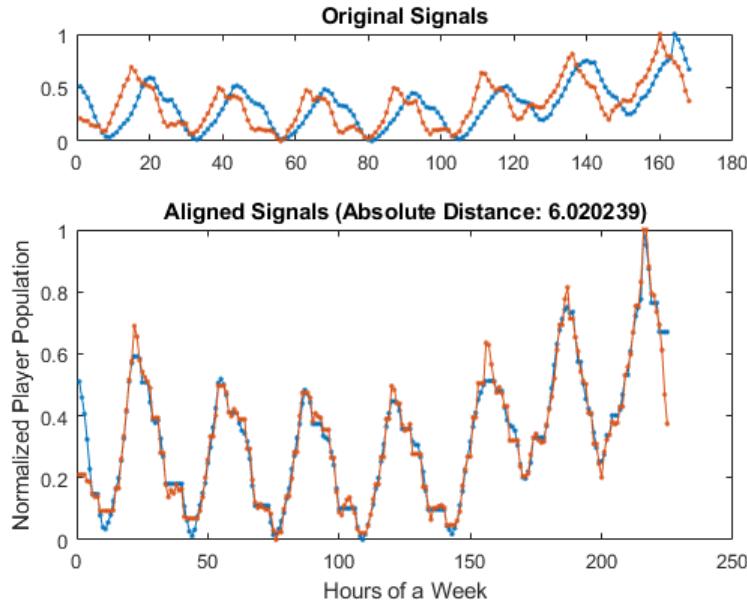


Figure 4.15: DTW Alignment of weekly player population of two games

#### 4.3.1.4 Cluster Technique and Parameter Selection

##### Clustering Technique

Agglomerative hierarchical clustering is used to perform the cluster analysis. An advantage of hierarchical clustering over partitioning methods, such as k-means is that the number of clusters needs not to be specified beforehand [130]. Since sufficient prior knowledge of the number of different weekly patterns that may exist is not available, hierarchical clustering was chosen.

##### Linkage

When forming clusters using hierarchical methods, linkage methods are used to determine how the clusters should be formed based on the distance between the data objects [117]. Three linkage methods were explored to select the best method suited for this clustering task. *Single Linkage* considers the distance between two clusters as the smallest distance between any 2 data objects in the two clusters. As opposed to single linkage, *Complete Linkage* uses the largest distance between any two data objects in the two clusters to represent the distance between the two

clusters. Moreover, *Average Linkage* calculates the average distance between all pairs of data objects in the two clusters to determine the distance between two clusters.

In order to determine the best linkage method for the clustering process, hierarchical clustering was conducted using each linkage method on the pre-calculated DTW distance matrix. Each linkage method was evaluated using *Cophenetic correlation coefficient*, which is a measure that indicates how well the distances between data objects, as provided in the distance matrix, are represented in the dendrogram [131]. More details about the measure were provided previously in *Chapter 2*.

The results depicted in Table 4.3 indicates that the Average Linkage method performs best in representing the original distance values between each pair of weekly population patterns of games in the dendrogram. Thus, the hierarchical cluster analysis process is continued with Average linkage. The resulting dendrogram is depicted in Figure 4.16.

Table 4.3: Cophenetic Correlation Coefficient for different Linkage methods

Cophenetic Corr. Coefficient	Linkage		
	Single	Complete	Average
	0.44	0.65	0.79

### Number of Clusters

One of the most important tasks in cluster analysis is deciding the optimal number of clusters. Dendograms can be used to assist in deciding the number of clusters, as they represent the natural divisions in the data. The dendrogram, as depicted in Figure 4.16 is quite skewed to the right and is not displaying any significant divisions of clusters. At first glance, this may indicate that there are no remarkable differences among the weekly patterns of games. But it is important to explore the types of different weekly patterns the player population data may exhibit, nonetheless how minor the differences are, as these could lead to interesting findings. Hence, different upper boundaries for the number of clusters were experimented. Specifically, the

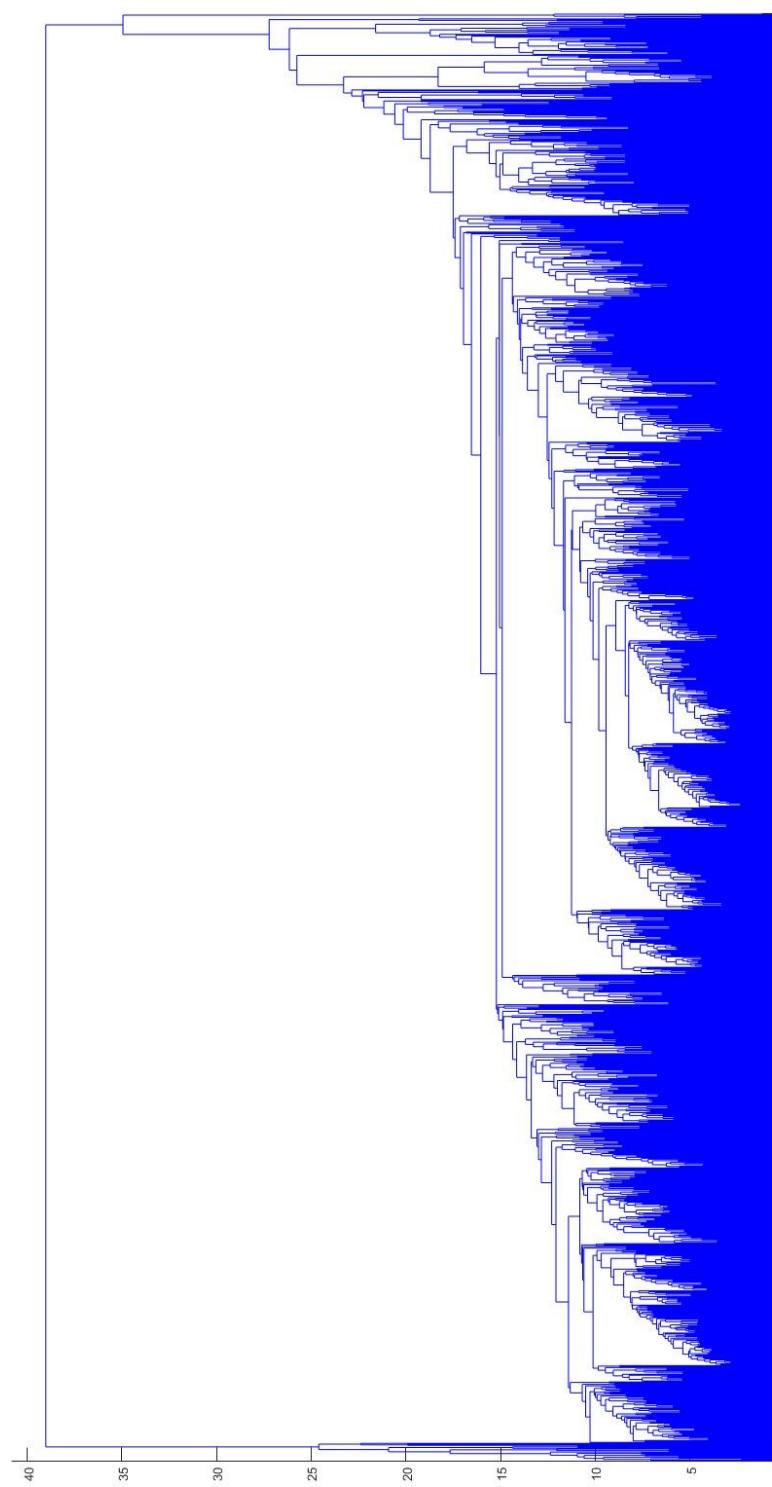


Figure 4.16: Dendrogram(1508Games):Clustering of weekly patterns based on the DTW distance by Average Linkage

number of clusters used were 5, 10, 15, 25, 50, 75 and 100. Only the clusters with 10 or more games were accepted as meaningful clusters as clusters with a low number of elements are not appropriate to make generalized claims about weekly patterns of games. Moreover, it was noticed that when 5, 10, 15, 25 and 50 were used as the upper bound for the number of clusters, a single dominating cluster with more than 1200 games is generated depicting a weekly pattern where the population is higher during Friday, Saturday and Sunday compared to other days. This negatively impacts the identification of diverse weekly patterns as more than 80% of games belong to a single cluster. However, at 75 and 100 boundaries the dominating cluster becomes separated. Hence, 75 was chosen as the optimum number of clusters for the cluster analysis process. From the 75 clusters, 10 clusters consisted of at least 10 elements. However, among those 10 clusters, one consisted of 10 games displaying unusual population fluctuations due to low population. Thus, only the 9 meaningful clusters are further explored. The archetypal weekly pattern shapes of each of these 9 clusters are depicted in Figure 4.18 and described later in Section 4.4. The process of generating these archetypal shapes is explained in the next section.

This section presented the details regarding the techniques and parameters chosen for the cluster analysis based weekly pattern identification methodology. To summarize, an agglomerative hierarchical clustering approach is used with Average Linkage and 75 as the maximum number of clusters. Also, Dynamic Time Warping (DTW) was chosen as the distance measure to measure the distance between weekly patterns for clustering. The 9 clusters with at least 10 games in them were selected to further explore. The next section presents the procedure of extracting a representative weekly pattern shape from the weekly patterns of games in a given cluster to generate the archetypal weekly patterns.

### **4.3.2 Representative Weekly Patterns Generation**

This section presents the procedure of generating archetypal weekly pattern shapes of the clusters.

Once the clusters are generated, it is necessary to construct weekly player population patterns representing each cluster. The simplest approach for this purpose, is to extract the average weekly pattern of all the weekly patterns of the games in each cluster. However, it is problematic due to alignment dissimilarities resulting from time zone differences. Thus, for this purpose a DTW based averaging procedure is proposed. This procedure also acknowledges the hierarchical clustering approach of joining elements one by one in a bottom-up fashion based on the distance between elements. It is presented in Algorithm 4.1. This approach generates the representative weekly pattern of a cluster by iteratively aligning pairs of weekly patterns in a cluster using DTW. It generates the average weekly pattern while following the order of weekly patterns of games in the hierarchical cluster tree.

---

**Algorithm 4.1** Visualizing Representative Weekly Pattern of a Cluster

---

```

Inputs:  $Data_{NxW}$  ,  $Order_K$ 
 $gameA \leftarrow Data[Order[1]][:]$ 
 $gameB \leftarrow Data[Order[2]][:]$ 
 $[iA, iB] \leftarrow DTW(gameA, gameB)$ 
 $avgAlign_{sizeOf(iA)}$ 
for  $i = 1 : sizeOf(avgAlign)$  do
     $avgAlign(i) \leftarrow \frac{1}{2} * gameA[iA[i]] + \frac{1}{2} * gameB[iB[i]]$ 
end for
for  $k = 3 : sizeOf(Order)$  do
     $tmpAvgAlign \leftarrow avgAlign$ 
     $gameB \leftarrow Data[Order[k]][:]$ 
     $[iA, iB] \leftarrow DTW(tmpAvgAlign, gameB)$ 
     $avgAlign_{sizeOf(iA)}$ 
    for  $i = 1 : sizeOf(avgAlign)$  do
         $avgAlign(i) = \frac{k-1}{k} * tmpAvgAlign[iA[i]] + \frac{1}{k} * gameB[iB[i]]$ 
    end for
end for

```

---

Algorithm 4.1 takes two inputs. The first input is a  $NxW$  matrix named  $Data$  where  $N$  is the total number of games used in the clustering process, which is 1508, and  $W$  is the number of data points representing a week. It also takes the connection order of games within the cluster in the dendrogram as in Figure 4.16 as an input named  $Order$ . It is an array of size  $K$  where  $K$  is the number of games within the cluster. It contains the indexes of games that belong to the cluster currently being

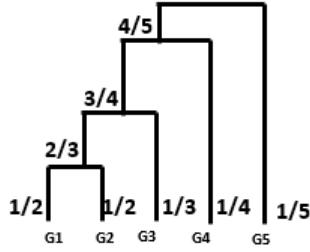


Figure 4.17: Example: Weights used in Representative Pattern generation for a cluster

considered, ordered based on how they were joined during the hierarchical clustering process. Initially, Algorithm 4.1 identifies the first two games joined in the cluster from *Order* and uses DTW to align them. The resulting  $iA$  and  $iB$  contains new alignment indexes of the two games. Next, it generates an average weekly pattern, represented by  $avgAlign$ , by iteratively averaging each pair of data points in the newly aligned pair of games. Next, the weekly pattern of the next joined game of the cluster and  $avgAlign$  is aligned using DTW and averaged. This iteratively continues until all games in the cluster are used to create the final representative weekly pattern.

Moreover, a weighting mechanism is used to correctly account for the number of weekly patterns involved in creating the average pattern. Since each game's weekly pattern is added to the average weekly pattern one by one, the number of weekly patterns involved to create the average pattern should be correctly accounted for in each iteration. Thus, as Figure 4.17 represents, weight is determined based on the number of games involved in each stage and more weight is given to the pattern(subtree) involving multiple games. For instance, in iteration 3 of Algorithm 4.1, the current average pattern,  $avgAlign$ , represents the average of 3 weekly patterns. Hence, it receives  $3/4$  and the other game's weekly pattern receives  $1/4$  as the weight during the averaging stage. In essence, in each iteration more weight is offered to  $tmpAvgAlign$ , which represents the average weekly pattern of multiple games.

All the details of time series clustering based weekly population pattern identification methodology were presented in the previous sections. The next section

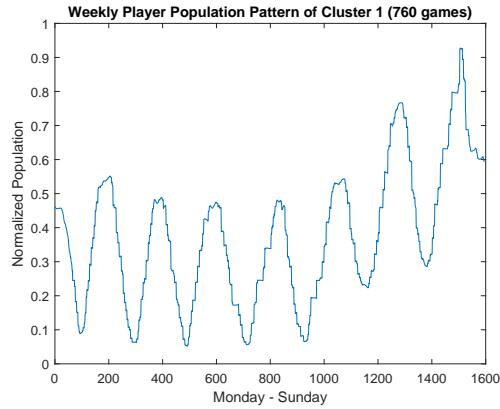
presents the outcomes of the clustering process.

## 4.4 Results and Discussion

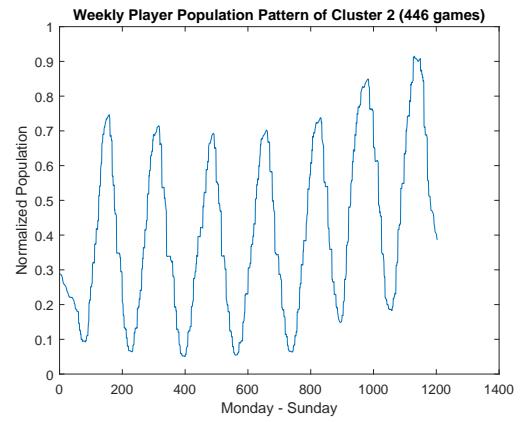
Through the clustering process, 9 meaningful clusters with at least 10 games in them were identified that exhibit different weekly player population fluctuation patterns of games. Figure 4.18 presents these discovered patterns where each peak in a pattern corresponds to a day of a week from Monday to Sunday. Table 4.4 depicts the characteristics of each weekly pattern. In Table 4.4, Mean Game Population represents the mean of the weekly pattern of the corresponding cluster presented in Figure 4.18. The Standard Deviation of Game Population is calculated by, first calculating the mean population of the normal week of each game in the cluster and then calculating the standard deviation of those values. The 3-Day Weekend Ratios represent the peak population values of Friday, Saturday and Sunday of the weekly pattern of the clusters in Figure 4.18. Weekends that include Friday are considered as population increases can be observed from Friday in the weekly patterns in Figure 4.18. The  $\frac{\text{Weekend}}{\text{Midweek}}$  ratio depicts the mean population of the period from Friday to Sunday divided by the mean population of the period from Monday to Thursday extracted from the weekly pattern of the cluster depicted in Figure 4.18.

#### 4.4. Results and Discussion

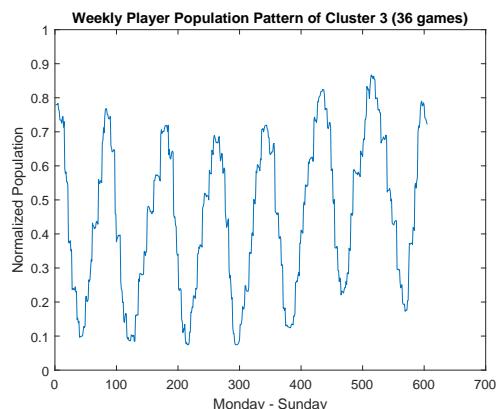
---



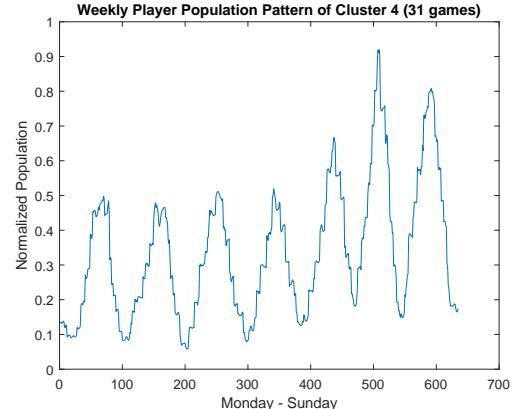
(a) Cluster 1: LowMidweek\_Fri-to-Sun-Upward Cluster



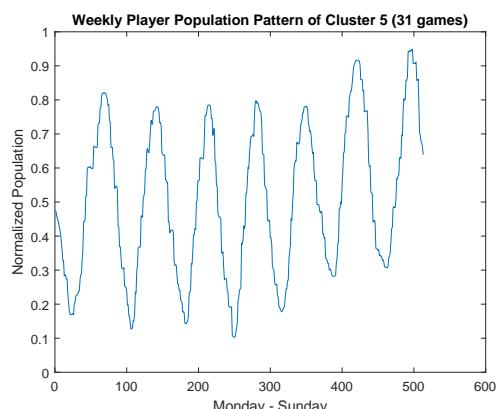
(b) Cluster 2: Fri-to-Sun-Upward Cluster



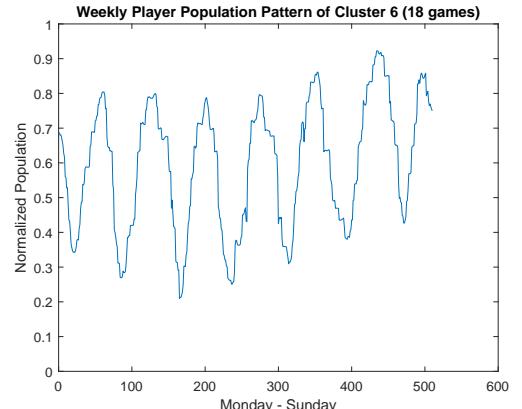
(c) Cluster 3: Fri-Sat-High Cluster



(d) Cluster 4: Sat-High Cluster



(e) Cluster 5: Sat-Sun-Same Cluster



(f) Cluster 6: Thu-to-Sat-Upward Cluster

#### 4.4. Results and Discussion

---

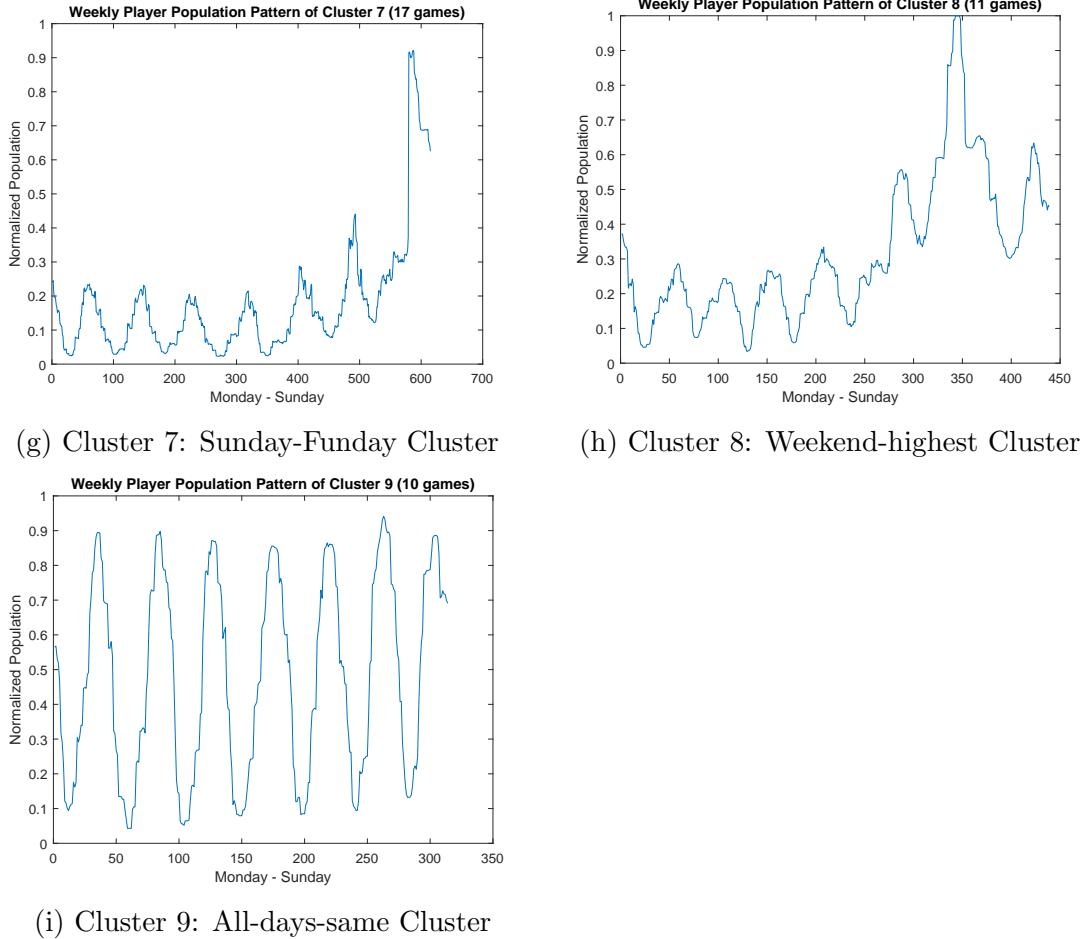


Figure 4.18: Representative Weekly Player Population patterns of Clusters: Represents how player population changes from Monday to Sunday. Each peak corresponds to a day of the week

Among the clusters, *LowMidweek\_Fri-to-Sun-Upward Cluster* has the highest number of games, which is 760, making it the dominating cluster with 50% of games in the dataset. As per that cluster, in most games, players tend to play games highly towards the end of the week. It can be seen that the population has started to increase on Friday and has continued to increase till Sunday. Further, *Fri-to-Sun-Upward Cluster* which has 446 games, making it another dominant cluster, displays a similar pattern. However, the player population difference between weekdays and weekend tends to be higher in *LowMidweek\_Fri-to-Sun-Upward Cluster* than *Fri-to-Sun-Upward Cluster*. Specifically, in *LowMidweek\_Fri-to-Sun-Upward Cluster*, the mean weekend population is 1.62 times the mean midweek population while it is 1.31 for *Fri-to-Sun-Upward Cluster* as per Table 4.4. Apart from *LowMid-*

Clust ID	No. of Games	Mean Game Popula-tion	Standard Devia-tion of Game Popu-lation	Peak Day of Week	3-Day Week-end Ratios	$\frac{\text{Weekend}}{\text{Midweek}}$ Ratio
1	760	0.38	0.045	Sunday	[0.54,0.76,0.93]	1.62
2	446	0.41	0.041	Sunday	[0.72,0.83,0.91]	1.31
3	36	0.46	0.042	Saturday	[0.82,0.85,0.78]	1.30
4	31	0.34	0.036	Saturday	[0.66,0.92,0.81]	1.46
5	31	0.51	0.031	Sunday	[0.77,0.92,0.94]	1.17
6	18	0.60	0.038	Saturday	[0.86,0.92,0.85]	1.18
7	17	0.17	0.038	Sunday	[0.26,0.42,0.91]	2.27
8	11	0.31	0.024	Saturday	[0.55,0.99,0.63]	2.44
9	9	0.48	0.018	Saturday	[0.85,0.94,0.88]	1.15

Table 4.4: Characteristics of the 9 weekly patterns

*week-Fri-to-Sun-Upward Cluster* and *Fri-to-Sun-Upward Cluster*, all other clusters have a comparatively lower number of games in them. In general, even these non-dominating clusters represent a weekly pattern where population is higher during the weekend than weekdays. However, there exist some significant differences in how population varies during the weekend among these clusters. For instance, in *Fri-Sat-High Cluster*, *Sat-High Cluster*, *Thu-to-Sat-Upward Cluster* and *Weekend-highest Cluster*, player population is greater on Saturday than Friday and Sunday. This rise is higher in *Weekend-highest Cluster*. This suggests to us that players tend to engage with some games mostly on Saturdays. By contrast, players are also likely to play some other games heavily on Sundays as depicted in *Sunday-Funday Cluster*. It is interesting to observe that the player population has intensely escalated on Sunday in *Sunday-Funday Cluster* compared to all other clusters. On the other hand, *Sat-Sun-Same Cluster* has the smallest difference between peak Saturday and Sunday population, which is 0.02. Lastly, *All-days-same Cluster* seems to be quite interesting as it represents a weekly pattern where population appears to be consistent throughout the whole week. It has the lowest difference between weekend and midweek population compared to the other clusters as its mean weekend population is only 1.15 times the mean midweek population. It could also be observed from the

Table 4.4 that in all clusters the player population within midweek is lower than that of during the weekends as  $\frac{\text{Weekend}}{\text{Midweek}}$  ratio is higher than 1 in all clusters. Among them, *LowMidweek-Fri-to-Sun-Upward Cluster*, *Sat-High Cluster*, *Sunday-Funday Cluster* and *Weekend-highest Cluster* have the highest difference between weekend and weekdays which can also be observed from Figure 4.18.

#### 4.4.1 Extraction of Game Characteristics

In order to provide further insights into the archetypal weekly population patterns, the common characteristics of games, namely, tags, age requirements and overall population size of games in each cluster were explored.

**Tags:** Tags are used for characterizing games. Each game has multiple tags of different sizes where *tag size* means the number of players who have assigned a certain tag to the game. A weighting criterion is used to determine how each game in a cluster contributes towards a certain tag. The weight a game assigns to a certain tag is calculated relative to the overall population size of the game. It is calculated as such so as to allow fair inclusion of tags of games with less population within a cluster. As per the definition, tag size has an indirect connection with the overall population size of a game. Thus, the weight contribution of a game  $g$  towards a certain tag ( $Tag_x$ ) is calculated as per Equation 4.4.

$$Weight_{g,Tag_x} = \frac{Size_{g,Tag_x}}{\max_{i=1..T}(Size_{g,Tag_i})} \quad (4.4)$$

In Equation 4.4,  $Size_{g,Tag_x}$  is the number of players who have assigned  $Tag_x$  to the game  $g$ . Since the size of  $Tag_x$  is divided by the maximum tag size value out of all  $T$  tags of that game, the weight a game contributes towards a tag is now relative to the overall player population of a game. Finally, the percentage of a certain Tag in a cluster is determined by first calculating the sum of weights contributed by all games in the cluster and then dividing it by the number of games in the cluster and finally multiplying by 100. The top 10 tags with the highest percentage in each cluster are presented in Table 4.5.

#### 4.4. Results and Discussion

---

ClustID	Tags
1	Action[40%], Singleplayer[35%], Multiplayer[31%], Strategy[31%], Adventure[29%], Open World[24%], Indie[24%], RPG[23%], Simulation[23%], Survival[16%]
2	Multiplayer[36%], Action[34%], Free to Play[30%], Strategy[29%], Simulation[25%], Singleplayer[22%], Open World[18%], Adventure[18%], RPG[18%], FPS[13%]
3	Indie[40%], Action[37%], Multiplayer[36%], RPG[31%], Strategy[27%], Adventure[24%], Singleplayer[21%], Massively Multiplayer[20%], Open World[20%], Simulation[20%]
4	Action[51%], Multiplayer[36%], Adventure[36%], Indie[33%], Singleplayer[27%], RPG[25%], Open World[22%], Simulation[21%], Survival[21%], Strategy[21%]
5	Free to Play[51%], Multiplayer[43%], Open World[40%], RPG[32%], Action[30%], Massively Multiplayer[28%], Survival[27%], Adventure[26%], Strategy[26%], Simulation[22%]
6	Free to Play[98%], Massively Multiplayer[56%], Action[52%], Multiplayer[51%], RPG[37%], MMORPG[35%], Anime[34%], FPS[29%], Open World[26%], Adventure[26%]
7	Action[52%], Indie[51%], Adventure[32%], Casual[30%], Singleplayer[25%], Multiplayer[24%], Strategy[23%], Funny[23%], RPG[20%], Great Soundtrack[17%]
8	Action[65%], VR[39%], Adventure[30%], Multiplayer[29%], First-Person[29%], Co-op[28%], Singleplayer[24%], FPS[22%], Indie[20%], Strategy[19%]
9	Action[38%], Strategy[36%], Free to Play[30%], Singleplayer[25%], Adventure[20%], Platformer[19%], Multiplayer[18%], Great Soundtrack[18%], Simualtion[15%], Casual[13%]

Table 4.5: Top 10 Tags of each cluster sorted by the weighted percentage of games

**Age Restrictions:** Based on the time availability, interest and other factors, weekly playing patterns could differ among players of different age groups. The rating system of ‘Entertainment Software Rating Board (ESBR)’ assigns each game an age-based rating category based on the content of the game. To explore the distribution of games in each cluster with respect to the age restrictions, the percentage of games of each age-based rating category in each cluster was calculated. Results are depicted in Figure 4.19.

**Overall Population Size:** For each game, the average player population throughout the 6 months time period the data were collected was calculated. Next, for each cluster, Minimum, Average and Maximum of Average player population of all games

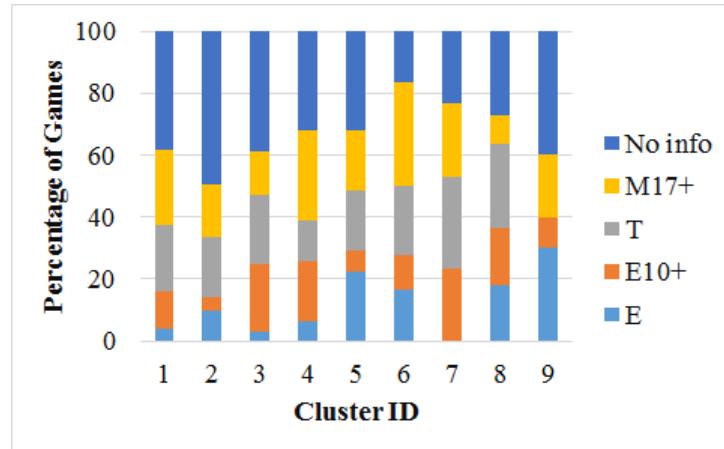


Figure 4.19: Age based Class Distribution of Games in Clusters:- E: Everyone, E10+: Everyone10+ (Ages 10 and up), T: Teens (Ages 13 and up), MA17+: Mature17+ (Ages 17 and up)

was calculated. These value ranges provide a perception about the population sizes of games in each cluster as presented in Figure 4.20.

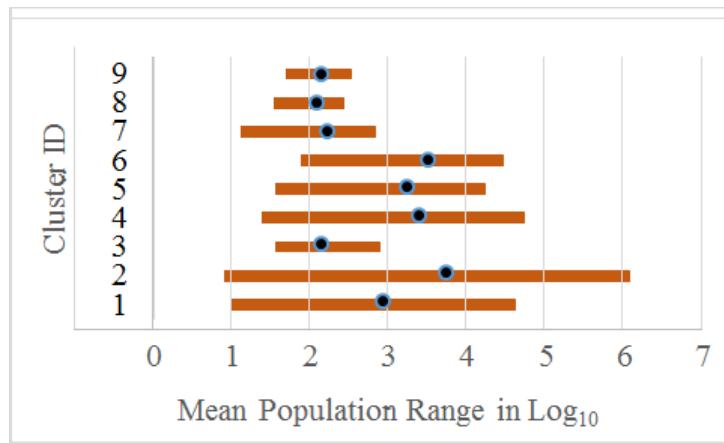


Figure 4.20: Range of Mean Population of games in Clusters: Per each cluster, mean population of each game over 6 months is calculated. The minimum, mean and maximum of those values are converted to log<sub>10</sub> and presented

#### 4.4.2 Discussion

Extraction of game characteristics in each cluster revealed several perceptions about the games that display diverse weekly player population patterns. Games displaying the pattern of *LowMidweek\_Fri-to-Sun-Upward Cluster* are mostly Action, Single player, Multi player and Strategy games and belong to T and M17+ age classes mostly. The mean population ranges between 9 to 41742 in those games. Such

#### *4.4. Results and Discussion*

---

a high population range is observable as *LowMidweek\_Fri-to-Sun-Upward Cluster* consists of the highest number of games. However, it is interesting to note that the pattern of *Fri-to-Sun-Upward Cluster* is observable in games of a much larger range of population although *Fri-to-Sun-Upward Cluster* has lesser games than *LowMidweek\_Fri-to-Sun-Upward Cluster*. It could be due to the popular titles in *Fri-to-Sun-Upward Cluster* such as *PUBG*, *DOTA2* and *Counter Strike:Global Offensive*. Games in *Fri-to-Sun-Upward Cluster* are mostly Multi player, Action, Free to Play and Strategy combinations and belong to T and M17+ age classes mostly. While the major tags in *Fri-Sat-High Cluster* are Indie, Action and Multiplayer, its mean population range is quite small and most games belong to E10+, T and M17+ age classes. In *Sat-High Cluster*, the Action tag is leading and its percentage is considerably higher than the next tags, which are Multiplayer and Adventure. Its mean population ranges between 23 and 54927 and most games belong to E10+ and M17+ age classes. Most games displaying the *Sat-Sun-Same Cluster* pattern are Free to Play, Multi player and Open World. Interestingly, Open World appears among the top 3 tags only in this cluster. Also, its maximum mean population is 17513, which is somewhat smaller than *Sat-High Cluster* and most games belong to E, T and M17+ age classes. Almost all games displaying the *Thu-to-Sat-Upward Cluster* pattern are Free to Play, as its percentage is 98%. Also, the existence of Massively Multiplayer, Multiplayer and MMORPG describes that most games in this cluster are multi player. Probably as a result, the minimum mean population is highest in this cluster. Also, most games in *Thu-to-Sat-Upward Cluster* belong to M17+ class followed by T, E and E10+ age classes. The *Sunday-Funday Cluster* pattern is displayed mostly in Action and Indie games. Interestingly, games tagged as Casual and Funny also exist in *Sunday-Funday Cluster*. The mean population of games have a smaller range and also a small minimum value of 12. Interestingly, this cluster contains some popular titles, *Far Cry Primal* and *Assassin's Creed Syndicate* from Ubisoft developers. Also, 2 games from the same sequel, *The Jackbox Party Pack 3* and *4* also appear. Since these 2 are party games, their appearance

#### *4.4. Results and Discussion*

---

in *Sunday-Funday Cluster* which has a pattern of a much lower population during the weekdays and a significantly higher population on Sunday is understandable. Games in this cluster belong to mostly T, E10+ and M17+ and none in E age class. *Weekend-highest Cluster* has mostly Action and VR games and VR did not appear in other clusters. Thus, it is arguable whether there is any relationship between access to VR equipment and time to play VR games has any relationship with the pattern of population increasing highly on Saturday compared to the weekdays. Also, *Weekend-highest Cluster* has the lowest mean population which may indicate that a small number of players play VR games. Also, games in this cluster mostly belong to T, E and E10+. *All-days-same Cluster* games are mainly Action, Strategy and Free to Play. The small range of mean population indicates that all games in this cluster have somewhat similar population size. Moreover, most games in this cluster belong to E age category compared to M17+ and T which could be related to the pattern of *All-days-same Cluster* where population is similar across all seven days of the week.

Analysis of tags across clusters indicates that the discovered weekly patterns are not associated with a single unique tag but with different combinations of tags as presented in Table 4.5. However, few clusters had tags that are unique and assigned to the majority of games in that cluster. Some instances are Free to Play and Multiplayer in *Thu-to-Sat-Upward Cluster*, Casual and Funny in *Sunday-Funday Cluster* and VR in *Weekend-highest Cluster*. Moreover, the distribution of age-based classes appears to be only slightly different between clusters. Each cluster contains games belonging to each age class except *Sunday-Funday Cluster* which does not have games from E class and *All-days-same Cluster* which does not have games from T class. *All-days-same Cluster* has the highest percentage of games belonging to E class representing games everyone can play without any age restrictions. *Sunday-Funday Cluster* has the highest percentage of games belonging to E10+ class and T class. The highest percentage of games of M17+ class is in *Sat-High Cluster*. Moreover, the mean values of the mean population of games in clusters are quite

different among clusters as depicted in Figure 4.20. The largest range in mean population is observable in *Fri-to-Sun-Upward Cluster* while the range is smallest in *All-days-same Cluster*. Also, the mean population is highest in *Fri-to-Sun-Upward Cluster* closely followed by *Thu-to-Sat-Upward Cluster* and *Sat-High Cluster*. The smallest mean population is in *Weekend-highest Cluster* closely followed by *All-days-same Cluster* and *Fri-Sat-High Cluster*. The differences of mean population of games among clusters may indicate the influence of overall population of a game to a weekly player population pattern.

Table 4.6 depicts a summary of the characteristics of the games in each cluster along with some example games including major games in the cluster. Cluster labels provide an indication of the shape of the weekly pattern displayed by the games in each cluster. It was revealed from the analysis that the games in each cluster cannot be described by a single tag, but by combinations of multiple tags. The most common tags of the games in each cluster are presented in the table. The cluster to which a game belongs to can be predicted based on the tags assigned to the game and its mean population by referring to Table 4.6. However, it would not be straightforward for all games as games have multiple tags and some clusters also contain overlapping tags.

## **4.5 Conclusion**

This chapter presented the study conducted to investigate short term seasonality in player population fluctuations, specifically focused on daily and weekly patterns to understand the player population changes in the presence of temporal factors, namely, the time of the day and day of the week on game player population fluctuations. Player population data of 1963 games collected over a period of 6 months was used for the study.

It was identified that the majority of the games (68%) display daily patterns where player population rises and falls in a cycle in a day. Furthermore, a majority of games (77%) were identified as displaying weekly patterns in population fluctuations.

#### 4.5. Conclusion

---

Cluster	Tags	Mean Population Size	Games
<b>LowMidweek_Fri-to-Sun-Upward Cluster</b>	Action, Singleplayer, Multiplayer, Strategy	911.19	This is the largest cluster and it contains various games.
<b>Fri-to-Sun-Upward Cluster</b>	Multiplayer, Action, Free-to-Play, Strategy	5655.63	PUBG, DOTA2 and Counter Strike:Global Offensive
<b>Fri-Sat-High Cluster</b>	Indie, Action, Multiplayer	146.78	Golf With Your Friends,Clone Drone in the Danger Zone, EverQuest
<b>Sat-High Cluster</b>	Action, Multiplayer, Adventure	2552.43	Grand Theft Auto V, DARK SOULS III, Assassin's Creed Origins
<b>Sat-Sun-Same Cluster</b>	Free-to-Play, Multiplayer, Open World	1802.01	Heroes & Generals, The Lord of the Rings Online, Neverwinter
<b>Thu-to-Sat-Upward Cluster</b>	Free-to-Play, Massively Multiplayer, Multiplayer and MMORPG	3403.46	Payday 2, Black Squad
<b>Sunday-Funday Cluster</b>	Action, Indie, Casual, Funny	176.57	Castle Crashers, The Jackbox Party Pack 3 and 4, Far Cry Primal
<b>Weekend-highest Cluster</b>	Action, VR	126.24	Arizona Sunshine, The Lab
<b>All-days-same Cluster</b>	Action, Strategy, Free-to-Play	142.88	Star Wars Rebellion, Spellstone, Bejeweled 3

Table 4.6: Characteristics of the games in Clusters

As presented in Chapter 2, previous studies that have investigated daily and weekly patterns in games have focused on either single games [17], [26], [82] or a few popular games [32]. Our study has extended the current knowledge about the existence of daily and weekly patterns of player population fluctuations of games using a larger dataset of popular games. While further confirming the existence of daily and weekly patterns in games as mentioned in previous studies, this study also reveals that there are games that do not display daily and weekly patterns in player population fluctuations. Thus, while a majority of games have displayed short term seasonality, this study also identified that there are games that do not display daily or weekly patterns which the previous studies have not been able to identify due to the smaller number of games investigated in those studies.

The study also identified 9 weekly player population patterns that games display. Whilst the most common weekly pattern, displayed by 50% of the games, was a pattern where player population increases from Friday to Sunday, several other patterns were also revealed. Some archetypes showed the highest increase occurs on Sundays, while for some it was Saturday. Another archetype had no difference among all seven days of the week. Also, another archetype had a considerably similar increase on both Saturday and Sunday. Previous studies in the literature have reported that there is a higher number of players and playing activity during the weekend compared to weekdays in *World of Warcraft*, *Warhammer Online* [17] and *Everquest II* [82]. Using a dataset of 50 games Chambers et al. have observed a similar pattern [32]. However, as these studies are limited to a single or a few games, insights regarding other weekly patterns that could exist have not been identified. Using a larger dataset of games the study presented in this chapter revealed that there are multiple weekly player population patterns games display. Whilst the literature only indicates that a higher player population can be observed during the weekend, this study reveals the various weekly patterns games display along with insights regarding the proportion of games displaying each pattern, weekend and midweek population ratio of each pattern, which days of the week have a higher population

#### *4.5. Conclusion*

---

in each pattern and what characteristics the games displaying each pattern have.

The characteristics of the games in each cluster, namely, tags, mean population size and age restrictions were also analysed to better understand the games displaying each archetypal weekly pattern. It was identified that the distribution of age restrictions of games was not highly distinguishable among archetypes. Common tags assigned to the games in each cluster were identified. However, the analysis revealed that the games in each cluster cannot be described by a single unique tag rather by a combination of tags. Furthermore, differences of the mean player population of games among clusters were also identified. The cluster analysis revealed that crisp clear definitions regarding the games that display each weekly pattern cannot be discerned by tags alone due to the occurrence of tags that overlap between some clusters. However, the tags and the mean player population can be used to characterize the games that display each weekly pattern to some extent. For instance, it was identified that the weekly pattern in which player population rises very highly on Sunday compared to other days is displayed by Action, Indie, Casual, Funny games with a mean population of around 176.

The main practical implication of the findings of this study is that for upcoming game developers, who do not have their own data for analysis, who can become aware of the various weekly player population patterns games display and learn what to expect from their games. Also, game server infrastructure providers can become aware of the weekly population patterns of various types of games. They can then use that knowledge when providing shared game hosting services. For instance, knowing that Action, Indie, Casual, Funny games with a mean population of around 176 tend to display a weekly pattern where population becomes extremely high on Sunday, they can avoid hosting such games together on the same server to avoid server outages on Sunday. Instead, they can host games that display different weekly patterns together. Moreover, third-party game marketers can also learn when most players of different types of games can be expected within the week and run promotional campaigns on such days to reach more players.

#### *4.5. Conclusion*

---

The study conducted in this chapter was focused on addressing the research question “How does player population of games fluctuate during a day and a week?”. This was addressed by revealing that in a majority of games the pattern in which player population fluctuate during a day and a week is recurring. Also, the study revealed the average daily pattern games display and different weekly patterns games display along with the characteristics of games that display each weekly pattern. One limitation of the study is that the explored features related to the games displaying each archetypal weekly pattern were limited to tags, age limitation and mean population.

In conclusion, this chapter revealed that the majority of games display short term seasonality, namely, daily and weekly, in player population fluctuations. Furthermore, archetypal weekly player population fluctuation patterns that games display and their frequencies and game characteristics were also discovered.

In the next chapter focus will be given to long term player population fluctuation patterns to understand the player population changes happening across the lifetime of a game.

# **Chapter 5**

## **Player Population based Life Cycle Patterns of Games**

The previous chapter investigated short term seasonality in player population fluctuations of games. It revealed the existence of daily and weekly population patterns depicting the player population changes in the presence of the temporal external factors; time of the day and day of the week. Going beyond the short term population changes, this chapter is focused on the long term player population fluctuations. The chapter considers the player population changes in the presence of the temporal external factor; time since the game was released. Hence, this chapter presents a study conducted to identify product life cycle shape archetypes of games based on long term player population fluctuations to address the second research question of the thesis.

Products display life cycle shapes depicting the changes in product sales. While the most common product life cycle shape contains four life stages, namely, introduction, growth, maturity and decline there are several other life cycle shapes products display [132] [133]. Since games are a product with its own unique characteristics, its life cycle shape could also be distinctive. However, as explained in Chapter 2, life cycle shapes of games depicting the changes in player population size of games throughout the lifetime have not been thoroughly explored in previous studies. Re-

---

vealing archetypal life cycle shapes of games is important to better understand how the popularity and player attraction to various types of games change through time. Also, it aids in understanding the life expectancy of games and scheduling promotions, monetization and game updates at different life stages to further extend the game popularity. Moreover, it could also provide insights about the life stages of various kinds of games that could be helpful for developers to decide what kind of games they would prefer to develop and what to expect from them.

In this chapter, an investigation of product life cycle shapes of games is conducted using player population data of 683 games available in Steam. Player population data since the release date of each game is used in the study. In order to extract the life cycle shapes of individual games two piecewise linear regression algorithms are introduced in the study. Moreover, the archetypal life cycle shapes games display are revealed through a clustering process which is focused on the life cycle shapes during the first year and first three years after the game release. Characteristics of games displaying each life cycle archetype are analysed to generate further insights about the archetypes.

The key findings of the study indicate that there are four distinct product life cycle archetypes games display. Two of the archetypes display profiles of decreasing population with varied decay rates. Specifically, the most common life cycle archetype depicts a pattern where the initially high population size rapidly decreases during the first few months after release and the population size remains low afterwards. The other archetype displays a slowly decreasing population pattern. Moreover, whilst one archetype displays population growth throughout the first year, another archetype displays a nearly steady population throughout the first year that drops at the end of the year. Furthermore, investigating the games displaying life cycle archetypes during the first year and first three years after release, it was identified that life cycle shapes of some games change after the first year while for some other games it stays the same.

This chapter first provides the research procedure of the study including the

data collection and preprocessing procedure, the piecewise linear regression based algorithms used to extract life cycle patterns of games and the clustering process conducted to identify life cycle archetypes. Then the results of the study are presented including the identified life cycle archetypes and an analysis of game characteristics of the archetypes. Finally, the chapter conclusion is presented.

## **5.1 Research Procedure**

This section presents the research procedure of the study conducted in this chapter. An overview of the research procedure is first provided in this section and it is followed by a detailed description of the data collection and pre-processing approach, piecewise linear trend extraction approaches used to extract life cycle shapes, and clustering approach used to discover life cycle archetypes.

### **5.1.1 Overview**

The main objective of this study is to uncover archetypal life cycle patterns of games based on player population fluctuations. For this purpose, population data series are investigated from several perspectives. Specifically, life cycle shapes during the first year and first three years after game release are explored using two piecewise linear trend extraction approaches. This is also depicted in the overview of the research procedure in Figure 5.1. Initially, games are selected and their player population series are pre-processed to handle missing data, to smooth, and to normalize. Afterwards, the life cycle pattern of each game is extracted.

In this study, the life cycle shape is determined based on the player population data of games. Product life cycles are usually determined based on sales. However, life cycle shapes of games based on other aspects can also be found in the literature, namely, genre life cycles of games based on the number of games released [2], retention related life cycle shapes based on the number of players [3] and playtime [9]. Investigating the player population-based life cycle shapes of games is useful in sev-

### 5.1. Research Procedure

---

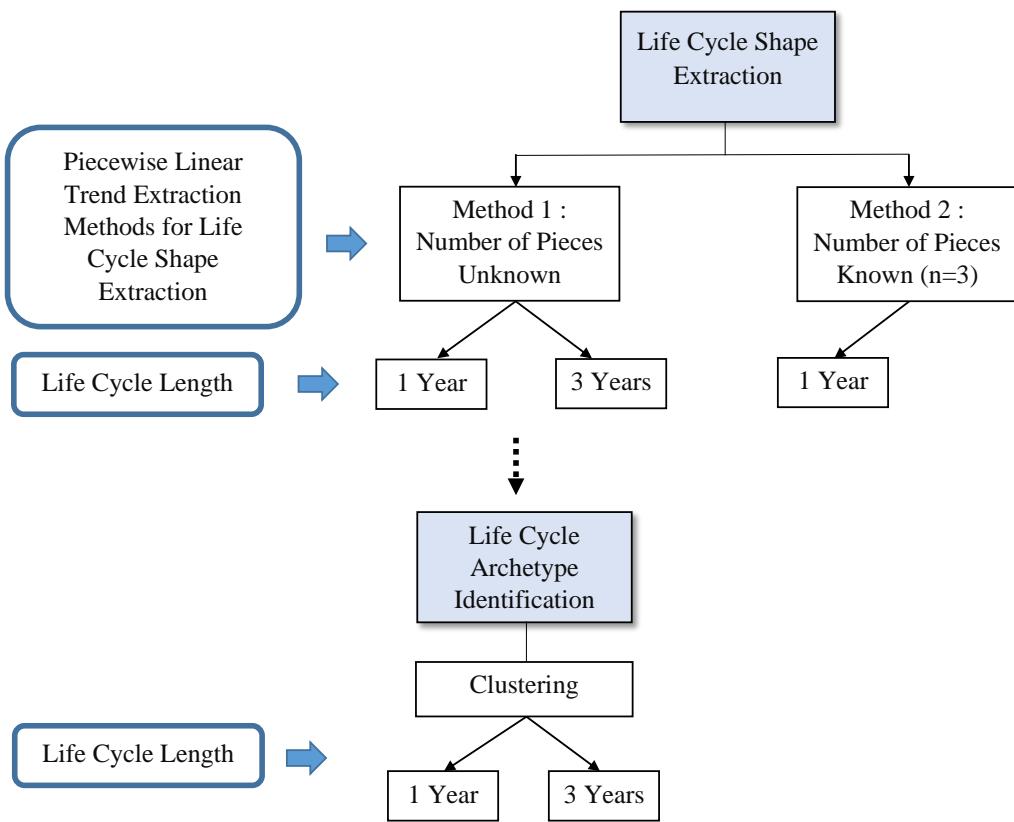


Figure 5.1: Research Procedure

eral ways. Firstly, it can represent how the popularity of a game change through time as the number of players is indicative of the popularity. Also, it can indicate the retention of the game as the changes in the population size represents how many players the game still has since game release or since a prior time step [3]. Player population-based life cycles can also represent how the actual number of players change through the lifetime of a game as sales of games do not always indicate how many players are actually playing the game despite purchasing. Hence, player population-based game product life cycle shapes, which are different from sales-based product life cycles are investigated in this study. The player population-based game product life cycle shapes are defined in the study as life cycle shapes that are extracted from the fluctuations of player population of games. The life cycle shape of a game is extracted from the player population data series of the game. Since the life cycle would have different life stages as represented by varying trends of population those need to be identified as well. Hence, two piecewise linear regression approaches

### *5.1. Research Procedure*

---

are introduced to extract the life cycle shape of a game including stages. The first approach attempts to identify the life stages, their positions within the series along with the trend of each stage. It is a type of piecewise linear regression when the number of pieces is unknown. The next approach assumes that the population life cycle of a game contains only three main stages, namely increase, saturation and decrease. The assumption somewhat matches with the classical product life cycle shape when introduction and growth are considered a single stage. This approach attempts to identify the optimal positions of the three stages along with the trend of each stage. It is a type of piecewise linear regression problem when the number of pieces is known [134]. Two approaches are used for life cycle extraction as both approaches have different limitations with respect to the number of life stages and time complexity and it is not known beforehand whether one approach would provide better outcomes or both would provide similar outcomes. Further, one provides a broader, more abstract description; while the second offers the potential of a more nuanced differentiation between the categories.

After extracting the life cycle shapes of each game through piecewise linear trend extraction approaches, clustering is conducted. It is conducted to identify archetypal life cycle shapes of games. However, the player population series of games are of varying length due to differences in release years. Hence, it becomes less interpretable and problematic to compare the similarities and differences between life cycle shapes, especially for clustering. Thus, the study is conducted based on the population data of a fixed period since game release, specifically, the first year after release and first three years after release, separately. Two periods are chosen as it provides the opportunity to closely investigate the life cycle shapes during a short period as well as a longer period. It is important as currently, to the best of our knowledge, there is no information regarding the average lifespan of video games. Also, if only the life cycles of the first three years are investigated, closer investigations on the first year would not be possible as smaller trends of the first year would not be visible on the three year life cycles. Furthermore, it is observed

that the life cycle archetypes of the first year identified using life cycle shapes extracted from the two piecewise linear trend extraction approaches are quite similar. However, the time complexity and the error of the piecewise linear fits of the second approach is higher than the first approach. Hence, only the life cycle shapes resulting from the first piecewise linear trend extraction approach is used to extract life cycle archetypes of the first year and first three years for further analysis. The characteristics of games representing each archetype are also analysed in this study to provide further insights.

### **5.1.2 Data Collection and Pre-Processing**

This section presents the details about the data collection and data pre-processing.

#### **5.1.2.1 Data Selection**

A subset of games of the *Gameset2* introduced in Chapter 3 is used for this study. Several considerations were given to the selection of games for this study. First, the games have to have a strong player base as indicated by their player population size. This criterion is important as life cycle patterns are investigated based on player population data. Hence, *Gameset2* was chosen. As explained in Chapter 3, *Gameset2* is created by removing the non-games from the 1350 Steam applications with the highest player population within the last 24 hours on 9th September 2019. Furthermore, the daily player population data series of each chosen application was downloaded from SteamDB as SteamDB contains historical player population data of games. Some more constraints were introduced as this study is focused on life cycle patterns based on player population since the release of games. The games that have no release date information available were removed. Also, games that did not have player population data since the release date of the game were removed. In addition to that, games that did not have at least one year of post-release population data were not selected. Moreover, games that were released on Early Access mode are also not used for the study as the Early Access mode allows the release of games

## *5.1. Research Procedure*

---

that are not fully developed making them different from non-early access games. Based on all the aforementioned selection criteria, the final dataset was reduced to 683 games.

### **5.1.2.2 Missing Data Handling**

As the first data pre-processing strategy, missing data were handled. The player population data collection in SteamDB was started in 2015. Population data prior to 2015 was imported by SteamDB from various sources such as *archive.org*<sup>1</sup> [135]. Due to this, player population data retrieved from SteamDB could contain missing values. However, preliminary analysis of the dataset revealed that a majority of games contain only 0- 3.1% missing data. Hence, various missing data imputation strategies for time series data such as linear interpolation, spline interpolation, nearest neighbour were explored to impute missing data. Ultimately, missing data were imputed using linear interpolation as it was capable of imputing missing data while being less impactful upon the overall shape and trend of the series compared to other approaches.

### **5.1.2.3 Data Smoothing**

Data smoothing is an important step in data pre-processing in this study as it aids in extracting high-level player population fluctuation patterns by smoothing out the low-level fluctuations. In this context, the low-level fluctuations means the small population changes such as weekly fluctuations that when removed highlights the population trends in the series which are regarded as high-level population fluctuations. Since life cycle patterns are identified based on the overall shape or trend of player population fluctuations, data smoothing is performed to highlight the trend.

First, smoothing is performed to remove weekly seasonality. The overall trend of a series can be better identified if the impact of low-level fluctuations due to weekly seasonality is removed. Thus, a moving average with a window size of 7

---

<sup>1</sup><https://archive.org/>

### 5.1. Research Procedure

---

days is applied to each player population series to extract the first approximation of the trend by excluding the impact of weekly seasonality. Here a value at a certain point of the series is approximated by calculating the mean of values within a 7 days window surrounding that data point. The moving average technique was introduced in Chapter 2. Figure 5.2 presents the moving average based data smoothing of a sample game.

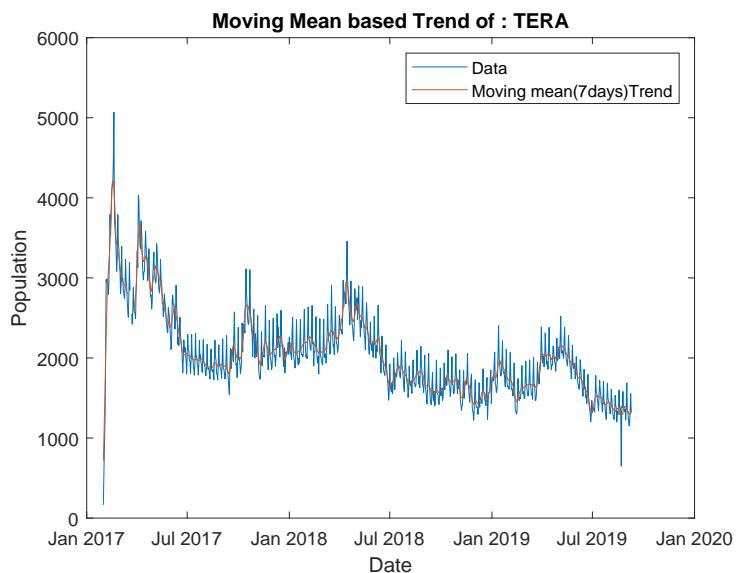


Figure 5.2: Trend of the population series of the *TERA* game extracted by removal of weekly seasonality through moving average based data smoothing

Next, further smoothing was performed using the RLOESS (Robust Locally Estimated Scatterplot Smoother) smoothing technique [136]. This smoothing step was conducted as the population series displayed further low-level fluctuations after weekly seasonality impact was removed. These could be due to the influence from various sources such as sale events, updates and twitch views. LOESS (Locally Estimated Scatterplot Smoother) is a frequently used smoothing technique for time series data. It is used in one of the prominent time series decomposition approaches, STL decomposition [106]. LOESS is a non-parametric smoothing approach based on locally weighted quadratic polynomial regression. In this approach, the smoothed value of a data point is calculated based on its neighbouring data points that lie within a user specified span. The span value is between 0-1 where 1 indicates 100% of the data of the series. The higher the span value is the smoother the series would

be. Regression weights are calculated for each data point within the span giving higher weight to data points closest to the data point to be smoothed. A quadratic polynomial function fitted to the data within the span by weighted least squares is used to smooth a data point. However, since LOESS is susceptible to the impact of outliers in the data series a robust version of LOESS, named RLOESS, has been proposed which has less sensitivity to outliers. RLOESS has an additional robust weight calculation step. More details about LOESS and RLOESS can be found in [136], [137]. Considering these facts and exploring the series smoothed using LOESS and RLOESS with span values of 0.25, 0.5 and 0.75, RLOESS was chosen with 0.5 as the span value. It was used to further smooth the previously extracted weekly seasonality removed trend series.

#### **5.1.2.4 Data Normalization**

Since the player population sizes of games vary substantially, the population series of each game are rescaled to the 0 - 1 range as per Equation 5.1 where  $x$  represents a point in the series. This is useful later in the clustering process to focus on the shape of the life cycle rather than the magnitude of the population.

$$\bar{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (5.1)$$

This section presented the data collection and preprocessing steps conducted to prepare the dataset. Details about the criteria used to select the 683 games, missing data imputation strategy, data smoothing and normalization were explained. The next section presents the methods introduced to extract life cycle shapes of each game from the population series.

### 5.1.3 Piecewise Linear Regression for Life Cycle shape extraction

This section introduces the two piecewise linear trend extraction approaches used in the study to extract life cycle shapes of each game.

#### 5.1.3.1 Method 1 - Piecewise linear regression when the number of pieces is unknown

This section introduces a piecewise linear trend extraction algorithm to extract the life cycle shape of a game without prior knowledge about the number of life cycle stages.

The main objective of the proposed algorithm is to identify the life cycle shape represented through the use of a minimum number of life stages while the overall error of the linear fits of life stages is also minimized. Since a piecewise linear regression approach is proposed for this purpose, each life stage will be represented by a linear fit of a piece identified through the least-squares method. Moreover, it is hypothesized that the best piecewise linear fit for a series is the one that has the minimum number of pieces and minimum overall error. The error is measured by Root Mean Squared Error (RMSE) depicted in Equation 5.2 where  $n$  is the length of a data series,  $\hat{y}_i$  is the value of a data point identified by the piecewise linear fit and  $y_i$  is the actual value of the data point. However, achieving such an optimal solution would not be possible as these two objectives could be contradictory. For instance, minimum RMSE for the piecewise fits could be achieved by having as many pieces as possible leading to a possible overfit. In the same way, the minimum number of pieces for a series could be achieved by sacrificing RMSE leading to a possible underfit. Since RMSE and the number of pieces of a piecewise linear fit involves a trade-off, the proposed algorithm continually controls this trade-off to obtain the best possible piecewise fit. Furthermore, heuristics are used in the algorithm to make the convergence faster while sacrificing the optimality of the final solution. Moreover, several thresholds are introduced to control the optimality of the final

piecewise fits. Algorithm 5.1 presents the proposed approach.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad (5.2)$$

Algorithm 5.1 iteratively identifies the pieces for the piecewise linear fit. The process of identifying the first piece is described here in detail. The same process is repeated from the end of the first piece to identify the rest of the pieces. An incremental window based approach is conducted as the initial step in identifying the first piece. The minimum length of the piece is chosen to be 30 days, as it is assumed a life stage length should be at least 30 days. The algorithm finds the best linear fit for the first 30 days of the series and records the RMSE. The best linear fit is the one that minimizes the residual sum of squares. This is also known as least-squares approach [138]. The window length is then incrementally increased by 1 and the best linear fit for the data within the window is identified. RMSE of the fit is recorded. This process is continued until the window end reaches the end of the series. The starting point of the window remains constant at the beginning of the series to allow the window to incrementally increase. Figure 5.3 depicts how the window length incrementally increases by 1 data point. Then the recorded RMSE values are normalized as per Equation 5.6.

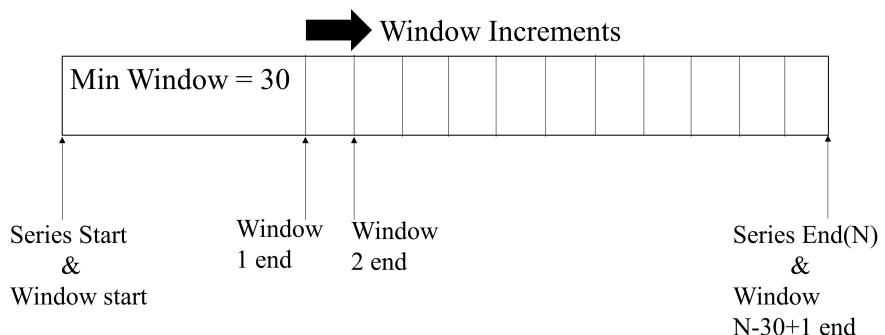


Figure 5.3: Incrementing Windows for series

---

**Algorithm 5.1** Piecewise Linear Trend Extraction : Method 1

---

```

Inputs: Xdata , Ydata ,minPieceSize, threshold
seriesLen = length(Ydata)
pieceStartI =1
pieceEndI = -1
remainingLen = seriesLen - pieceStartI+1
while (remainingLen > minPieceSize) do
    iterations = remainingLen-minPieceSize+1
    for i = 1 : iterations do
        find the best linear fit for the data subset (pieceStartI : pieceStartI +
        minPieceSize-1 +i-1)
        append error (RMSE) of the fit to errorArr
    end for
    normalize RMSE errorArr to 0-1 range
    for i = 1 : length(errorArr) - 1 do
        if  $\frac{errorArr[i+1]-errorArr[i]}{errorArr[i]}*100 \leq errorThreshold$  then
            Append errorArr[i+1] to localOptimalErrArr
            Append i+1 to localOptimalIndexArr
        end if
    end for
    if length(localOptimalErrArr) > 0 then
        optimalErr = minimum value of localOptimalErrArr
        optimalPieceEndI = index of the optimalErr value
        if length(localOptimalErrArr) - optimalPieceEndI  $\geq 1$  then
            for i = length(localOptimalErrArr) : -1 : optimalPieceEndI +1 do
                tempOptimalErr = localOptimalErrArr(i)
                errDiff = tempOptimalErr - optimalErr
                lenGain =  $\frac{(localOptimalIndexArr(i)-1)}{(length(errArr)-1)} - \frac{(localOptimalIndexArr(optimalPieceEndI)-1)}{(length(errArr)-1)}$ 
                if errDiff + (1 - lenGain)< threshold then
                    optimalErr = tempOptimalErr
                    optimalPieceEndI = i
                    break
                end if
            end for
        end if
    end if
    pieceEndI      =      pieceStartI+      minPieceSize-1      +      localOptimalIn-
    dexArr(optimalPieceEndI) -1
    save pieceStartI and pieceEndI
    pieceStartI = pieceEndI
    remainingLen = seriesLen -pieceStartI +1 ;
    pieceEndI =-1;
end while
if remainingLen  $\leq$  minPieceSize then
    merge the remainingLen to the last piece
end if
```

---

The first piece of the series could be any of the recorded windows. However, the first piece should be chosen so as to minimize the overall error of the piecewise fits and to minimize the number of pieces. Minimizing the number of pieces could be achieved by increasing the number of data points or the length of a piece. Considering the trade off between error and the number of pieces, several local optimal solutions are chosen as possible candidates for the first piece. In order to choose the local optimal solutions, the algorithm goes through the recorded RMSE values of each window. If the RMSE of  $window_{i+1}$  is less than or equal to RMSE of  $window_i$ ,  $window_{i+1}$  could be regarded as a local optimal solution. The reason is that  $window_{i+1}$  covers more length of the series than  $window_i$  while having a low error value. Thus, by choosing  $window_{i+1}$  as a local optimal solution the overall error of the series and the number of pieces of the series could be minimized.

However, there could be data series where the error for windows keeps on increasing as the window length increases. Such series would not have any local optimal solutions if the previous selection criterion is used. Hence, the criterion of local optimal solution selection is relaxed and it can be controlled by a threshold as given in Equation 5.3. A window is chosen as a local optimal piece when the error percentage in Equation 5.3 is less than or equal to the user-determined error threshold. This relaxed equation also includes the previous criterion of the error of  $window_{i+1}$  is less than or equal to the error of  $window_i$  when the error threshold is chosen to be zero. The local optimal solution space could grow as the error threshold is increased. Especially, for series where the error keeps on increasing as windows increases, the algorithm would be able to identify local optimal solutions where the rate of error increase is minimal based on the threshold. The  $errorThreshold$  value for the study is chosen to be 1% so as not to over increase the local optimal solution space which could in turn increase time complexity, and to not increase the number of pieces.

$$\frac{error_{i+1} - error_i}{error_i} * 100 \leq errorThreshold \quad (5.3)$$

Figure 5.4a depicts the RMSE values of the identified local optimal windows for

the first piece of *Monster Hunter :World* game and Figure 5.5a depicts the same for *SCP: Secret Laboratory* game. Once the local optimal solutions are identified, one solution out of these needs to be chosen as the optimal first piece. Initially, giving priority to minimizing RMSE, the window with the smallest error is chosen. This is represented by the black dot in the Figures 5.4a and 5.5a. However, there could be other local optimal solutions that could aid in minimizing the overall number of pieces of the fit while sacrificing the RMSE. For instance, in Figure 5.5a all the local optimal solutions positioned on the right hand side of the chosen solution represented by the black point could minimize the overall number of pieces as their window length is higher than the chosen solution. However, those solutions might increase the overall RMSE of the final fit. Hence, measures are taken to compare the local optimal solutions that provide length gain with the chosen solution in order to identify the optimal piece. For this purpose, the error difference and the length gain is calculated between the chosen solution and each local optimal window beginning from the rightmost. Also, a threshold as in Equation 5.4 is used to determine if there are any better optimal solutions than the chosen one based on the error and length difference.

$$\text{error difference} + (1 - \text{length gain}) < \text{threshold} \quad (0 < \text{threshold} < 2) \quad (5.4)$$

where

$$\text{error difference} = \text{normalizedRMSE}_{\text{current}} - \text{normalizedRMSE}_{\text{chosen}}$$

$$\text{length gain} = \text{normalizedLength}_{\text{current}} - \text{normalizedLength}_{\text{chosen}}$$

The RMSE values of each window and the length of each window is normalized to bring the values into the 0-1 range. This normalization step is vital for the threshold determination as it provides equal importance to the RMSE and length. Moreover, normalization helps in determining a threshold value that can be universally used for any data series rather than being unique to a single series. The length of data

### 5.1. Research Procedure

---

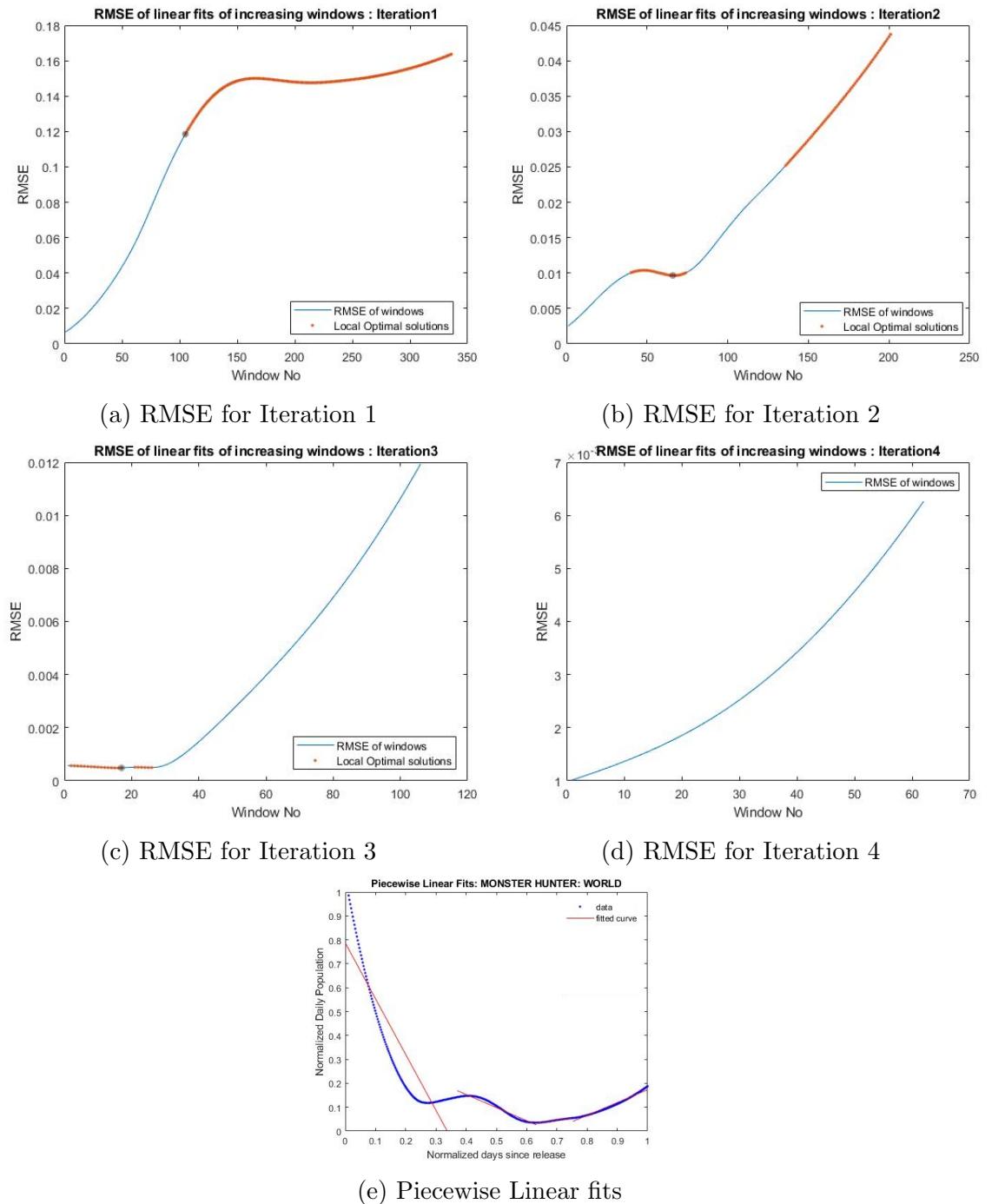


Figure 5.4: RMSE of iterations and final piecewise linear fit for *Monster Hunter: World* game

### 5.1. Research Procedure

---

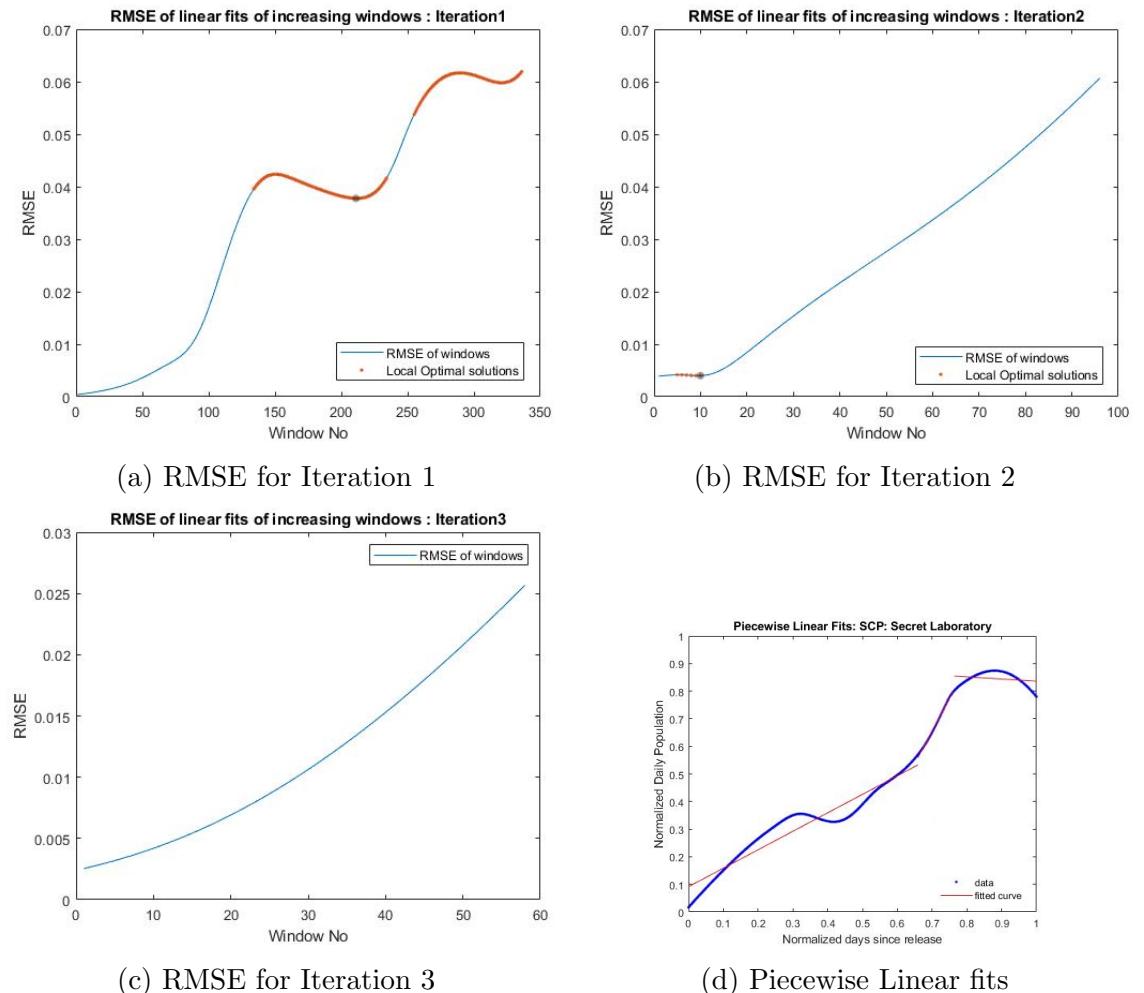


Figure 5.5: RMSE of iterations and final piecewise linear fit for *SCP: Secret Laboratory* game

points in each window is  $WindowNo + (30 - 1)$ . Thus, the window number is used in Equation 5.5 to normalize the length of windows. Also, since RMSE values are scale-dependent normalization was performed earlier as in Equation 5.6. It can be seen in Figure 5.5a that the locally optimal solutions positioned in the right hand side of the chosen solution have higher RMSE and higher length than the chosen solution. Ideally, if it is possible to identify a window that provides more length gain than the chosen solution, while having only a small increase in RMSE, it should be used as the optimal piece instead of the chosen solution. However, a numerical value is needed to represent the tolerable combination of RMSE and length. For this purpose, if there are any locally optimal solutions on the right hand side of the chosen solution, the error difference and inverse of length gain are calculated as per Equation 5.4 for those local optimal solutions starting from the rightmost one. Calculating the inverse of length gain aids in controlling RMSE and length together using a single threshold. If any locally optimal solution is encountered that agrees with the threshold, it would be chosen as the optimal solution. Going through the locally optimal solutions starting from the rightmost is important to quickly identify if there are better solutions, as length gain is highest towards the right. If there are better locally optimal solutions than the chosen one based on the threshold, one of them will be chosen as the optimal piece. If no better solution exists, the initially chosen optimal solution with the minimum RMSE will be kept as the optimal piece. Once the optimal first piece is identified, the whole process is repeated to identify the other pieces starting from the end of the first piece.

$$normalizedLength_i = \frac{windowNo_i - windowNo_{min}}{windowNo_{max} - windowNo_{min}} \quad (5.5)$$

Here  $windowNo_i$  represents the window number of the  $i^{th}$  window of the series.

$$normalizedRMSE_i = \frac{RMSE_i - RMSE_{min}}{RMSE_{max} - RMSE_{min}} \quad (5.6)$$

Here  $RMSE_i$  represents the RMSE of the  $i^{th}$  window of the series.

### 5.1. Research Procedure

---

If the remaining length of the series becomes less than 30 days at any iteration, the remaining part of the series will be combined with the last created piece and the best linear fit for that will be calculated to complete the piecewise linear fit for the series. Figure 5.4e presents the final pieces of the series and RMSE plots of each iteration for *Monster Hunter :World* game and Figure 5.5d depicts the same for *SCP: Secret Laboratory* game.

**Threshold value selection:** In order to extract piecewise linear trend using Algorithm 5.1 a threshold value needs to be chosen to indicate the RMSE and the number of pieces expected from the final fit. Hence, the RMSE and the number of pieces resulting from different threshold values are investigated and presented in Figure 5.6.

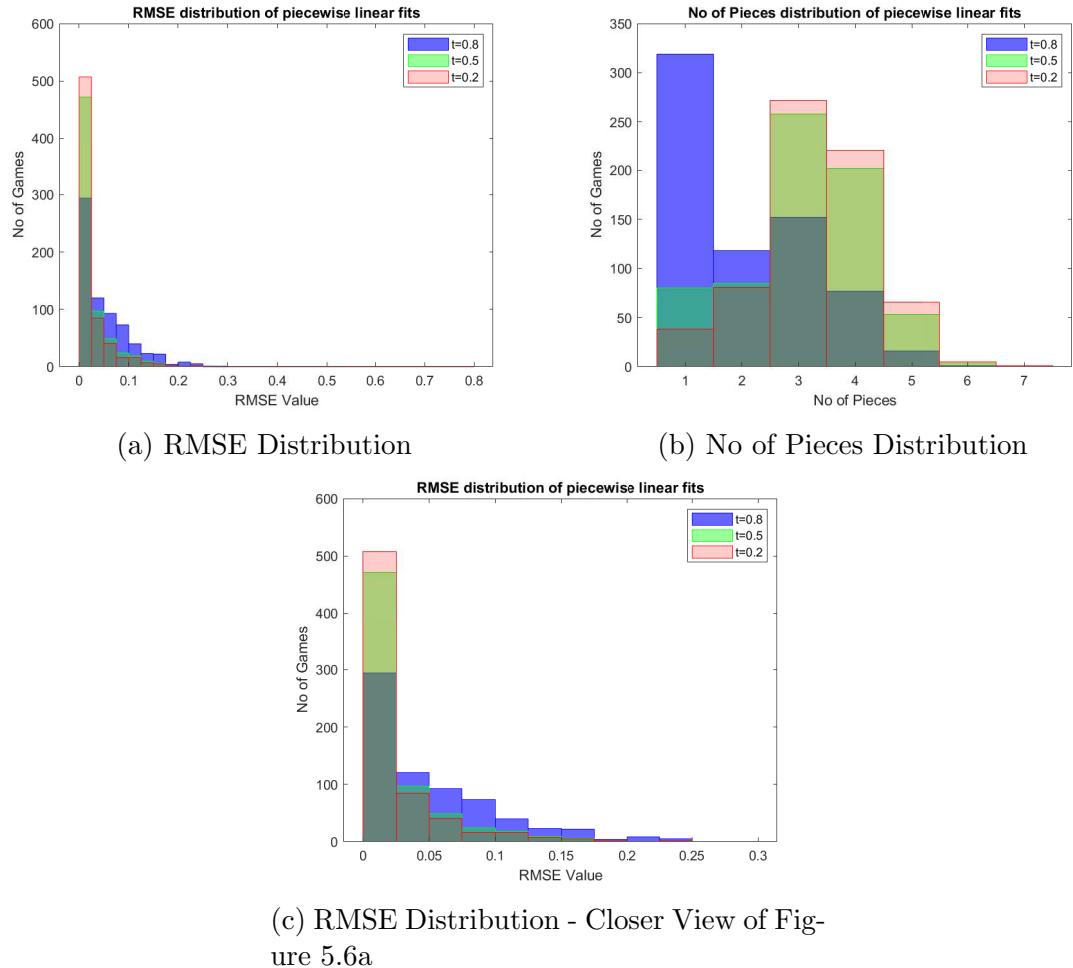


Figure 5.6: Distribution of RMSE values and No of pieces of the piecewise linear fits when threshold is chosen as  $t=0.2, 0.5, 0.8$  in Algorithm 5.1

As per Figure 5.6b the distribution of number of pieces is slightly different for both  $t = 0.2$  and  $t = 0.5$ . Closer observations show that when  $t = 0.5$  there are lesser games with more number of pieces compared to when  $t = 0.2$ . However, when  $t = 0.8$  there is a considerable decrease in the number of games with multiple pieces and a considerable increase in the number of games with a single piece. Figure 5.6a indicates that the distribution of RMSE is slightly different for both  $t = 0.2$  and  $t = 0.5$ . However, when  $t = 0.2$  there are more games with lower RMSE values compared to when  $t = 0.5$ . Moreover, when  $t = 0.8$  there are fewer games with low RMSE values compared to when  $t=0.2$  and  $t=0.5$ .

Hence, as per Figure 5.6 it can be understood that when  $t$  increases there are more games with higher RMSE and more games with fewer pieces. Thus,  $t = 0.5$  is chosen as the threshold for the piecewise linear trend extraction using Method 1 given in Algorithm 5.1 for this study.

#### **5.1.3.2 Method 2 - Piecewise linear regression when the number of pieces is known**

This section presents an algorithm for piecewise linear trend extraction for life cycle shape representation when it is assumed that there are three life stages in the life cycle.

The algorithm assumes that a game has only three stages in the life cycle and attempts to identify the optimal positions for the three stages by piecewise linear regression. For this purpose, the algorithm conducts an exhaustive search over the space of all possible combinations of three pieces. The search is conducted to find the optimal three pieces and their linear fits, while sacrificing time complexity. The optimal solution is the one that has the minimum error of the piecewise linear fit given by RMSE. The minimum piece size is chosen to be 30 days in this algorithm as well assuming a life stage should be at least 30 days long. Algorithm 5.2 presents this approach.

This section introduced two approaches for piecewise linear regression: one in

---

**Algorithm 5.2** Piecewise Linear Trend Extraction : Method 2

---

```

bestRMSE = max of series
for p1 = minPieceSize : (seriesLen- 2*minPieceSize) do
    for p2 = (p1 + minPieceSize) : (seriesLen- minPieceSize) do
        totRMSE = 0
        find fit for piece1 → 1:p1
        find fit for piece2 → p1:p2
        find fit for piece3 → p2:p3
        totRMSE = RMSE(piece1) + RMSE(piece2) + RMSE(piece3)
        if totRMSE<bestRMSE then
            record p1, p2
            bestRMSE=totRMSE
        end if
    end for
end for

```

---

which the number of pieces is unknown and the other in which it is known. The next section explains how the life cycle shapes extracted from these piecewise linear fits are used in a clustering approach to identify archetypal life cycle patterns of games.

#### 5.1.4 Life Cycle Archetype Discovery through Clustering

Archetypes of life cycle shapes of games can be obtained by a time series clustering approach. The details of the clustering approach are presented in this section.

Two subsets of the dataset are used separately in the clustering process. The player population series of games are of varying lengths. Hence the life cycle shape clustering is separately conducted using the population data of games during the first year after a game is released and first three years after a game is released. It is hypothesized that this will aid in identifying different and similar archetypes during the two periods and if there are any games transitioning between archetypes of the two periods. Piecewise linear trend in these two subsets (one year and three year data) are extracted and used in the clustering process to identify the life cycle archetypes based on shape. The two piecewise linear regression algorithms presented earlier are used separately to extract the piecewise trend. Clustering is also separately conducted using the extracted trend. It is hypothesized this will also aid in identifying the strengths and weaknesses of each algorithm for the clustering

process.

Agglomerative hierarchical clustering is chosen as the clustering technique as it does not require any prior knowledge about the number of clusters. Also, the resulting dendrogram can be used to assist in identifying the number of clusters. A linkage method has to be chosen for hierarchical clustering. Hence, three linkage methods, namely, single, average and complete linkage methods were used. The cophenetic correlation coefficient value was used to choose a linkage method out of these three. It indicates how well the distance between series is represented in the dendrogram by the linkage method. The linkage methods and cophenetic correlation coefficient was introduced in Chapter 2. Moreover, Euclidean distance is chosen as the distance measure for the clustering process since subsets of similar length series are used in the clustering process. Equation 5.7 presents the Euclidean distance formula used to measure the distance between a pair of piecewise linear trend series. In Equation 5.7,  $x$  and  $y$  represent two different piecewise linear trend series of the same length and  $n$  represents the length of the series.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5.7)$$

Once the clusters are formed the archetypal life cycle shape displayed by games in each cluster needs to be identified. For this purpose, the mean life cycle shape of all games in the cluster is extracted. The final representative shape is generated using Equation 5.8 which represents the point-wise mean of all the piecewise linear trend series of games in a cluster. The resulting shape is normalized to 0-1 scale.

$$(y_1, \dots, y_n) = \left( \frac{\sum_{i=1}^N t_{(i,1)}}{N}, \dots, \frac{\sum_{i=1}^N t_{(i,n)}}{N} \right) \quad (5.8)$$

Here  $N$  represents the number of games in the cluster,  $t_{(i,j)}$  represents the  $j^{th}$  value of the  $i^{th}$  game in the cluster,  $n$  represents the length of the series.

## 5.2 Results and Discussion

### 5.2.1 Archetypes within 1 year after game release

Clustering is conducted to reveal archetypal life cycle patterns during the first year after a game is released. The piecewise linear trend of population fluctuations of the 683 games in the dataset are extracted separately using Algorithm 5.1 and Algorithm 5.2. Hierarchical clustering results are separately presented using the trend extracted using Algorithm 5.1 and Algorithm 5.2.

#### 5.2.1.1 Archetypes resulting from Algorithm 5.1

Once the piecewise linear trend is extracted from games using Algorithm 5.1, hierarchical clustering is conducted. Table 5.1 depicts the cophenetic correlation coefficient values of the three linkage methods used in clustering. As per the results, the average linkage method has performed better than the complete and single linkage. Hence, the average linkage is chosen as the linkage method for the clustering process. The resulting dendrogram is presented in Figure 5.7.

Cophenetic Corr. Coefficient	Linkage		
	Single	Complete	Average
	0.78415	0.83767	0.85247

Table 5.1: Cophenetic Correlation Coefficient for different linkage methods; Piecewise linear trend extracted using Algorithm 5.1 from the population data of first year after game release

The dendrogram depicted in Figure 5.7 is used to determine the number of clusters. The height in the dendrogram indicates the difference between clusters at any selected cut off level for the number of clusters. As per the dendrogram, the highest difference between clusters can be observed when the number of clusters is chosen as 2 as the height is 9.8. As the number of clusters increases, the difference between the clusters decreases which is indicated by the height in the dendrogram. Rather than simply using 2 as the number of clusters, preference is given for the largest reasonable number of clusters to identify as many unique archetypes as possible.

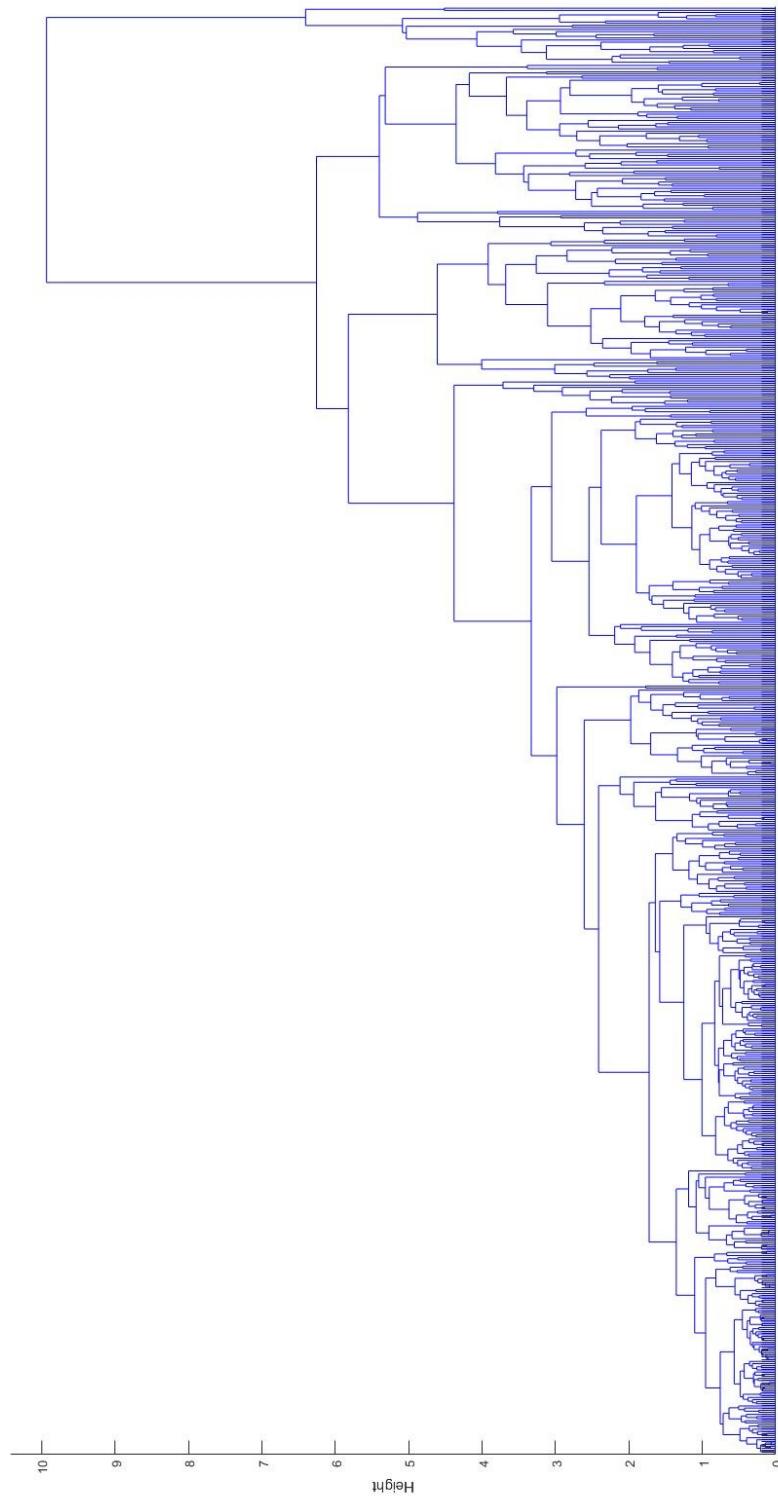


Figure 5.7: Dendrogram for hierarchical clustering; Piecewise linear trend extracted using Algorithm 5.1 from the population data of first year after game release

## 5.2. Results and Discussion

---

Thus, numbers from 2 to 10 were separately used as the number of clusters and the resulting life cycle patterns of clusters were explored to select the most appropriate number of clusters. Based on the exploration, 5 was used as the number of clusters as it provides a reasonable number of clusters with more than 10 elements in each. This can also be observed from the dendrogram. In the dendrogram when the number of clusters is chosen to be a value beyond 5, the clusters further separate making some clusters smaller. Hence, the number of clusters was chosen to be 5. However, among the 5 clusters, 1 cluster had only 2 games in it. Hence, that cluster is not considered in further analysis. The life cycle shapes displayed by games in each cluster are presented in Figure 5.8.

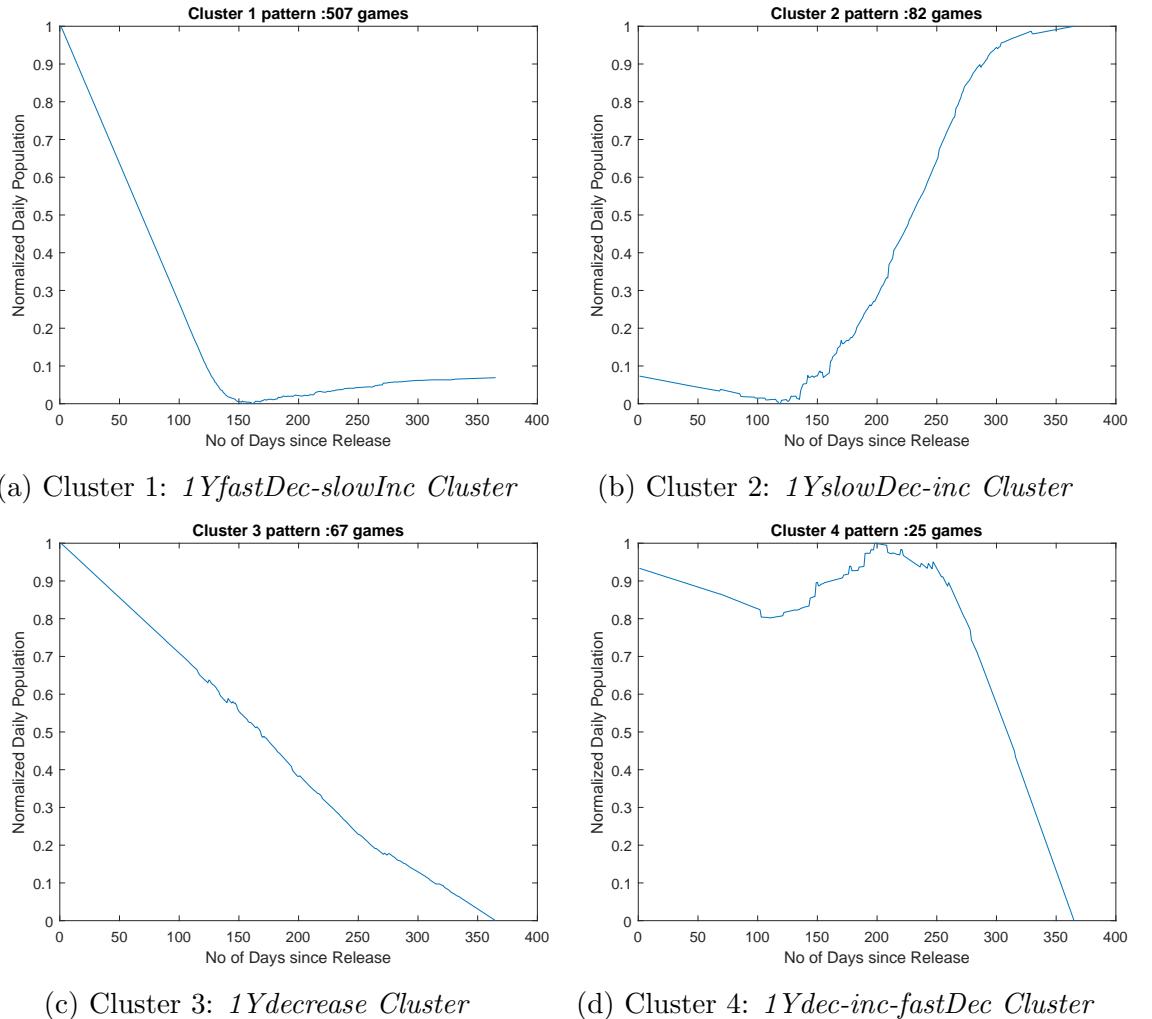


Figure 5.8: Archetypal life cycle patterns of games within the first year after release; Piecewise linear trend extracted by Algorithm 5.1

Analysing the shapes in Figure 5.8, it can be observed that the games exhibit 4

different shapes. *1YfastDec-slowInc Cluster* consists of 507 games which represent 74.2% of games in the dataset. Thus, the pattern of *1YfastDec-slowInc Cluster* seems to be the most common pattern among games. The pattern indicates that the population keeps on decreasing soon after release at a rate of 0.71% for around four months but later the population remains low increasing slowly. The rate of decrease is calculated by finding the slope of the linear fit for the considered period of the corresponding archetypal shape from Figure 5.8. *1YfastDec-slowInc Cluster* pattern indicates that for the majority of the games, the population is high soon after release. However, this does not stay the same throughout the year. The interest of players who joined the game initially would keep on decreasing resulting in decreasing of the population. Ultimately the game will be left with few players that are really interested in the game. These players will continue to play the game throughout the first year. *1YslowDec-inc Cluster* contains 82 games representing 12% of games in the dataset. The cluster represents games where the player population decreases for a while initially and starts to increase at a high rate of 0.50%. *1Ydecrease Cluster* which has 67 games representing 9.8% of games, displays a decreasing pattern. In that pattern, the population decreases throughout the year after release at a rate of 0.28% indicating a slow decay profile. *1Ydec-inc-fastDec Cluster* containing 25 games representing 3.6% of the dataset, displays a decrease and an increase during the year followed by a fast decrease of population at the end of the year. However, the rates of the initial decrease and increase are quite low with values of 0.11% and 0.13% respectively, indicating a nearly steady population during that period. The decreasing rate at the end of the year displayed during the last 3 months is relatively high depicting a rate of 0.86%.

### **5.2.1.2 Archetypes resulting from Algorithm 5.2**

The same clustering process was conducted using trend extracted from Algorithm 5.2 in order to compare the outcomes of Algorithm 5.1 and Algorithm 5.2. The cophenetic correlation coefficient values of the different linkage methods used in

the clustering process is presented in Table 5.2. Based on the results, the average linkage has performed better compared to other linkage methods. Also, in the previous clustering process in which trend from Algorithm 5.1 was used, average linkage proved to be better than other linkage methods. The dendrogram of this clustering process is presented in Figure 5.9.

Cophenetic Corr. Coefficient	Linkage		
	Single	Complete	Average
	0.82159	0.80005	0.86103

Table 5.2: Cophenetic Correlation Coefficient for different linkage methods; Piecewise linear trend extracted using Algorithm 5.2 from the population data of first year after game release

As per the dendrogram in Figure 5.9, the highest difference between clusters can be observed when the number of clusters is 2 as that solution has the largest height in the dendrogram. However, as previously, numbers from 2 to 10 were used separately as the number of clusters and the resulting life cycle cluster solutions were explored to select the most appropriate number of clusters. Based on that exploration 3 was used as the number of clusters. The representative patterns of each cluster is given in Figure 5.10.

As per the life cycle archetypes in Figure 5.10 one dominating shape exists represented by *1YfastDec-slowDec Cluster* which contains 575 games in it. That pattern depicts that the population decreases soon after release for around four months at a high rate of 0.74% and then saturates with a slow decrease. The same pattern was observed in the previous clustering process when Algorithm 5.1 was used to extract the life cycle shapes. *1Ydec-flat-inc Cluster* contains 93 games and represents a pattern where the population initially decreases and flattens but later starts increasing faster at a rate of 0.61%. This pattern resembles the *1YslowDec-inc Cluster* pattern of Algorithm 5.1 to some extent. *1Yinc-slowDec-fastDec Cluster* contains 15 games and depicts a pattern where the population increases slowly initially but later slowly decreases until it decreases fast at the end of the year at a rate of 1.12%. This pattern is somewhat similar to *1Ydec-inc-fastDec Cluster* pattern of Algorithm 5.1

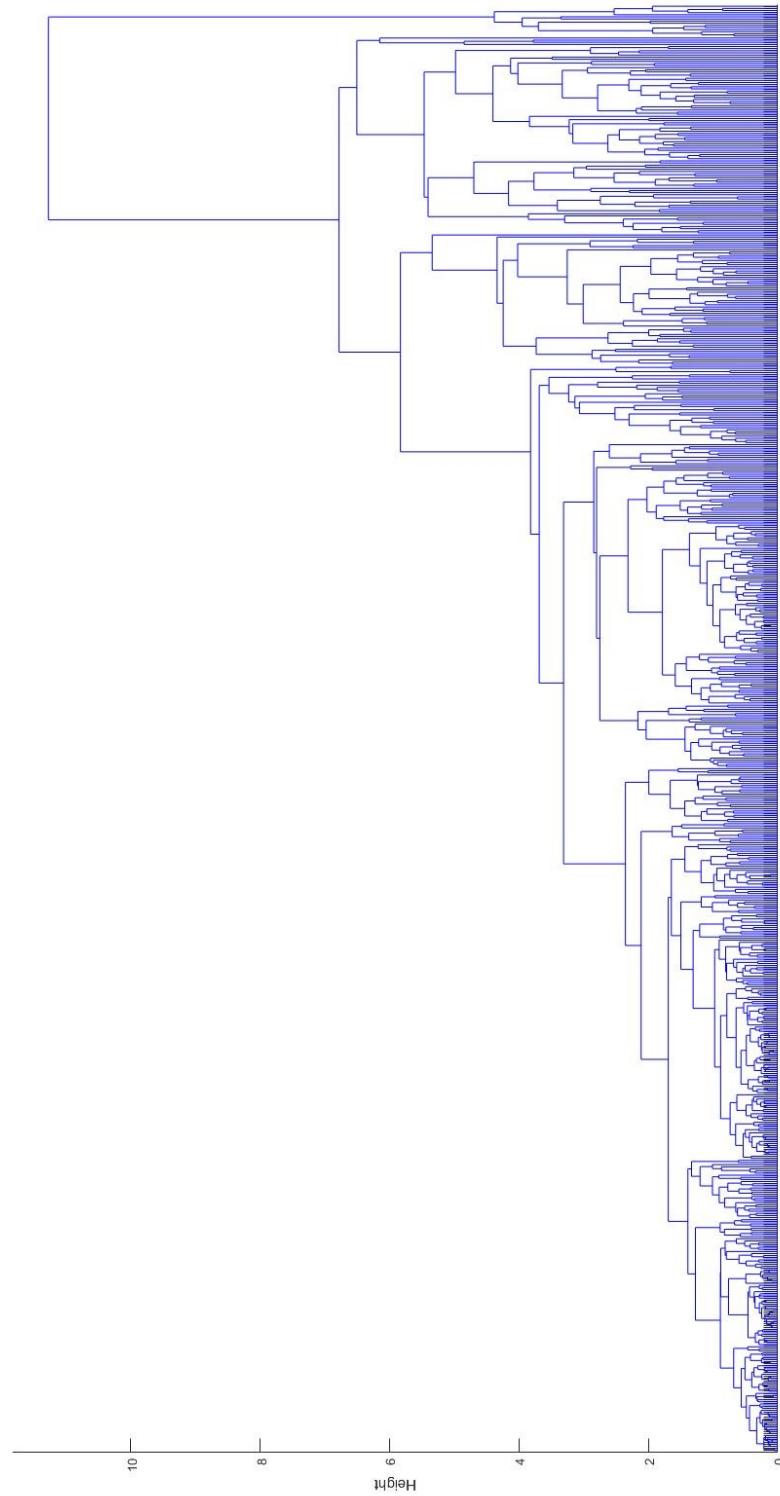


Figure 5.9: Dendrogram for hierarchical clustering; Piecewise linear trend extracted using Algorithm 5.2 from the population data of first year after game release

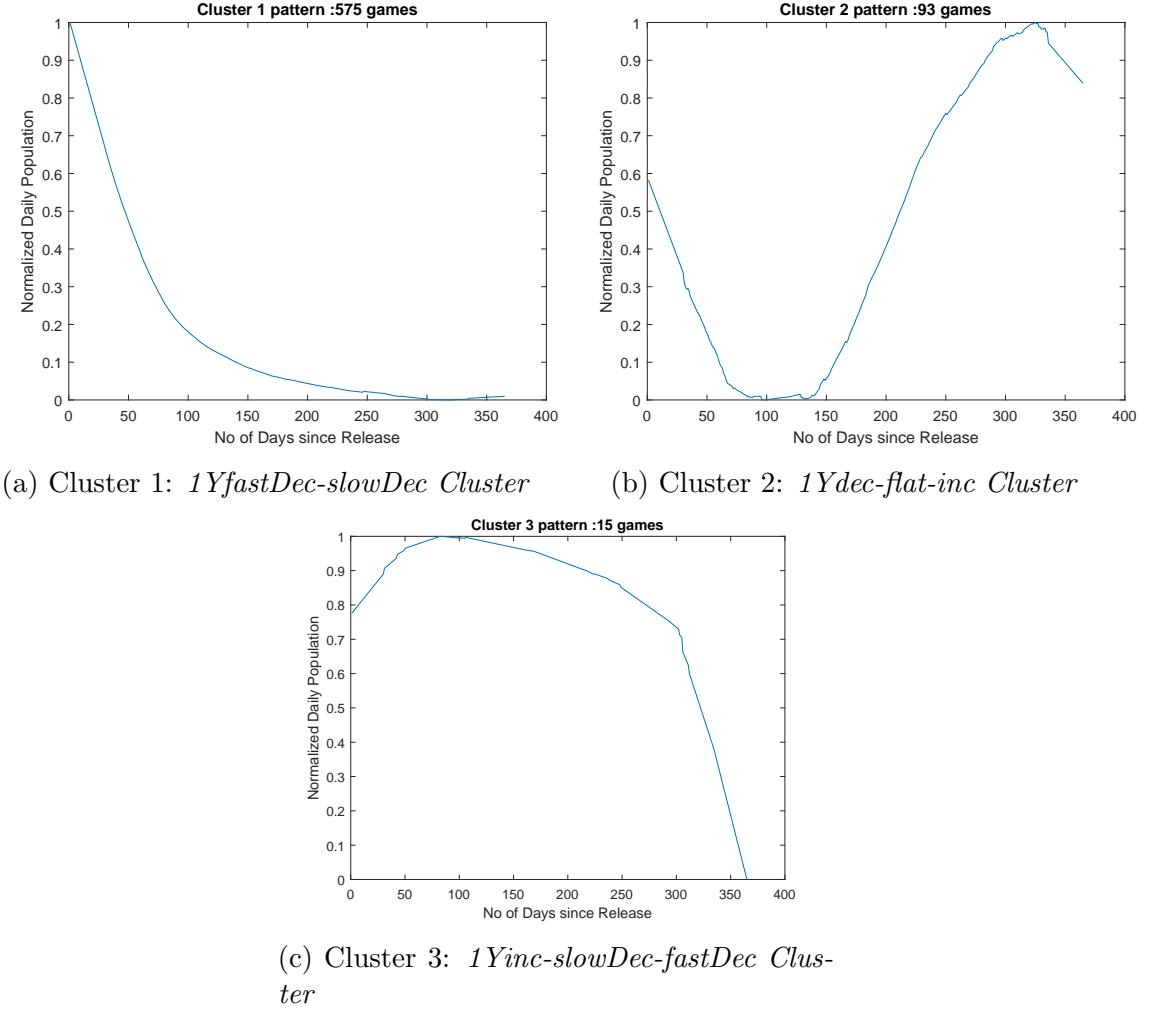


Figure 5.10: Archetypal life cycle patterns of games within the first year after release; Piecewise linear trend extracted by Algorithm 5.2

based cluster approach.

Additionally, the two piecewise linear regression algorithms and the resulting archetypes were compared to understand their differences and similarities. As observed earlier, similarities exist in the life cycle archetypes resulted from both algorithms. However, the main difference is the lack of the *1Ydecrease Cluster* pattern resulted from Algorithm 5.1, among the clusters resulted from Algorithm 5.2. It represented a decreasing population pattern. Moreover, the RMSE values of piecewise linear fits resulted from Algorithm 5.1 and Algorithm 5.2 were compared. A right-tailed t-test indicated that the mean of the RMSE values of the fits resulting from Algorithm 5.2 is statistically significantly higher than the same from Algorithm

5.1 indicated by a P-value of 2.4787e-99 at 5% significance level. The main reason behind this could be the flexibility in the number of pieces in Algorithm 5.1. Moreover, the time complexity to extract piecewise linear trend fit using Algorithm 5.2 is higher than Algorithm 5.1 due to the exhaustive searching approach. Due to these reasons, Algorithm 5.1 is chosen as the piecewise linear trend extraction approach for the cluster analysis in the rest of the study.

This section presented the life cycle archetypes during the first year after game release. Two sets of archetypes generated using Algorithm 5.1 and Algorithm 5.2 were presented and their differences and similarities were discussed. The next section presents the life cycle archetypes during the first three years after game release.

### **5.2.2 Archetypes within 3 years after game release**

The same clustering process was repeated in order to identify the archetypal life cycles displayed within the first three years after a game is released. This aids in thoroughly analysing life cycle shapes games display and compare between the first year and first three years of a game's life cycle. Games that had three years of player population data since release date were chosen which resulted in 411 games. Moreover, RLOESS data smoothing is conducted choosing 0.2 as the span parameter. Earlier when smoothing population data of the first year, 0.5 was chosen which represented around 6 months of data span as 0.5 means 50% of the series. In order to match with that smoothness, 0.2 was chosen as the span for three years data. Moreover, as mentioned earlier piecewise linear trend extraction is performed using Algorithm 5.1. Apart from these changes, the same clustering process was conducted.

The cophenetic correlation coefficient values of the three linkage methods for hierarchical clustering is depicted in Table 5.3. As before, the average linkage method has performed better. The resulting dendrogram is provided in Figure 5.11. As before, numbers from 2 to 10 were used as the number of clusters and the resulting life

cycle clusters were explored to select the most appropriate number for the number of clusters. Based on that, 6 was chosen as the number of clusters to identify as many archetypes as reasonably appropriate. However, 2 clusters of that cluster solution contained only 2 games in each. Hence, those two clusters are removed from further analysis. The archetypal life cycles of the chosen 4 clusters are depicted in Figure 5.12.

	Linkage		
	Single	Complete	Average
Cophenetic Corr.	0.82713	0.82603	0.8706
Coeffcient			

Table 5.3: Cophenetic Correlation Coefficient for different linkage methods; clustering life cycle shapes of three years

As per Figure 5.12 the most common life cycle shape is displayed by the 333 games in *3YfastDec-slowDec Cluster* containing 81% games in the dataset. It displays a pattern where the population decreases initially for around 6 months at a rate of 0.38% and slowly decreases at a rate of 0.03% afterwards. *3Yincrease Cluster* contains 44 games, representing 10% of games in the dataset and displays a pattern where population increases throughout the three years. *3Ydecrease Cluster* displays a decreasing population pattern throughout the three years with slight fluctuations. It contains 17 games representing 4% of games in the dataset. Moreover, *3Y3stageDec Cluster* representing 2% of games also displays a decreasing pattern. However, the speed of decrease is slower initially at a rate of 0.11% for a period of 7 months and then the speed increases afterwards for a period of over a year and then slows down.

So far, the life cycle archetypes of one year and three years were presented. For the sake of simplicity, the life cycle clusters during the first year after game release resulted from Algorithm 5.1 would be referred to as *1Yclust*. Also, the life cycle clusters during the first three years after game release resulted from Algorithm 5.1 would be referred to as *3Yclust* hereafter in the chapter as appropriate. The next section provides a discussion on the comparison between the archetypes of *1Yclust* and *3Yclust*.

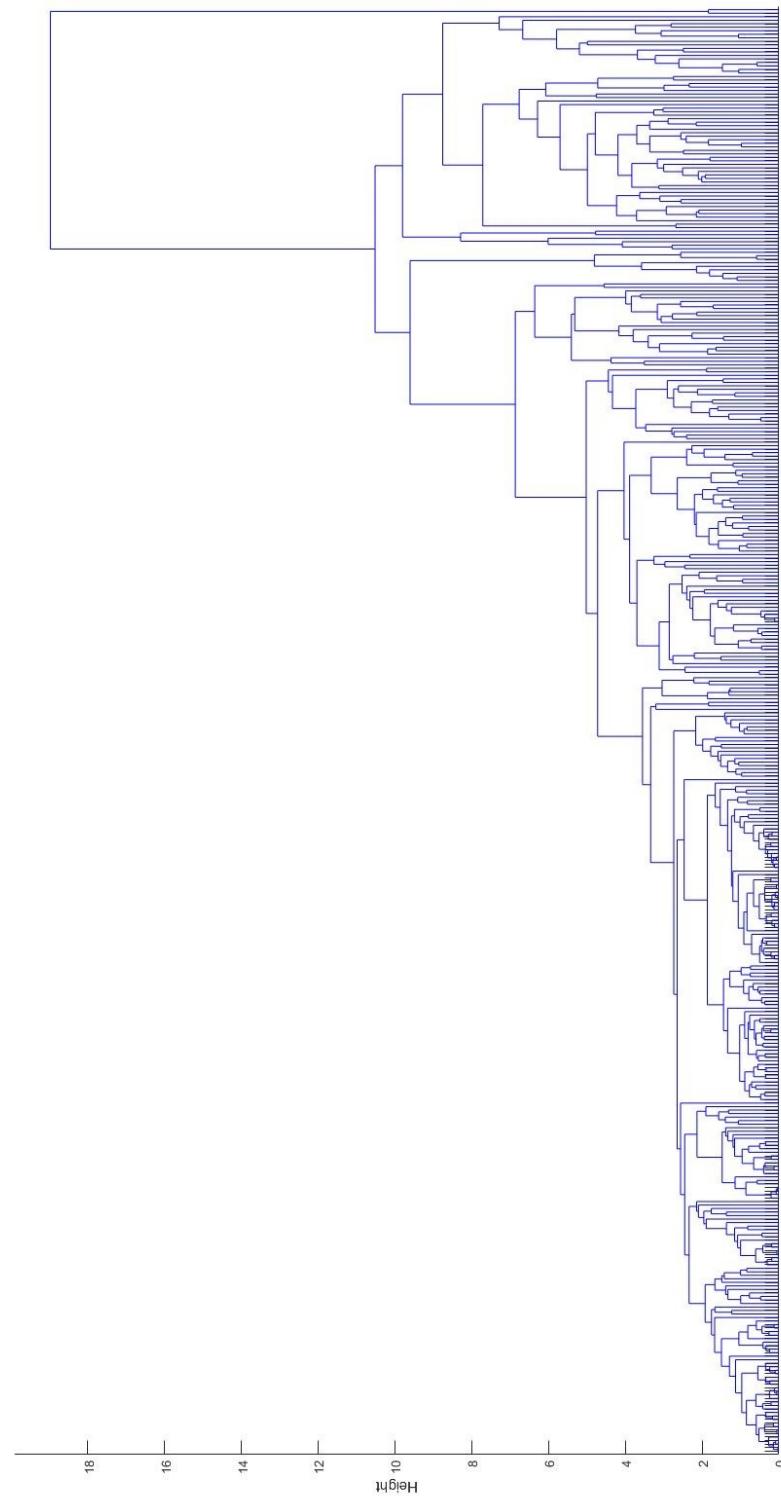


Figure 5.11: Dendrogram for hierarchical clustering of life cycles of first three years after game release

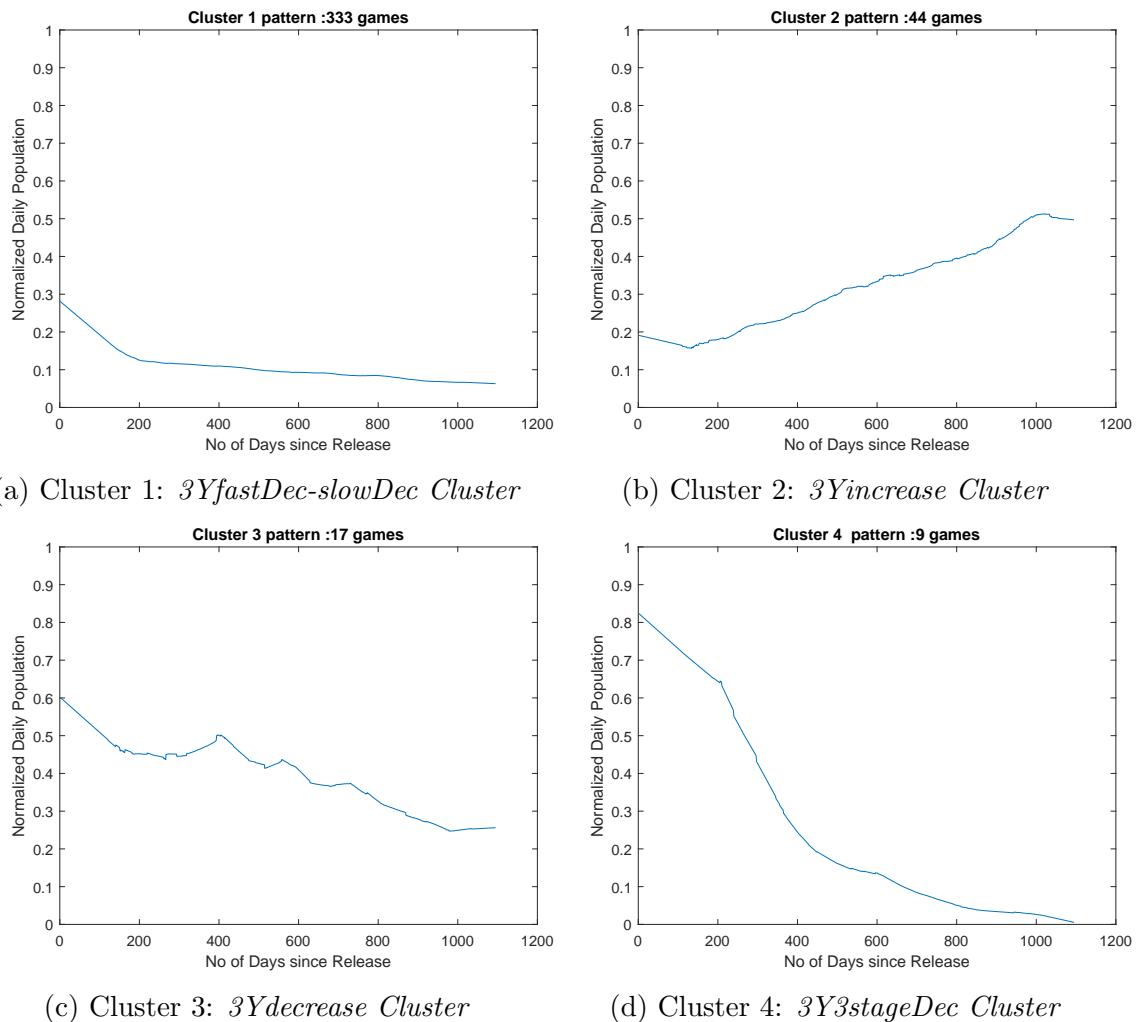


Figure 5.12: Archetypal life cycle patterns of games within the first three years after release

### 5.2.3 Archetype Comparison

In this section, the archetypes from  $1Yclust$  and  $3Yclust$  are compared with respect to the shapes and the games displaying each archetype.

A comparison between the archetypal shapes of  $1Yclust$  and  $3Yclust$  revealed several similarities and differences. Figure 5.13 depicts each pair of archetype from  $1Yclust$  and  $3Yclust$  for comparison. The difference between each pair of archetypes during the first year is also presented in Figure 5.13 by calculating the absolute difference between each data point during the first year. The absolute difference between two numbers,  $x$  and  $y$  is  $|x - y|$ . The pairs of archetypes of  $1Yclust$  and  $3Yclust$  compared are:

- Cluster 1 of  $1Yclust$  and  $3Yclust$ :  $1YfastDec-slowInc$  Cluster and  $3YfastDec-slowDec$  Cluster
- Cluster 2 of  $1Yclust$  and  $3Yclust$ :  $1YslowDec-inc$  Cluster and  $3Yincrease$  Cluster
- Cluster 3 of  $1Yclust$  and  $3Yclust$ :  $1Ydecrease$  Cluster and  $3Ydecrease$  Cluster
- Cluster 4 of  $1Yclust$  and  $3Yclust$ :  $1Ydec-inc-fastDec$  Cluster and  $3Y3stageDec$  Cluster

The Cluster 1 shape of both  $1Yclust$  and  $3Yclust$ , namely,  $1YfastDec-slowInc$  Cluster and  $3YfastDec-slowDec$  Cluster, were similar overall. However, in  $1YfastDec-slowInc$  Cluster population has decreased throughout the first 4 months while in  $3YfastDec-slowDec$  Cluster it is for the first 6 months. The rate of population decrease during those periods were 0.71% for  $1YfastDec-slowInc$  Cluster and 0.38% for  $3YfastDec-slowDec$  Cluster indicating that the number of players decreases faster in  $1YfastDec-slowInc$  Cluster. These rates were identified from Figure 5.13 by finding the slope of the linear fit for the population during the considered periods in the corresponding cluster pattern. The Cluster 2 shape of both  $1Yclust$  and  $3Yclust$ , namely,  $1YslowDec-inc$  Cluster and  $3Yincrease$  Cluster, also have a similarity as

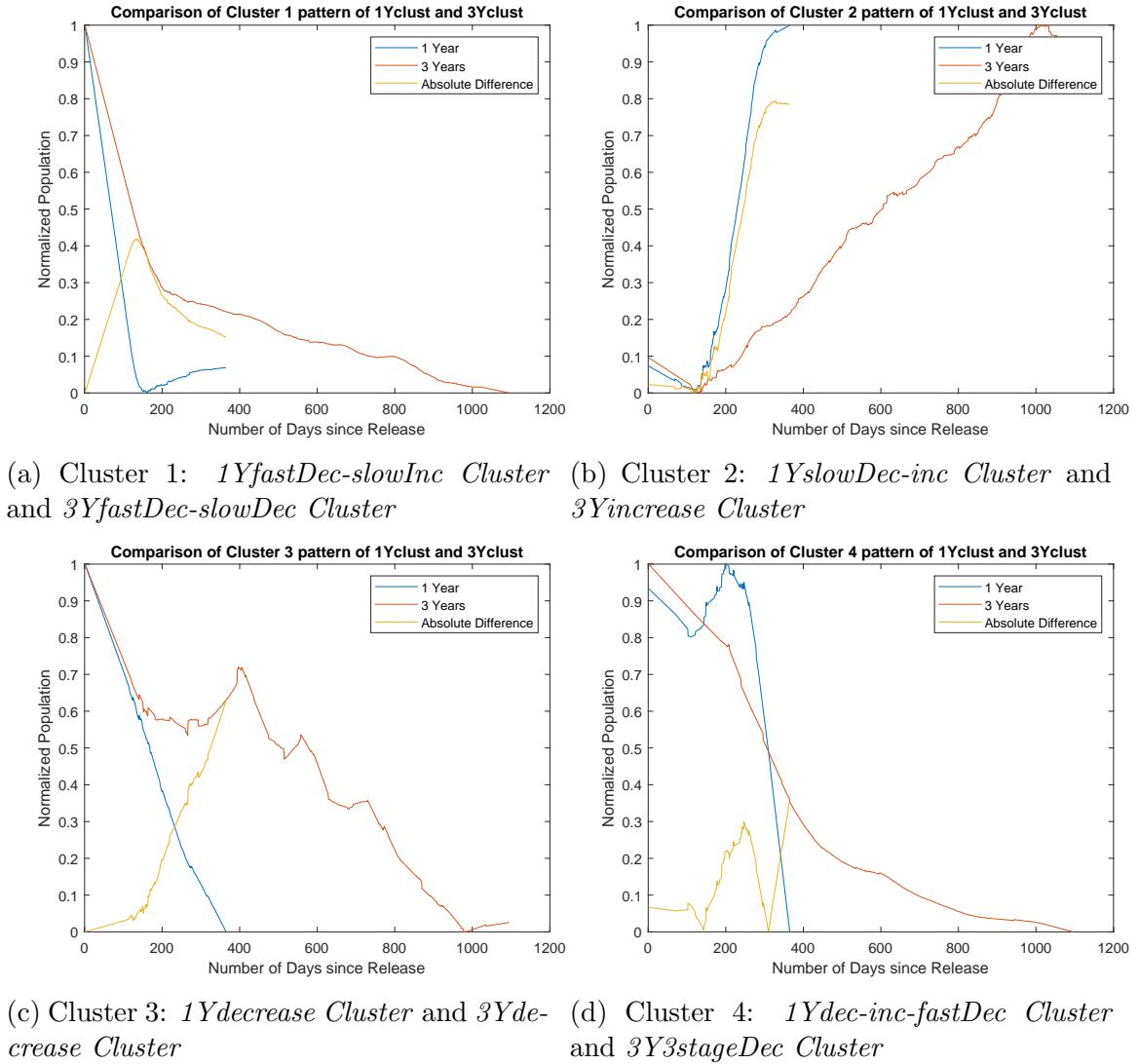


Figure 5.13: Difference between the one year and three years life cycle patterns within the first year after release

both represent an increasing population pattern. The pattern initially has a period of decrease for the first 4 months in both *1YslowDec-inc Cluster* and *3Yincrease Cluster* with minimal difference between them. It is followed by an increasing phase in both *1YslowDec-inc Cluster* and *3Yincrease Cluster*, however, with different rates of increase. Specifically, in *1YslowDec-inc Cluster*, the rate is 0.50% whereas in *3Yincrease Cluster* it is 0.09% indicating a faster growth rate in *1YslowDec-inc Cluster*. The shape of Cluster 3 in both *1Yclust* and *3Yclust*, namely, *1Ydecrease Cluster* and *3Ydecrease Cluster*, represents a decreasing pattern. However, the pattern of *3Ydecrease Cluster* has some smaller fluctuations after the first 6 months while the pattern of *1Ydecrease Cluster* continues to decrease. The Cluster 4 pattern of *1Yclust* and *3Yclust*, namely, *1Ydec-inc-fastDec Cluster* and *3Y3stageDec Cluster*, is slightly different. However, a faster population decrease rate can be observed at the end of first year in both *1Ydec-inc-fastDec Cluster* and *3Y3stageDec Cluster* where it is a 0.86 % decrease rate in *1Ydec-inc-fastDec Cluster* during the last 3 months while it is 0.26% in *3Y3stageDec Cluster* during the last 5 months. Moreover, the ranking of clusters based on the percentage of games each cluster holds were similar across both *1Yclust* and *3Yclust*. In general, it can be understood that *1Yclust* and *3Yclust* have similarities and *3Yclust* archetypes could be a possible extension of the *1Yclust* archetypes.

The games displaying each archetype of *1Yclust* and *3Yclust* were analysed to further understand the life cycle patterns. Especially, to understand transitions of games from one year archetypes to three year archetypes. Hence, the percentage of games common in each pair of clusters in *1Yclust* and *3Yclust* were calculated. As mentioned earlier, only 411 games were used in generating the *3Yclust* as those were the only games that had three years of population data out of the 683 games used in generating *1Yclust*. Hence the percentage is calculated relative to the number of games in each cluster in *1Yclust* that was also used in *3Yclust* generation process. Table 5.4 depicts the calculated percentages where each cell represents the percentage of games common between each considered cluster pair from *1Yclust* and

$3Yclust$ .

		3Yclust			
		Cluster 1: 3YfastDec- slowDec Cluster	Cluster 2: 3Yin- crease Cluster	Cluster 3: 3Yde- crease Cluster	Cluster 4: 3Y3stageDec Cluster
<b>1Yclust</b>		Cluster 1: 1YfastDec- slowInc Cluster	95.30	4.02	0.67
		Cluster 2: 1YslowDec- inc Cluster	40	48	10
		Cluster 3: 1Ydecrease Cluster	35.82	13.15	13.15
		Cluster 4: 1Ydec-inc- fastDec Cluster	29.41	17.64	29.41
					23.52

Table 5.4: Game transition percentages from first year archetypes to three year archetypes

As per Table 5.4, 95.3% of games from *1YfastDec-slowInc Cluster* have appeared in the *3YfastDec-slowDec Cluster* indicating that the *3YfastDec-slowDec Cluster* pattern is an extension of the *1YfastDec-slowInc Cluster* life cycle pattern. It means that games that display a life cycle where the population starts from a high value and continue to decrease for around 4 months and then remain low for the rest of the year would continue to remain low during the rest of the three year period as well. This is the most common life cycle archetype that the majority of games display. The majority of games that display the increasing population pattern of *1YslowDec-inc Cluster* would continue to display the same pattern in *3Yincrease Cluster*. However, 40% of games would transition to the *3YfastDec-slowDec Cluster*, probably due to a possible disruption to the growth. Moreover, a majority of games displaying the *1Ydecrease Cluster* pattern transitions to the *3YfastDec-slowDec Cluster* pattern. This indicates that games that displayed a slow decaying profile in which population decreases slowly throughout the first year would later

observe a further slower decrease for the rest of the three year period. This indicates the existence of both slow decaying and fast decaying population life cycles of games. The *1Ydec-inc-fastDec Cluster* pattern has a similar percentage of transitions to *3YfastDec-slowDec Cluster* and *3Ydecrease Cluster*. It also has 23.52% of games staying in the same corresponding cluster, which is *3Y3stageDec Cluster*.

Considering the shape similarities and percentage of common games, it can be inferred that the three years life cycle archetypes are a possible extension of the first year archetypes, especially for Cluster 1 (*1YfastDec-slowInc Cluster* and *3YfastDec-slowDec Cluster*) and Cluster 2 (*1YslowDec-inc Cluster* and *3Yincrease Cluster*) archetypes of *1Yclust* and *3Yclust*. It indicates that most games that display each first year archetype would later show the corresponding three year archetypes. Furthermore, game transitions between archetypes are also common.

This section presented the details of the life cycle shape clustering process and presented the archetypal life cycle patterns of games within the first year and first three years after game release. The next section presents a thorough analysis and discussion of the archetypal patterns by investigating the game characteristics in each cluster.

#### **5.2.4 Analysis of Game Characteristics**

Analysing the games in each cluster could provide more insights about the game life cycle patterns. Hence, this section provides insights related to the characteristics of games displaying each archetypal life cycle pattern. First, the process related to the analysis of game characteristics is explained. Then the outcomes are presented.

The game characteristics investigated are the tags of games, release year, release month, mean population, publishers, developers, price and reviews of games. These are investigated separately for *1Yclust* and *3Yclust* archetypes. The top 20 tags assigned to each game in Steam are used to analyse the tags in clusters. For each cluster, the percentage of games assigned to each tag is calculated. Since each cluster contains a large number of tags (Eg: Cluster 1 has 380 tags, Cluster 4 has 120 tags),

a subset of tags that is representative of the cluster has to be selected for further analysis. The tags that represent at least 20% of games in the cluster is selected for that purpose. A percentage of 20% was chosen as it was identified that percentages higher than that result in many tags that are common among all clusters making it ineffective to identify tags unique to clusters, and percentages less than that results in some tags that are assigned to only a few games in the cluster making them not representative of the clusters. Hence, after selecting the tags that represent at least 20% of games, the percentage of games associated with each tag is compared between clusters and between the dataset and each cluster to generate insights. Moreover, histograms of release year distribution and release month distribution are used to analyse the release date related characteristics. Release month could be used to determine if a game was released during a major sale event period of Steam, such as the winter sale and summer sale. Furthermore, the mean population of each game during the considered time period of life cycle, which is either 1 year or 3 years, is analysed by exploring box plots. Also, publishers and developers of each game are extracted from Steam. The number of games in each cluster associated with each publisher is calculated and the top 10 publishers in each cluster are analysed. The same process is conducted for the developer. When it comes to price analysis, the most frequent price of a game during the 1 year or 3 years period is first extracted as price fluctuates due to discounts throughout the game life cycle. Histograms depicting the price distribution of games in each cluster are analysed. Moreover, the positive review percentage assigned to each game in Steam is extracted and its distribution is used in the analysis. Apart from analysing the histograms, the difference between clusters with respect to release year, release month, price and positive review percentage is numerically measured by calculating the Euclidean distance between histograms using the percentage of games assigned to each bin. This game characteristic analysis process is repeated for both the  $1Yclust$  and  $3Yclust$ . Furthermore, the game characteristics of each cluster are compared with the game characteristics of the  $gameset2$  as well to provide insights regarding

the similarity and differences of the games in each cluster compared to the overall set of games used in the study. The findings of the analysis related to each cluster in *1Yclust* and *3Yclust* are presented in the following sections. The charts used for analysis along with a description can be found in Appendix A.

#### **5.2.4.1 Game characteristics of Life Cycle Archetypes during First year after release (*1Yclust*)**

This section presents the game characteristics observed in clusters of *1Yclust*. In general, it was observed that most characteristics do not significantly distinguish games between each cluster. A large number of tags were present but some common tags could be found. Moreover, Publishers and Developers were too diverse to find any common publisher or developer from each cluster.

***1YfastDec-slowInc Cluster:*** This archetype has a high preponderance of Action, Adventure and Open World games. All games have been released after the year 2009 in this cluster. It is worth noting that when the overall game set is considered, there are games released since 2004. Interestingly, a lesser number of games have been released in June, July, December and January months in this cluster compared to other months. A similar pattern is observable in the overall game dataset as well. The most popular Steam summer sales and winter sales are usually held in these months. Thus, it seems that most games in this archetype have been released avoiding major summer, winter sale events. However, around 80% of games are non-Free-to-Play games in this cluster, which is 3% higher than the non-Free-to-Play games in the dataset. The most common price range was \$10-20 for this archetype representing 30% of the games. The overall game set also has the highest percentage of games in the \$10-20 price range which makes this result unsurprising as this cluster has the highest percentage of games from the gameset. Also, compared to other clusters, this cluster has the highest percentage of games in the \$10-20 price range. All games are priced less than \$60 except for one game. Most of the games of this archetype are loved by players as more than half of the

games (62%) had 80-100% positive reviews. The overall game set also has a similar distribution.

***1 YslowDec-inc Cluster:*** Strategy and Simulation games seem to appear higher in this cluster. *DOTA 2*, *World of Tanks Blitz*, *Black Desert Online*, *Train Simulator*, *Tabletop Simulator* are a few of the games of this archetype. Most games of this cluster have been released with almost the same percentages from 2012 - 2018 except for the year 2016. But in the overall gameset the percentage of games released every year is increasing. Moreover, November, December, February and March seem to be the popular release months while other months also have lesser numbers of games released from this cluster. November is the most popular release month in the entire gameset also. But there are other months more popular than December, February and March in the gameset. All the games are priced less than \$60 while the majority (51%) is less than \$30. However, the overall gameset has few games priced higher than \$60. Moreover, the largest difference between price distributions was observed between this cluster and *1Ydec-inc-fastDec Cluster* where the Euclidean distance between the histograms was 7.75. Also, more than 70% of the games are non-Free-to-Play games, whereas it is 77% for the game set. Also, most of the games (86%) have more than 60% positive reviews. The overall gameset has 89% games with more than 60% positive reviews. The difference between the distributions of positive review percentage was smallest between this cluster and *1YfastDec-slowInc Cluster* with a Euclidean distance of 1.71.

***1 Ydecrease Cluster:*** The majority of games in this cluster are Survival and Casual games. For instance, *Dead by Daylight*, *Clicker Heroes*, *Crusaders of the Lost Idols* and *The Hunter Classic* are some games in this cluster. Most games (83%) in this archetype have been released after 2014. In the overall gameset only 76% games are released after 2014. The smallest difference between release year distributions were observed between this cluster and *1YfastDec-slowInc Cluster* with a Euclidean distance of 3.34 indicating their similarity in release year distribution. The most popular release months seem to be June and September-December. But in the

overall game set, June and December are less popular release months. Interestingly, the popular release months in this cluster coincides with some major steam sale events, which are Halloween, Autumn, Winter and Summer Sales. However, this cluster has almost equal percentages of Free-to-Play games and non-Free-to-Play games which is different from the proportions in the full gameset. All games are priced less than \$60, while the most common price range is \$10-20 depicting 15% of games. The \$10-20 price range is the most common price range in the overall gameset as well, but it represents 30% of the games in the complete gameset. The smallest difference between price distributions was observed between this cluster and *1Ydec-inc-fastDec Cluster* with a Euclidean distance of 4.64. When it comes to reviews, most games seem to have positive reviews of 60-80%. When the overall game set is considered, the number of games that have 80-100% positive reviews is 34% higher than the number of games that have 60-80% positive reviews. However, the number of games that have 80-100% positive reviews are 6% less than that of those in the 60-80% range in this cluster. This indicates that not many games displaying this archetype have received a high level of positive reviews compared to other clusters. This could be one reason for the decreasing pattern this archetype displays.

***1 Ydec-inc-fastDec Cluster:*** Most games that display this pattern were Sports, Football and some shooter games. *Football Manager 2015, 2016, 2017, 2018, Pro Evolution Soccer 2016, NBA 2K15* are some of those sports games. Release year distribution indicates that the games have been released each year in almost equal amounts. This agrees with the yearly released versions of sports games appearing in this cluster. It seems that a new game such as *Football manager* is released every year. The pattern in this archetype explains how players' interest drops when a new game of the same title is released every year. Moreover, the games of previous years may become unavailable for sale when the new game of the current year is released. Thus, making it available to only the past purchasers. The most popular month for release seems to be November where 36% of games are released.

November is the most popular release month in the overall gameset as well. Also, over 84% of games in the cluster are released from September to December. This period also coincides with the US National Football League and National Basketball League. The difference between the distribution of release months is highest between this cluster and *1YfastDec-slowInc Cluster* with a Euclidean distance of 9.85. Also, games of this cluster had the highest median mean population. Furthermore, this archetype has almost the same percentages of Free-to-Play and non-Free-to-Play games and the prices are less than \$20. But in the overall gameset there are games priced higher than \$20. The positive reviews percentage is between 40 - 100% for games in this cluster.

#### 5.2.4.2 Game Characteristics of Life Cycle Archetypes during First Three years after release *3Yclust*

This section presents the game characteristics observed in clusters of *3Yclust*.

***3YfastDec-slowDec Cluster:*** Games in *3YfastDec-slowDec Cluster* appear to have Action, Adventure and Indie tags in common. Also, most games(66%) are released after 2014. Similarly, over 76% of games are released after 2014 in the overall game set. The most common release months appear to be October and November each corresponding to 11% of games. However, interestingly, the least frequent months were June, July, December and January which coincide with major Steam sale months. The overall game set also shows similar distributions. Most games are non-Free-to-Play games in this cluster and most games (38%) are in \$10-20 price range. In the overall gameset, there were only 30% of games in the \$10-20 price range although it was the most common price range. Most games (92%) in this cluster has received 80-100% of positive reviews. In the overall game set, only 61% of games have received 80-100% of positive reviews.

***3Yincrease Cluster:*** Most games that are in *3Yincrease Cluster* have Simulation, Strategy and War tags. A higher number of games (52%) have been released after 2014. The percentage of games released after 2014 in the gameset is higher

than that of the cluster by 24%. August is the most common release month in this cluster closely followed by May and June with only a 7% difference. But in the gameset, August is the third most common release month. Also, there are games released in all 12 months of the year in this cluster. A higher percentage of games are non-Free-to-Play in this cluster and commonly priced under \$20 or between \$30-40. However, in the gameset, the percentage of games in the \$30-40 price range and under the \$20 range is different. Most games in this cluster have also received 80-100% positive reviews closely followed by 60-80% positive reviews range with only 2% difference. However, this difference is 33% in the gameset. The largest difference between positive review percentage distributions was observed between this cluster and *3Ydecrease Cluster* with a Euclidean distance of 75.

**3Ydecrease Cluster:** Games that were in *3Ydecrease Cluster* commonly have Action, RPG, Open World, Survival and Free to Play tags. It contains games such as *Counter-Strike Nexon: Zombies*, *The Lord of the Rings Online* and *Call of Duty: Modern Warfare 2*. Release year of games seemed to be spread across 2010 - 2016 while 2013 and 2016 have an increase of 11%. However, in the larger game set, release years are spread across a wider range starting from 2004. The release months also appear to be distributed between June to December and February. But in the overall gameset, there are games released in the other months as well. Most games in this cluster are Free-to-Play while most games are non-Free-to-Play in the gameset. The smallest difference between price distributions was observed between this cluster and *3Yincrease Cluster* with a Euclidean distance of 5.97.

**3Y3stageDec Cluster:** Games in this cluster mostly consist of Sports, Football, Basketball, Simulation and Casual tags. Some of the games of this cluster are *Pro Evolution Soccer 2016*, *NBA 2K15*, *Clicker Heroes* and *Football Manager 2016*. Most games (88%) have been released during 2014-2016. The most common release months in this cluster are September and November with 33% games in each. But in the overall gameset the most common release months are October and November. The median of the mean population of games in this cluster is 6718 and it is the

highest compared to the other three clusters. Many games in this cluster are not available for purchase 1 year after release. This is probably due to the yearly release of games such as *Football Manager 2015*.

In this section, the characteristics of games in clusters of first year life cycles and first three year life cycles were explored. In general, most characteristics are similar among the corresponding cluster pairs of  $1Yclust$  and  $3Yclust$ . This is not unexpected as it was identified earlier that most clusters of  $3Yclust$  are an extension of the corresponding clusters of  $1Yclust$  and a high percentage of common games exist between cluster pairs. Moreover, while some characteristics revealed unique features of games in clusters some did not provide any distinguishable differences. For instance Publisher, Developer, Mean Population, Price were not very different between clusters.

### **5.3 Conclusion**

This chapter investigated the product life cycle shapes of games in order to explore the player population fluctuations since game release. The study was conducted using daily player population data since the release date of 683 games.

In this study, two piecewise linear regression algorithms were separately utilized to extract life cycle shapes from games. Then a clustering process was conducted to identify life cycle archetypes. Archetypes were generated for first year life cycle shapes and first three year life cycle shapes separately.

It was identified from the study that there are four life cycle archetypes displayed during the first year and four life cycle archetypes displayed during the first three years after game release. However, it was observed that the archetypes of first year and three years are quite similar and the three year archetypes are a possible extension of the first year archetypes. Furthermore, transitions of some games between archetypes were identified indicating a change of life cycle shape after the first year. The most common first year archetype displayed a decrease of player population dur-

### *5.3. Conclusion*

---

ing the first four months and the population size remaining low afterwards. Most games in the dataset (74.2%) belonged to that first archetype. This implies that the number of players continues to decrease rather than displaying a growth after game release in most games representing that popularity of most games does not last a long time. The next archetype displayed a population increasing life cycle. It was the second most common first year archetypal pattern displayed by 12% of games. The third archetype was a decreasing population pattern indicating a slow decaying life cycle. 9.8% of the games belonged to that archetype. The final archetype had a nearly saturated population throughout the first year which drops fast at the end of the year. This archetypal pattern was displayed by 3.6% of the games. The characteristics of the games displaying each life cycle shape were also explored in the study. Specifically, tags, release year, release month, mean population, publisher, developer, price and positive review percentage were analysed.

The outcomes of this study provide game developers and any game-related party insights regarding diverse life cycle shapes that games display. This is useful to understand the longevity of video games. Based on the above mentioned findings, the population of the majority of games keeps on decreasing rapidly and in some games decreasing more slowly, soon after release. However, the population keeps on growing since game release in some other games. This implies that some games can be instantly popular after release, but do not survive in the long run. However, the player population of some games is more sustained. This study provided insights regarding the characteristics of games that display each such life cycle shape. Based on that game developers can determine what games are more likely to display each life cycle shape. For instance, games displaying a growing population pattern during the first year are mostly released during November, December, February and March. December is the popular Steam winter sales month. But games displaying a fast decrease of population after release are mostly released in August, October and November and least released in June, July, December and January months. The least released months are the months in which Steam winter and summer sale events

are held. This implies that games that lose their population fast are released mostly during non-sale periods in Steam. Hence, game developers can consider these insights when selecting a month to release their game considering whether they prefer short term popularity or long term sustainability. However, it should be noted that the game characteristics analysed in this study, such as release month, tags, price alone would not determine the longevity of the game. The features of the game, the storyline and various other factors could play a role in the longevity of the game. It is beyond the scope of this study to analyse that but would be an interesting future work if relevant data becomes available. In addition, it was identified from the study that the life cycle shape in which player population stays high and saturated during the first year and drops fast at the year-end is mostly displayed by sports games. Moreover, the mean population of games was high in this archetype compared to the others. Based on this pattern, game developers interested in sports games can consider doing annual game releases to maintain high popularity for the game. Especially, releasing a new version of the sports game during a major annual sports event such as US National Football League would aid in maintaining a high and steady population during the year. In addition, the study revealed that the life cycle shapes during the first year and first three years are quite similar in shape indicating that the three years life cycle shape is an extension of the first year life cycle shape. Hence, game developers need to understand that the life cycle shape of a game is determined during the first year after release and the same shape will be continued afterwards. Thus, the first year of the game is critical and it is important for them to take actions to grow the population of the game during the first year. Also, three of the life cycle shapes indicate that the player population of games starts decreasing during the first year, either soon after release or at the year end. It further emphasizes that game developers have to focus on obtaining more players during the first year alone to increase the longevity of the game, as in a majority of games population decreases after release. In addition to these specific implications, this study is also helpful for game developers to obtain a general idea regarding the

### *5.3. Conclusion*

---

life cycle shapes games display based on the player population changes.

The study conducted in this chapter was focused on addressing the research question “How does player population of games fluctuate during the first year and first three years after game release displaying life cycle shape approximations?”. This was addressed by identifying that there are four life cycle shapes games display during the first year and first three years after release. Also, the characteristics of games displaying each archetypal life cycle were analysed. One limitation of the study is that only static features such as tags, release months of games displaying each life cycle shape were analysed. However, if data regarding the strategies games used to attract players, such as pre-release marketing campaigns, other promotional events after game release was available, more insights regarding how player population change during the life stages of the life cycle could have been generated. Moreover, since this study was conducted to investigate the life cycle shapes up to the first three years after release, life cycle shapes that games might display beyond this period was not investigated. However, in one of the life cycle shapes, the player population kept on increasing during the three years indicating that some games have a lifetime beyond three years. Hence, it is important to conduct future work to investigate life cycle shapes beyond the three year period.

In conclusion, it was identified from the study that games display four different life cycle shapes during the first year and first three years after release. Various insights were generated related to these patterns.

The next chapter will present a study focused on forecasting player population of games during sale event periods. It will be considering the player population fluctuations in the presence of sale events. The cluster information identified from this chapter will be used in generating the prediction model.

# **Chapter 6**

## **Forecasting Player Population of Games during Sale Events**

The previous chapter presented a study identifying life cycle archetypes of games that represent long term player population fluctuations since game release. This chapter considers the player population fluctuations of games in the presence of sale events. Hence, this chapter presents a study to introduce a forecasting model to predict the player population of games during sale events. Moreover, the life cycle archetypes and corresponding game clusters generated in the previous chapter are used in developing the prediction model.

Sale events in which the games are sold at a discounted price are very popular among game players. As presented in Chapter 2, there are various types of sale events on Steam such as daily deals, weekend deals, midweek madness, seasonal sales, launch discount, custom discounts, and weeklong deals. Sale events are commonly conducted to attract new players to increase the longevity of games. Due to the popularity of sale events the ability to forecast the maximum player population that can be expected during a sale event would aid game developers/publishers to better plan and schedule the sale events. Furthermore, it would also provide an indication of the number of players that would actually play the game rather than just purchasing the game. Although, players purchase games during sale events

---

not everyone would play the game right after or even at a later time [139]. Hence, forecasting sales/number of purchases during sale events would not provide a complete estimation regarding the number of players that would play the game soon after purchasing. However, predicting the maximum player population during sale events can provide an indication regarding the number of players who have actually played the game during the sale event which would include the existing players and the new players. In this study, three approaches are explored to generate a prediction model that can accurately predict the maximum player population during a sale event of a game. The three approaches are focused on generating a single prediction model for all games, generating a single prediction model per cluster of games that display similar life cycle shapes (The four clusters of games that display similar life cycle shapes during the first year after game release identified in Chapter 5) and generating prediction models per game. However, before generating such sale event specific population prediction models, general population prediction models that use the past population to predict the maximum population during non-sale periods are used to determine how accurately those models can predict the maximum player population during sale events. Then three sale event specific population prediction models are introduced to investigate how accurately the maximum population during sale events can be predicted by those models compared to the general population prediction models. The sale event specific population prediction models use sale event information as well rather than generating predictions solely based on the past population as in the general population prediction models. One introduced sale event population prediction model is a multi-layer perceptron model (*sAllModel*) that uses past sale event-related information of all games to predict the maximum population of a future sale event of a game. The key advantage of the proposed model is that it can be used by games with a short history of sale events as predictions are generated not only based on the game's history but also based on the sale event history of all games. In the other introduced prediction model (*sClustModel*) multiple MLP models per cluster of games, identified in Chapter 5, is

### *6.1. Methodology*

---

generated considering the similarity of games with respect to their life cycle shapes. A time series based population prediction model that uses both past population and price history of the game to predict population during a sale event of the game is also generated (*NARXModel*). It is based on a classic time series forecasting approach, namely, Nonlinear Autoregressive network with exogenous inputs.

Sale events of 293 Steam games were used to develop and evaluate the prediction models. Results indicate that in the single model for all games and the single model per cluster approaches, more accurate predictions on the maximum population during sale events can be generated using the sale event specific models compared to using the general population prediction models that are trained to predict population during non-sale periods. However, for the single model per game approaches, more accurate predictions on the maximum population during sale events can be generated by general population prediction models that only use the past population history of the game rather than using sale event specific models that use both the population and price history. Furthermore, it was identified that the most accurate predictions about the population during sale events can be generated by using the sale event history of all games utilizing a single *sAllModel* compared to only using sale event history of games that share the same life cycle shape utilizing *sClustModel*.

The rest of the chapter is structured as follows. First, the methodology of the study is presented including data collection, data pre-processing, and sale event and non-sale period data extraction. Next, the general population prediction models are presented and evaluated. Then, the sale event specific population prediction models are introduced and evaluated. Thereafter, the results and analysis is provided. Finally, the chapter conclusion is presented.

## **6.1 Methodology**

This study is focused on predicting the maximum player population during sale events of games. The methodology of the study consists of data collection and pre-processing, prediction model generation, and evaluation procedures. These are

## *6.1. Methodology*

---

presented in this section and the following sections.

The methodology consists of three approaches of prediction model generation. Prediction models are generated in a single model for all games, a single model for each cluster, and a single model per game approach. Furthermore, these three approaches are used to generate prediction models that predict maximum player population during sale events solely based on the past population data and based on both past population data and sale event related data. The mentioned details regarding the methodology are graphically depicted in Figure 6.1.

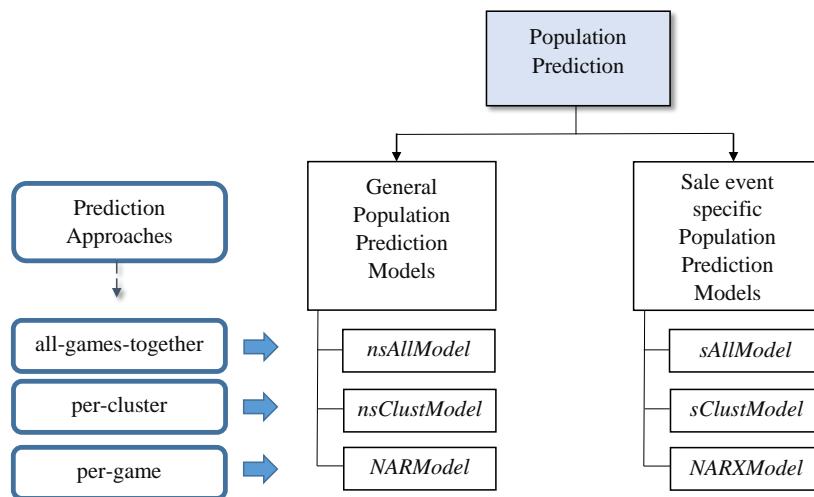


Figure 6.1: Methodology Overview

In this chapter, prediction models that solely use the past population of games to predict the future population is referred to as general population prediction models. The prediction models that use both sale event related information and the past population to predict population during sale events are referred to as sale event specific population prediction models.

This study is focused on forecasting the maximum player population during sale events of a game. A sale event of a game during which the game is sold for a discounted price generally lasts a single or multiple days, except for flash sales that only last a few hours. This study considers the sale events that last at least a single day and attempts to predict the maximum daily population that can be observed on any day during the period of the sale event of a game.

## *6.1. Methodology*

---

A general approach to predict population during sale events is to use a general population prediction model that uses past population to predict future population. Hence, initially, this study investigates how accurately general population prediction models can predict the maximum population during sale events. For this purpose, three general population prediction models (*nsAllModel*, *nsClustModel*, *NARModel*) are generated and their accuracy in predicting the maximum population during non-sale periods and sale events are recorded.

Population forecasting is commonly performed by generating a forecasting model per game. However, it is resource-intensive and less practical to create a forecasting model per game when there are many games. Also, to accurately model the player population fluctuations of a game for forecasting purposes, past player population data alone would not be sufficient. Player population time series of games are volatile and fluctuations occur not only due to sale events but also due to various external factors such as game updates, world events, etc. Thus, all such information about each game has to be collected and incorporated if a single forecasting model per game is generated. Moreover, games that have a short history of past population, would face the cold start problem where data is insufficient to generate predictions. Hence, it can be understood that an approach of generating a forecasting model per game is quite challenging. However, a Nonlinear Autoregressive model (*NARModel*), which is a prominent time series forecasting approach, is utilized in this study to predict population in a per-game approach using past population. It is used to predict population during non-sale periods.

Some of the mentioned challenges associated with generating a single forecasting model per game can be solved if a single forecasting model for all games focused on sale events can be generated. Firstly, creating a single model for all games would be less time consuming compared to creating a single model per game. This is because in a single model per game approach, a model has to be created and model parameters need to be fine-tuned for each game. It could take a long time when there are many games. However, for a single model for all games approach model

## *6.1. Methodology*

---

generation needs to be done once for all games. Secondly, a single model for all games approach would aid in addressing the cold start problem as it generates predictions based on other games as well. Hence, an Artificial Neural Network (ANN) model, specifically, a Multi-Layer Perceptron (MLP) model that uses the population of past seven days to predict the maximum population during non-sale periods is generated (*nsAllModel*). The model uses population data related to non-sale periods of all games.

Moreover, in order to reduce any shortfalls in the single model per game approach due to heterogeneity of population series of games, the product life cycle clusters of games identified in Chapter 5 can be used in the forecasting approach. Specifically, one *nsAllModel* per cluster of games sharing the same product life cycle archetype is generated to predict population during non-sale periods. This approach is referred to as the *nsClustModel* in the chapter. However, this single model per cluster approach would not be able to fully aid in addressing the cold start problem as the population data of the game during the first year is required for life cycle shape extraction. But the single model per game approach which does not use the cluster information would still be able to assist in addressing the cold start problem. Hence, both approaches are used in the study.

The three general population prediction models, generated in all-games-together approach (*nsAllModel*), per-cluster approach (*nsClustModel*) and per-game approach (*NARMModel*) are evaluated to determine how accurately those can predict the maximum player population during non-sale periods. Then the models are also evaluated to determine how accurately those models can predict the maximum population during sale events. Afterwards sale event specific population prediction models are introduced in the study. The sale event specific models use sale event information along with the population data prior to the event. It is expected that these models can predict population during sale events more accurately compared to the general population prediction models. Three sale event specific population prediction models are investigated in the study. A prominent ANN-based time series forecasting

model, namely, Nonlinear autoregressive exogenous model (NARX) is used as the per-game approach (*NARXModel*). It uses the population and price history of the game. An MLP model, which uses sale event information and past population as input features, is introduced for the all-games-together approach (*sAllModel*). Since such an approach use sale event information of all games, it can be especially useful for games that do not have a sufficient history of past sale events and suffer from the cold start problem. Moreover, a cluster-based MLP model that creates a single *sAllModel* per cluster of games displaying the same life cycle shape is created for the per-cluster approach (*sClustModel*).

The forecasting models and evaluation process are explained in detail later in the chapter. The following subsection provides the details of data collection.

#### **6.1.1 Data Collection and Preprocessing**

A subset of games in *Gameset2* introduced in Chapter 3 was used. It is the same dataset of games used in the previous chapter (Chapter 5). It contains player population data series in daily frequency. However, some games were removed based on particular filtering criteria. Games in which player population data is not available since release date and games in which release date is not available are removed. Moreover, games that did not have at least one year of post-release population data were removed along with games released in Early access mode. These constraints were used in the previous chapter to select games for game life cycle cluster generation. Since that cluster information is used in the models proposed in this chapter, the same dataset was used. However, Free to Play games are also removed as sale events are not applicable for those games.

The price history of those games was also obtained. It should be noted that Steam games are not generally put on for sale all around the world at the same time or the same discount level. Some sale events are restricted to selected countries and even the price is different across countries. Hence, the US price history of games are used in this study as the majority of the Steam users are located in the USA [121].

## *6.1. Methodology*

---

The price history obtained in USD currency contains the daily price information which includes initial price, final price and discount. Moreover, it was identified that the price history of some games is not available since the release date of the games, but, from a later date. But having the price details since the release date is important to extract some input features for the models. Hence, games whose price history was incomplete were not used. Thus the final dataset contained 293 games.

As the initial step of preprocessing, missing data in player population series are imputed by linear interpolation. Furthermore, each player population series is normalized to the 0-1 range. Since player population size of games are quite varied normalization is quite useful as the *sAllModel*, *sClustModel*, *nsAllModel* and *nsClust-Model* are generated for multiple games. Also, it would be helpful in understanding the RMSE values in the evaluation process.

### **6.1.2 Sale events and non-sale periods Extraction**

The daily player population and price information are available as time series in the dataset. Hence, sale events and non-sale periods have to be extracted from those series to be used in the MLP based prediction models used in this study. This section presents the sale events and non-sale periods extraction process.

This study is focused on predicting the maximum player population that can be expected on any day during a given sale event of a game. Hence, the sale events of each game which offer discounts have to be identified to create the dataset for the prediction models. Hence, the discount values of the daily price information series of each game are examined. While iteratively going through the discount series, whenever a non-zero discount value is encountered it is identified as the start of a sale event of the game. Then the consecutive discount values of the series are investigated to determine the end of that sale event. If the consecutive discount values of the series are similar to the previously observed discount value, those consecutive days are considered to be in the same sale event. Hence, the consecutive discount values are iteratively compared until a zero or different discount value is

### *6.1. Methodology*

---

observed to determine the end date of the sale event. This process identifies all sale events of games. The start date, end date, initial price, final price and discount of each identified sale event is recorded.

Several criteria are used to select the final set of sale events. First, each event should last at least one day. Hence, the models that predict population during sale events are not focused on flash sales that last for only a few hours. Moreover, some criteria are used which are related to the procedure of feature selection for the proposed *sAllModel*. Sale events conducted within the first seven days after game release are not used as the population of the seven days prior to the event are used to generate predictions. Moreover, the first sale event of a game is not used as it is difficult to assign a value to the feature related to the number of days between that event and the previous event. The final dataset contained 4956 sale events.

The extracted sale event set was analysed to investigate the population increase that can be observed during sale events. Figure 6.2 depicts the box plot of the relative increase percentage of the mean population during sale events. The population increase percentage is calculated as per Equation 6.1 where *MeanPopPreSale* is the mean population of the seven days prior to the sale event.

$$PopIncreasePercent = \frac{(MeanPopDuringSale - MeanPopPreSale)}{MeanPopPreSale} * 100 \quad (6.1)$$

As per Figure 6.2, the median of the population increase percentage that can be observed during a sale event is 41%. The minimum increase percentage is -98% indicating the existence of sale events that do not observe an increase in population. Specifically, 11.6% of the events in the sale event set displays a decrease in population during the sale event. However, it can be understood that the mean population of games has changed during sale events ranging from a minimum of -98% up to a maximum of 3190% of population increase.

Non-sale periods were also extracted from the dataset. For this purpose, periods of seven days that have zero discount were selected from the daily price information series of each game. Seven-day non-sale periods were chosen as it was identified that

### *6.1. Methodology*

---

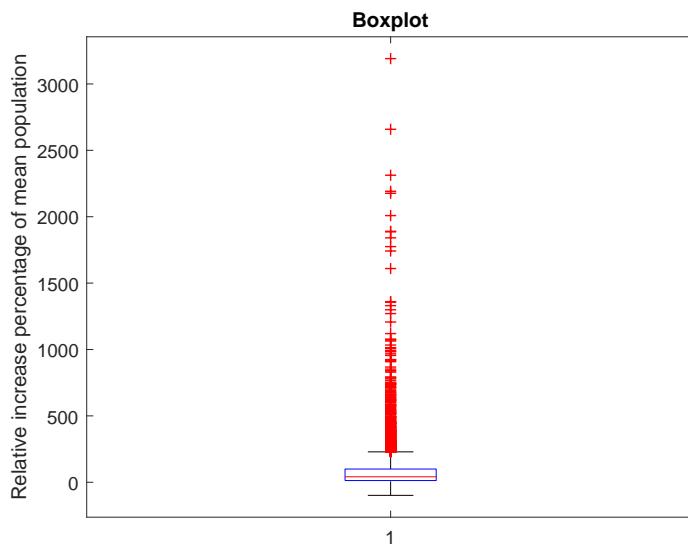


Figure 6.2: Boxplot of the relative increase percentage of mean population during sale events

both the average and the median of the duration of a sale event is seven days in the sale event set. Since comparisons are conducted between the predictions of the maximum population during sale event periods and during non-sale periods, later in the study, non-sale periods that have durations comparative to sale event periods were selected. Moreover, when selecting non-sale periods, only the non-sale periods that have not had a sale event during the seven days prior to the selected period is chosen. Such a selection criterion is used to select non-sale periods that have less influence from any sale event period of the game. Furthermore, the population during the seven days prior to the non-sale event period is also used in the prediction models. The selected set of non-sale periods contained 16400 instances.

Moreover, for each of the identified sale events the maximum player population that can be observed on any given day during the period of the event is also recorded. The forecasting models would be forecasting these extracted maximum population values during sale events. The maximum population of a sale event is recorded by extracting the population values corresponding to the period from the start date to the end date of the sale event from the player population series of the game and identifying the maximum value among the extracted values. The maximum player population for each non-sale event period is also recorded in a similar way.

## 6.2 General Population Prediction Models for non-sale periods

Prior to investigating how accurately the maximum player population during sale events can be predicted, it is important to investigate how accurately the maximum population during non-sale periods can be predicted. Hence, this section presents three prediction models designed to predict the maximum population during a given period based on the population prior to that period. Also an evaluation of those models is provided.

### 6.2.1 Prediction Models

This section presents three prediction models designed to predict the maximum population during a non-sale period. The three models represent three different prediction approaches, namely, all-games-together approach, per-cluster approach and per-game approach.

#### 6.2.1.1 Multi Layer Perceptron Model for non-sale periods (*nsAllModel*)

An Artificial Neural Network (ANN) is a machine learning model that mimics a biological neural network. A Multi Layer Perceptron (MLP) is an ANN that constitutes of perceptrons and contains one or many layers of hidden nodes in between the input and output layers. MLP is known to be a feed-forward neural network as data flows forward from the input layer to the output layer [140].

A Multi Layer Perceptron model that uses the past population to predict the maximum population during a non-sale period is used in this study. This model uses the daily population of the seven days prior to the considered period as input features. Population of seven days are used considering weekly seasonality in population fluctuations. Hence, the input layer of *nsAllModel* contains seven input nodes. Also, it contains a single layer of hidden nodes. The number of nodes in the hidden layer is chosen to be six as it has been identified that in MLP models

with a single hidden layer, the optimal choice for the number of hidden nodes is  $n - 1$  hidden nodes, where  $n$  is the number of inputs [141]. The hidden layer uses a hyperbolic tangent sigmoid activation function, known as *tansig*. The activation function used in the output layer is a linear transfer function, known as *purelin*. A linear activation function is used in the output layer since the goal of the model is to predict the maximum population which is a continuous variable. Moreover, this model is trained using the non-sale period population dataset extracted from all the games. Hence, the model is capable of predicting the maximum population during a non-sale period of a game based on the knowledge of the maximum population during non-sale periods of all games.

#### **6.2.1.2 Life Cycle Cluster based Multi Layer Perceptron Model for non-sale periods (*nsClustModel*)**

The *nsAllModel* presented in the previous section provides a single prediction model for all games to predict population during non-sale periods. However, rather than having a single *nsAllModel*, having several instances of the *nsAllModel* for similar sets of games could further enhance the prediction accuracy. Specifically, the life cycle shape based similarity of games can be used as it represents long term player population fluctuation patterns of games. Hence, the life cycle archetypes of games identified in Chapter 5 are used to generate the *nsClustModel*. Specifically, for each cluster of games displaying one of the four first year life cycle archetypes, a *nsAllModel* is generated.

#### **6.2.1.3 Non-linear Autoregressive Model (*NARmodel*)**

The *NARmodel* is used to generate predictions in a per-game approach where a single *NARmodel* per game is generated to predict the maximum population. Non-linear autoregressive model (NAR) is the non-linear variant of the prominent linear autoregressive time series forecasting model. The NAR model uses the past values of a given series to forecast the future values of the series. The NAR model uses

recurrent neural networks (RNN) in which the output produced by the model is used as an input to the model. The architecture of the *NARmodel* is depicted in Figure 6.3.

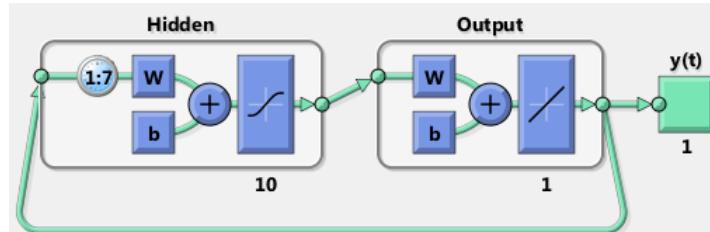


Figure 6.3: Architecture of Non-linear autoregressive model (*NARmodel*); The population of the past seven days is used as the 7 input variables. The output variable is the predicted population.

The *NARmodel* has a single hidden layer with 10 hidden nodes. It uses *tansig* and *purelin* as the activation functions. The population of the past 7 days is used for forecasting future population. Training of the model is conducted in open loop mode using actual population values in the series. Although a single step ahead forecast of population can be generated solely using the past values, multi-step ahead forecasting generation requires the population output value generated by the model to be fed back into the model through the closed-loop. However, during the training process of the *NARmodel*, the generated output is not fed back to the network to forecast the next value of the series. Instead, the expected output which is readily available is used as it is more accurate. Hence, the architecture of NAR takes the form of a feedforward network during training and learning is done through backpropagation. For each non-sale period in the dataset the population values during the period can be forecast by training a *NARmodel* using population data of the game prior to the non-sale period. Once the population of each day during the non-sale period is predicted, the maximum value of those predicted values are recorded as the predicted maximum population during the non-sale period.

### 6.2.2 Evaluation Procedure

This section presents the evaluation procedure conducted to evaluate the performance of the introduced *nsAllModel* and *nsClustModel*.

The simplest method of evaluating supervised learning models is the holdout method. In that approach, the non-sale periods dataset is divided into three sets namely, training, validation and test sets. The training set is used to train the network and the validation set is used to validate. Finally, the test set is used to obtain a final estimation of the model's performance. Specifically, the RMSE given in Equation 6.2, is used to measure the performance of the model in predicting the maximum population during all non-sale periods in the test set. In Equation 6.2,  $N$  depicts the number of instances used.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N E_i} \quad (6.2)$$

However, the evaluation conducted by the holdout method would be dependent on the way the dataset is divided which is identified as a drawback of the method [117].

Cross-validation is a prominent evaluation approach that makes better use of the holdout method addressing its limitation. In the cross-validation approach, the dataset is divided into  $k$  subsets randomly and the holdout method is performed  $k$  times using one subset as the test set and the rest as training and validation sets each time. However, in forecasting approaches where past data is used to predict future values, cross-validation is performed slightly differently preserving the time dependent information [106]. Since the goal of the *nsAllModel* and *nsClustModel* is to predict the maximum population during a non-sale period of a game based on the past population of games, a cross-validation approach inspired by the time series cross-validation technique is used.

Figure 6.4 depicts an overview of the cross-validation approach used to evaluate the *nsAllModel* and *nsClustModel*. It conducts the holdout method  $k$  times iteratively increasing the size of the training and validation sets while keeping the test

set size fixed. In each iteration the non-sale period dataset is divided into training, validation and test sets in a way that the dates of the non-sale periods in the training set appear earliest in the calendar followed by non-sale periods in the validation and test sets consecutively. Hence, this evaluation approach simulates forecasting of the future population during non-sale periods based on the population of past non-sale periods. Moreover, unlike the common cross-validation approach, this approach uses the full non-sale period dataset only in the last iteration and previous iterations use only a subset of the non-sale periods data based on time dependencies. The cross-validation was performed for 3 iterations using three different test sets of size 2460 which represents 15% of the non-sale periods in the dataset. The number 15% was chosen since 70-15-15 is a common dataset splitting percentage used by the holdout method. Moreover, only 3 iterations were conducted so that there is sufficient past non-sale period information for training in the first iteration of cross-validation. In each iteration after the test set is chosen, the rest of the non-sale period instances which are dated prior to all non-sale period instances in the test set is split into training and validation sets by splitting the set of non-sale period instances into 75% and 25% respectively. Table 6.1 depicts the boundary dates chosen to split the dataset into training, validation and test sets during cross-validation preserving the time order of non-sale period instances and the mentioned percentages of subsets.

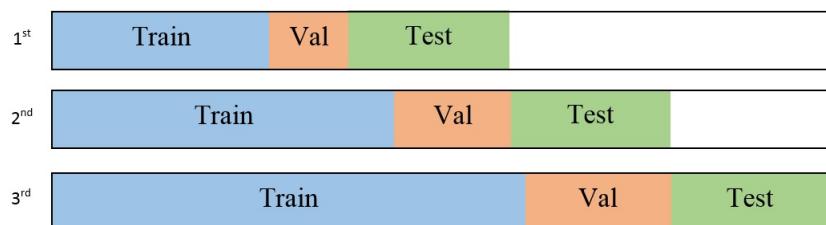


Figure 6.4: Iterative cross-validation: Non-sale period instances dataset is split into train, validation and test sets iteratively considering temporal placement of non-sale periods for forecasting and split percentages;test set size is constant and test sets in each iteration never overlaps

In order to calculate the overall performance of a model, the squared error in predicting the maximum population during each non-sale period in the test sets of all three iterations are recorded. Then the RMSE of all the test sets is calculated.

Iteration	Boundary Dates of Sets (Non-sale Periods)		
	Train	Validation	Test
1	$e_d < 30-01-2018$	$s_d \geq 30-01-2018 \text{ &}$ $e_d < 17-04-2018$	$s_d \geq 17-04-2018 \text{ &}$ $e_d \leq 10-11-2018$
2	$e_d < 16-06-2018$	$s_d \geq 16-06-2018 \text{ &}$ $e_d < 10-11-2018$	$s_d \geq 10-11-2018 \text{ &}$ $e_d \leq 13-04-2019$
3	$e_d < 27-10-2018$	$s_d \geq 27-10-2018 \text{ &}$ $e_d < 13-04-2019$	$s_d \geq 13-04-2019 \text{ &}$ $e_d \leq 09-09-2019$

Table 6.1: Boundary dates that provided the required split percentage of training, validation and test sets in each iteration of the cross-validation process;  $s_d$  represents the start date of a selected non-sale period and  $e_d$  represents the end date of a selected non-sale period

The RMSE of *nsAllModel* and *nsClustModel* can be used to compare the prediction error of the models. Evaluation of the *NARmodel* is conducted using the same three non-sale period test sets and the RMSE values are recorded.

### 6.2.3 Outcomes

Three general population prediction models, namely, *nsAllModel*, *nsClustModel*, *NARModel* were used in this study to predict the maximum player population during non-sale periods of games. A general evaluation of the models was conducted by following the evaluation procedure presented in the previous section. The results are depicted in Table 6.2. The results of that evaluation process indicate how accurately the models that are trained to predict maximum population during non-sale periods can predict the maximum population during non-sale periods. This scenario is named as *Non-Sale* in Table 6.2. In addition to that, two other evaluations were conducted to explore the performance of the prediction models in two different scenarios.

One scenario was focused on identifying how accurately the maximum population during sale events can be predicted by these models if they were trained using population during sale event periods instead of non-sale periods. This scenario is named as *Sale* in Table 6.2. For this purpose, the *nsAllModel* and the *nsClustModel* were trained using the extracted sale events dataset by using the population of the

seven days prior to the sale event as input features to predict the maximum population during the sale event. In order to evaluate the performance, the evaluation procedure presented in the previous section was conducted. The boundary dates used in the cross-validation process to select test sets from the sale event dataset are presented in Table 6.4.

The second scenario was focused on identifying how accurately the maximum population during sale events can be predicted by the *nsAllModel* and *nsClustModel* trained to predict population during non-sale periods. This scenario is named as *Sale using non-sale*. For this purpose, the models were first trained using the non-sale periods dataset. Then the sale event dataset was used as the test set. The results of the main evaluation and the other two variants are presented in Table 6.2 and 6.3.

Prediction Model	RMSE Results		
	Non-Sale	Sale	Sale using non-sale
<i>nsAllModel</i>	0.0301	0.1478	0.1984
<i>nsClustModel</i>	0.0288	0.1501	0.1986

Table 6.2: RMSE results from *nsAllModel* and *nsClustModel* for the three evaluation approaches; Non-Sale: Model trained on Non-Sale period data and tested on the same, Sale: Model trained on Sale event period data and tested on the same, Sale using non-sale: Model trained on Non-Sale period data and tested on Sale event period data

Prediction Model	RMSE Results	
	Non-Sale	Sale
<i>NARModel</i>	0.0329	0.1685

Table 6.3: RMSE results from *NARModel*; Non-Sale: RMSE for predicting maximum population during non-sale periods, Sale: RMSE for predicting maximum population during Sale periods

However, the *NARModel* cannot be used to evaluate the second scenario. Since the *NARModel* is a time series based forecasting approach, all the population data points of the series prior to a specific point are used to train the model iteratively by using the population of the past seven days as input in each iteration. Due to this inherent training process, the model can not be trained separately for non-sale

periods or sale event periods. However, the model can be evaluated to determine how accurately it can predict population during non-sale periods and sale-periods using past population data of the game. Hence, the results of the *NARModel* are presented separately in Table 6.3.

Iteration	Boundary Dates of Sets (Sale Event Periods)		
	Train	Validation	Test
1	$e_d < 19-12-2017$	$s_d \geq 19-12-2017 \text{ &}$ $e_d < 17-04-2018$	$s_d \geq 17-04-2018 \text{ &}$ $e_d \leq 10-11-2018$
2	$e_d < 17-04-2018$	$s_d \geq 17-04-2018 \text{ &}$ $e_d < 10-11-2018$	$s_d \geq 10-11-2018 \text{ &}$ $e_d \leq 13-02-2019$
3	$e_d < 27-10-2018$	$s_d \geq 27-10-2018 \text{ &}$ $e_d < 13-02-2019$	$s_d \geq 13-02-2019 \text{ &}$ $e_d \leq 09-09-2019$

Table 6.4: Boundary dates that provided the required split percentage of training, validation and test sets in each iteration of the cross-validation process;  $s_d$  represents start date of a sale event and  $e_d$  represents end date of a sale event

As per the results in Table 6.2, it can be seen that the results of the *nsAllModel* and the *nsClustModel* has nearly similar RMSE values in all three scenarios; *Non-Sale*, *Sale* and *Sale using non-sale*. Two sample t-tests validated that there is no statistically significant difference of the mean errors in all three scenarios between the two prediction models. This indicates that using life cycle shape based clusters in the prediction models have not introduced any improvement over the *nsAllModel* which uses a single model for all games. Hence, it can be understood that when predicting maximum population during non-sale periods or sale event periods using the population of past seven days, life cycle shape information of games do not provide any advantage in improving the prediction accuracy in prediction approaches that create a single model for multiple games.

Moreover, as per the results in Table 6.2, the lowest prediction error can be observed in the *Non-sale* scenario where the models are trained to predict the maximum population during non-sale periods and tested on the same. The Figure 6.5 depicts the regression plot for *nsAllModel* displaying the correlation between the actual and predicted values for the test set 3. It indicates that there is a high correlation between the predictions generated by the model and the actual values

regarding the population during non-sale periods as it has a R-value of 0.97. However, as per Table 6.2, the prediction error is comparatively higher when the models are trained to predict the maximum population during sale events and tested on the same, which is the *Sale* scenario. A right-tailed two-sample t-test also revealed that the mean error in the *Sale* scenario was statistically significantly higher than the mean error in the *Non-sale* scenario in the *nsAllModel* with a p-value of 6.0901e-97 and in the *nsClustModel* with a p-value of 8.3178e-109. This indicates the difficulty in predicting population during sale events compared to non-sale periods. Even though the models were trained specifically for sale periods, the models have displayed higher prediction error compared to the models trained for non-sale periods. Hence it can be understood that it is challenging to generate accurate predictions regarding the maximum population during sale events by using the past population alone.

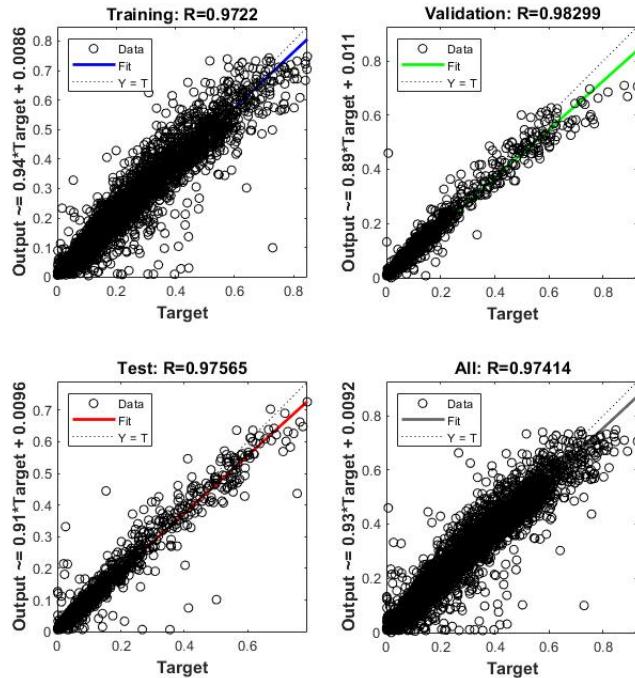


Figure 6.5: Regression plot of actual and predicted values resulting from *nsAllModel* when the model is trained to predict population during non-sale periods and tested on the same. (The *Non-sale* scenario) The figure depicts the outcomes for test set 3 of the non-sale period dataset. It can be seen that the model correlation between the predicted and actual values are high as represented by the R value of 0.97

As per Table 6.3, the *NARModel* also displays a higher prediction error in predicting the maximum population during sale events compared to predicting the maximum population during non-sale periods. A right-tailed two-sample t-test validated the statistical significance of the error difference with a p-value of 5.8993e-120.

Furthermore, in Table 6.2 the highest RMSE can be observed in the scenario of *Sale using non-sale* where the models trained to predict population during non-sale periods are used to generate predictions during sale periods. Figure 6.6 depicts the regression plot of *nsAllModel* displaying the correlation of predicted and actual values where R is low at 0.69. A right-tailed two-sample t-test also indicated that the mean error in the *Sale using non-sale* scenario is statistically significantly higher than the mean error in the *Non-sale* scenario in the *nsAllModel* with a p-value of 2.4688e-181 and in the *nsClustModel* with a p-value of 1.2573e-183. It can be seen that the RMSE increase in the *Sale using non-sale* scenario is 6.59 times the RMSE in the *Non-sale* scenario for the *nsAllModel* and 6.8 times for the *nsClustModel*. Hence, it can be understood that the error has increased by approximately 6-fold when the general population prediction models are applied to predict the maximum population during sale periods. This indicates that general population prediction models trained to predict population during non-sale periods cannot be used to predict the population during sale periods with the same accuracy. Hence, it is important to explore if sale event specific models that not only use past population but also sale event information can predict population during sale events more accurately. Hence, in the next section sale event specific prediction models are introduced. The error values displayed by the models in the *Sale using non-sale* scenario is used as the baseline to compare the performance of the sale event specific prediction models.

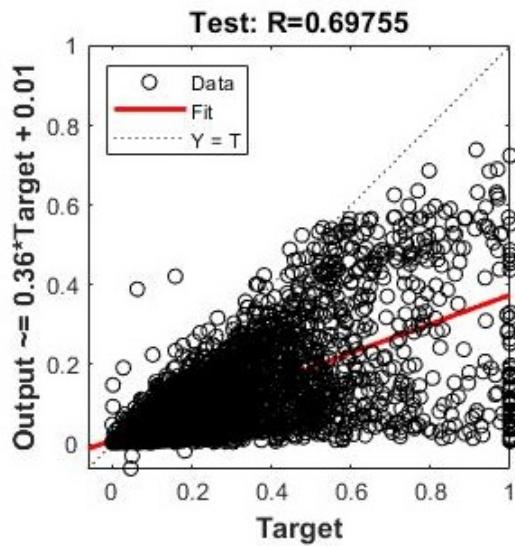


Figure 6.6: Regression plot of actual and predicted values resulting from *nsAllModel* when the model is trained to predict population during non-sale periods and tested on predicting population during sale events. (The *Sale using non-sale* scenario) The figure depicts the outcomes for the sale event dataset used as the test set. It can be seen that the model has mostly under-predicted. It can be expected as the model was trained to predict population during non-sale periods where high population increases are not common compared to sale event periods.

## 6.3 Sale event specific Population Prediction Models for sale event periods

In order to accurately predict the maximum player population during sale events, three sale event specific prediction models are introduced in the study. The models not only use past population but also use sale event information in generating predictions. This section presents the models and presents the evaluation approach.

### 6.3.1 Prediction Models

This section presents the three sale event specific population prediction models that are based on the all-games-together, per-cluster and per-game approaches.

#### 6.3.1.1 Multi Layer Perceptron Model for sale events (*sAllModel*)

In this section the Multi Layer Perceptron Model (*sAllModel*) used to predict the maximum player population during a sale event is introduced. The *sAllModel* is

a sale-event specific prediction model that uses sale event related features as input to predict the maximum population during a sale event. Hence, initially, a feature selection procedure is conducted to select the features for the model.

### Feature Selection

Feature selection is an important step in constructing a predictive model. This section provides details of the feature selection approach of the *sAllModel*.

In order to develop the *sAllModel* to forecast player population during sale events, features that can be used as input variables for the model have to be determined. Hence, features that can be extracted from player population series and price series of each game related to sale events were identified. Table 6.5 depicts the features identified related to each sale event. Among these initially identified 13 features, 3 of the features are related to the discount of the sale event. Those 3 features are the Initial Price, Final Price and Discount. However, Initial Price was not selected for the final feature set of the model as it can be derived from the other two features.

Feature Name	Description
Start Date	Start date of event
End Date	End date of event
Initial Price	Initial price prior to discount
Final Price	Final price after applying discount
Discount	Discount
Duration	Number of days of the event
Days since last Event	Number of days since the last sale event of the game
Days since Release	Number of days since the release of game
Pre event count	Number of sale events of the game prior to current event
Final Price Percentage	Percentage of the final price relative to the lowest observed final price under the same initial price
Month	Month(s) of the sale event
Weekend Count	Number of Saturdays and Sundays during the sale event period
Pre Population	Mean player population of the 7 days prior to the sale event

Table 6.5: Initially identified features for the prediction model; After the feature selection process Initial Price and Start Date features are removed

Since some of the features are not purely numerical those had to be converted. Hence, the start date and the end date was converted to a serial date number by recording it as a whole and a fractional number of days from January 0, 0000. Also, since the month is a categorical variable it was encoded as 12 binary variables. The month or months in which a sale event is held is marked as 1 in the corresponding variable out of the 12 that represent the month(s). Month is encoded as 12 binary variables instead of a single integer between 1 to 12 as an integer would assign a magnitude to each month that would be misleading as they are only categories. For instance, although January comes right after December, when December is recorded as 12 and January is recorded as 1 an incorrect distance between those months are introduced as an integer encoding approach is not aware of the cyclic nature of months ordering.

Although some of the features are self-explanatory some require an explanation. When discounts are offered some game players might check whether it is the cheapest offer compared to the previous offers in the history of the game. Hence, the percentage of the final price relative to the lowest observed final price can be calculated as a feature. However, the initial price of a game sometimes changes multiple times throughout the lifetime of a game which is not associated with any discount. Such a price change lasts for a while permanently and discounts are offered relative to that current price. Hence, instead of calculating the percentage of the final price relative to the lowest final price observed during the complete history of the game, the percentage of the final price relative to the lowest final price observed during the period the current initial price existed is calculated. The total number of Saturdays and Sundays observed within the event is recorded as it indicates whether the sale event has any weekends. Moreover, the number of days since the release of a game is used as it indicates the age of the game which would represent how long the game has been in the market and its life cycle stage.

All the mentioned features corresponding to each sale event in the sale event dataset created earlier were extracted from population series and price series of each

game. As mentioned earlier, the first sale event of each game was removed from the dataset since it is not possible to calculate the feature value of *Days since last Event* feature. Also, *Pre Population* feature cannot be extracted if the first event is held within seven days from the release date.

All the mentioned features were chosen based on the domain knowledge of sale events. It is assumed that these features would have an influence on the maximum player population during a sale event of a game, which needs to be predicted. However, a standard feature selection procedure was conducted to determine the final subset of features out of these initially chosen feature set.

Backward elimination is a prominent feature selection technique [142]. It is regarded as a wrapper method as feature subsets are chosen based on the performance of the model that uses the feature subset [143]. The backward elimination approach is given in Algorithm 6.1. This approach initially uses all available features as input to the model and eliminates one feature in each iteration until only a single feature is left. To eliminate a feature in a certain iteration, the model is trained using all features except a single feature which is iteratively set aside and the performance of the model is recorded. The feature chosen to eliminate in a given iteration is the one when eliminated the model trained using all other features provides the best model performance measured by some evaluation criteria. Root Mean Squared Error (RMSE) is used as the evaluation criteria in this study.

Following Algorithm 6.1, the *sAllModel* is trained using feature subsets iteratively and its performance in terms of predicting the maximum player population of a sale event is recorded as the Root Mean Squared Error. To train the *sAllModel* and record its performance for the backward elimination approach, the sale event dataset is first divided into training, validation and test sets in 70%, 15% and 15% portions. In order to preserve the temporal nature of the problem, the dataset was not randomly divided rather based on a date as the boundary preserving the mentioned portion percentages. All events prior to 27.10.2018 were in the training set and all events after 03.02.2019 were in the test set while events in between those

**Algorithm 6.1** Backward Elimination for Feature Selection adapted from [144]

---

Let  $S_1$  be the set containing all  $N$  features of the feature set  $N_f$ ;  
 Train the model with  $S_1$  and record performance as RMSE using a validation data set  
**for**  $N = 1$  upto  $N_f - 1$  **do**  
**for each**  $s \in S_N$  **do**  
     Set  $S = S_N - \{s\}$   
     Train the network with  $S$  feature set  
     Record model performance when  $s$  feature is removed by computing RMSE ( $E_s$ ) using a validation data set  
**end for**  
     Find the minimum  $E_s$  and corresponding feature  $s$   
     Set  $S_{N+1} = S_N - \{s^*\}$  where  $s^*$  is the feature when removed resulted in minimum  $E_s$  in the previous loop  
**end for**

---

dates were in the validation set. The performance of the *sAllModel* in forecasting the maximum population of events in the validation dataset was recorded by calculating the squared error between the actual and predicted value and calculating the mean of those errors, and recording the square root of the mean squared error value, recognized as RMSE. The validation dataset was used in order to avoid over fitting.

Figure 6.7 depicts the backward feature elimination approach outcomes. Note that the *month* is considered as a single feature although it was encoded as 12 binary variables. Hence, when the month feature needs to be eliminated all 12 variables are eliminated together. As per Figure 6.7, the minimum RMSE can be observed when only 1 feature is removed from the model, which is the *Days since last Event*. Moreover, the largest RMSE is observed after the elimination of 12 features, which are all features except the *Pre Population* feature. This indicates the importance of the *Pre Population* in predicting the sale event population. Since *Pre Population* has not been eliminated in any of the iterations, it indicates that the mean population of the seven days prior to the event is an important feature in predicting population during sale events. The RMSE values of the models resulting after elimination of features one by one appears to be different and increasing as per Figure 6.7. However, it is important to investigate if there are any statistically

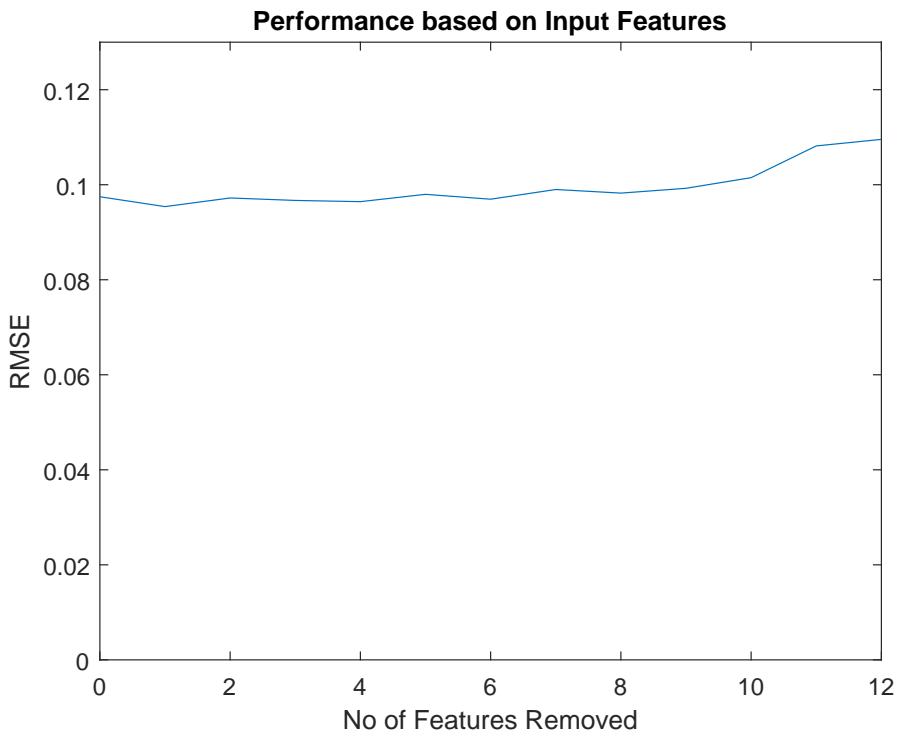


Figure 6.7: Backward feature elimination outcome: Backward elimination is performed as per Algorithm 6.1. Order of eliminated features: Days since last Event, Final Price, month, Start Date, End Date, Pre event count, duration, Initial Price, Final Price Percentage, Weekend Count, discount, Days since Release

significant difference between the sets of squared error values resulted from each model for the used validation set. Conducting a statistical test rather than directly opting for the model with the lowest RMSE would aid in better understanding the importance of the features. Hence, the set of error values resulting from the model with the lowest RMSE which is the model where only 1 feature is removed, is compared against the set of error values resulting from each of the other models by conducting paired t-tests. For this purpose, 11 paired t-tests were conducted. The outcomes indicated that the mean difference between the error values resulted from the model where only a single feature is removed is not statistically significantly different from the error values of the models where 2, 3, 4, 5, 6, 7 and 8 features are removed. Also, there was no statistically significant difference between the error values of the model where only one feature is removed and the error values of the model where none of the features are removed where p-value was 0.2411 at 0.05 significance level. However, it was identified from the paired t-tests that

the differences between the error values resulting from the model where only one feature is removed and the error values of the models where 9, 10, 11 and 12 features are removed are statistically significant. Hence, it can be identified that the error values of any models up to the removal of 8 features are not statistically significantly different from the error values of the model where only a single feature is removed which has the lowest RMSE value as per Figure 6.7. Hence, it can be seen that by choosing any model where only up to 8 features had been removed, a lower RMSE from the model can be obtained in the prediction task. Thus, based on the backward elimination outcomes the model where none of the features are removed is selected which use all available features.

However, two features were removed as those were redundant and derivable according to domain knowledge. The Start date was removed as duration and end date can be used to derive the start date. The Initial price was removed as the final price and discount can be used to derive it. Thus, the final feature set resulting from the feature selection contains all features in Table 6.5 except Start Date and Initial Price.

Although a correlation and redundancy analysis was not conducted for feature selection, the conducted backward feature elimination approach indicates what features should be selected to obtain the best performing model that can be created with the currently available features. The final feature set for the model was selected by analysing the RMSE values resulting from the backward incremental feature elimination depicted in Figure 6.7 and the statistical significance analysis of the error values. As explained in detail earlier, the outcomes of the feature selection process indicated that choosing any model where only up to 8 features had been removed according to the backward feature elimination order, would provide lower RMSE and would not have any statistically significant difference between the RMSE values of the models. Hence, even if more features are removed from the currently selected feature set, better performing models cannot be achieved as the RMSE values resulting from the feature removed models do not have any statistically significant

improvement. Furthermore, it should be noted that predicting player population regardless of during sale events or not is a difficult task as it is difficult to capture all the relevant features. The proposed prediction approach is a relatively primitive approach for predicting population during sale events as only a limited set of features have been used. However, better performing models could be created if more features relevant to the sale events, such as the type of the sale event, if the sale event was displayed on the Steam home page, could be captured. The model provided in this chapter is the best performing model, with respect to prediction accuracy, that can be created with the currently available features.

Feature normalization is conducted prior to using the selected features in the *sAllModel* and *sClustModel*. It is important as the value ranges of the selected features are quite varied. The optimization algorithms used in neural networks might be impacted and might not converge faster if feature values are in various ranges. Hence, all features are normalized to 0 -1 range by min-max normalization prior to training the models.

### ***sAllModel* Introduction**

After selecting the features for the *sAllModel*, the model can be generated. This section presents all the parametric, algorithmic and architectural information of the model.

The architecture of the MLP model used in this study for sale event population prediction (*sAllModel*) contains a single layer of hidden nodes. A single hidden layer is chosen as it has been identified that MLP models with a single hidden layer are powerful enough for supervised learning problems [145]. Moreover, having multiple layers of hidden nodes could sometimes result in over fitting or underperform when the dataset is not of substantial size [146]. Since, the *sAllModel* has 22 input variables, 21 was chosen to be the number of hidden nodes. The output layer of the *sAllModel* consists of only a single node as only a single value, which is the maximum player population of a sale event, is predicted by the model. In the *sAllModel*, *tansig* and *purelin* are used as the activation functions in the hidden layer and the output

layer respectively. Figure 6.8 depicts an overview of the *sAllModel*, representing all the mentioned hyperparameters and function choices. Multilayer perceptron models are trained using the backpropagation algorithm. The *sAllModel* uses Levenberg Marquardt algorithm as the backpropagation optimization technique. It is a combination of the gradient descent algorithm and the Gauss-Newton algorithm [147]. It is also identified as a fast algorithm for moderate size neural networks [148].

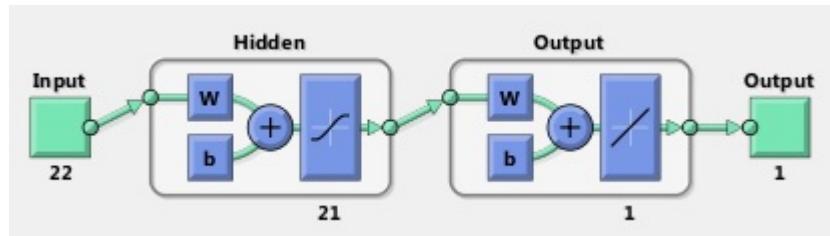


Figure 6.8: *sAllModel* Overview

### 6.3.1.2 Life cycle Cluster based Multi Layer Perceptron Model for sale events (*sClustModel*)

The *sClustModel* is introduced to utilize the life cycle shapes of games in the prediction model. In the prediction approach for non-sale periods, two models were introduced where the *nsAllModel* is a single model for all games approach and the *nsClustModel* is a prediction approach where a single model per cluster of games displaying similar life cycle shapes is generated. In the same manner, the *sClustModel* is introduced to be used in a cluster-based approach to predict the maximum population during sale events, in addition to the introduced *sAllModel* which is a one model per game approach.

In the *sClustModel*, one *sAllModel* per each cluster of games displaying the same life cycle shape is generated. Hence, the *sClustModel* is a combination of four *sAllModels* each specific to a cluster of games displaying same life cycle shape. The overview of *sClustModel* is depicted in Figure 6.9. The maximum population during a sale event is predicted by first identifying to what cluster the game of the sale event belongs, and then using the corresponding *sAllModel*. It is expected that using the life cycle shape in addition to the feature *Days since Release* in the *sClustModel*

would provide more prediction accuracy as it is a best practice to consider the age of a game in determining the discounts provided during the sale events [87]. In fact, the recommended best practice for sale events provided by Steam suggests increasing the discount percentage of a game in a stair-step approach and aim to schedule more sale events in the tail of the life cycle rather than at the beginning [87]. However, one limitation of the *sClustModel* approach is that it can only be used after the first year or a fixed period after game release after identification of the game life cycle shapes.

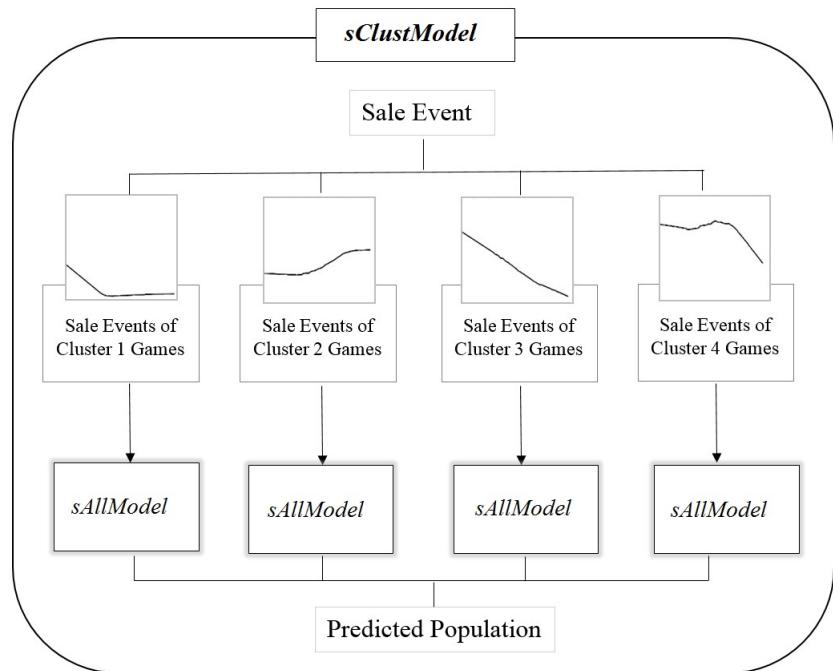


Figure 6.9: *sClustModel*: Life cycle shape cluster based Multi Layer Perceptron model overview: To predict the maximum population of a given sale event, first the cluster to which the game of the sale event belongs is identified, then the corresponding *sAllModel* is used to generate the prediction

### 6.3.1.3 Non-linear Autoregressive exogenous model (*NARXmodel*)

NARX model is the non-linear variant of the prominent linear autoregressive model which has exogenous inputs used in time series forecasting. In the NARX model the future values of a series are forecast using its past values and past values of other relevant independent variables known as exogenous variables. This is depicted in

Equation 6.3 where future values of the series  $y$  is forecasted [149].

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), x(t-1), x(t-2), \dots, x(t-n_x)) \quad (6.3)$$

The only difference between the *NARXmodel* and the *NARmodel* is that no exogenous variables are used in NAR and forecasts are solely based on the past values of the series. Thus, one purpose of introducing the *NARXmodel* in the study is to determine if more accurate predictions on population during sale events can be generated in a per-game approach using past price and discount data along with past population, compared to the *NARmodel*.

The NARX model uses recurrent neural networks (RNN) in which the output produced by the model is used as an input to the model. Moreover, since the NARX model is based on a neural network approach it is capable of identifying non linear relationships between variables from the data to generate accurate forecasts. Furthermore, it has been identified that unlike prominent statistical time series forecasting approaches such as ARIMA, neural network approaches do not require the input data series to be stationary where its statistical properties such as mean and variance does not change over time [106] [150]. Thus, the seasonality or trend of the series does not need to be removed.

The architecture of the *NARXmodel* introduced in this study is depicted in Figure 6.10. It uses the player population data as the main input and three exogenous inputs namely daily initial price, final price and discount percentage in order to forecast player population of the game. Specifically, in order to forecast the population at time  $t$ , the population and exogenous variable values of the past 7 days are used as input considering weekly seasonality. One purpose of introducing the *NARXmodel* is to determine if past population and price-related information of the game alone is sufficient to forecast player population during a sale event without access to the past history of other games. Since sale event information is not explicitly provided it is expected that the *NARXmodel* learns population changes related to discounts and price so that it implicitly learns the behaviour during sale events. In the *NARX-*

*model* the number of nodes in the hidden layer were chosen to be 10. Also, the *tansig* activation function is used in the hidden layer while the *purelin* activation function is used in the output layer. Although single step ahead forecast of population can be generated solely using the past values, multi-step ahead forecasting generation requires the population output value generated by the model to be fed back into the model through the closed loop. However, during the training process of the *NARXmodel*, the generated output is not fed back to the network to forecast the next value of the series. Instead, the expected output which is readily available is used as it is more accurate. Hence, the architecture of NARX takes the form of a feedforward network during training and learning is done through backpropagation.

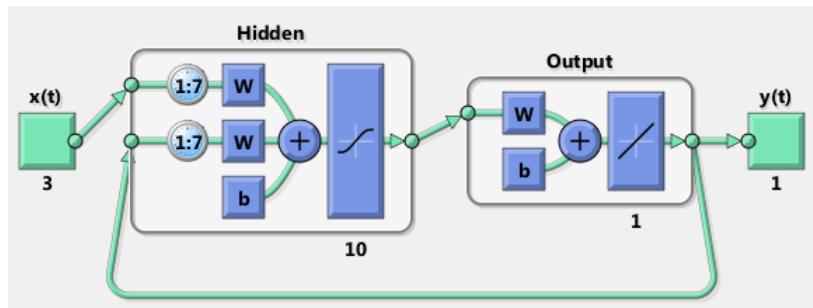


Figure 6.10: Architecture of Nonlinear autoregressive exogenous model (*NARX-model*)

### 6.3.2 Evaluation Procedure

The evaluation of the introduced sale event specific population prediction models is conducted by following the evaluation process presented in Section 6.2.2. It consists of a cross-validation approach conducted using three test sets. The sale event dataset is used to evaluate the three models (*sAllModel*, *sClustModel* and *NARXmodel*). The boundary dates used to split the dataset in the three iterations of the cross-validation process is depicted in Table 6.4. The RMSE values corresponding to each model in predicting the maximum player population during sale events is recorded. Moreover, the RMSE for test sets in each iteration of cross-validation is also recorded separately to investigate if there are any differences in the results obtained using different sized

training sets.

## 6.4 Results and Discussion

This section presents the results obtained from the experiments and a matching discussion.

Three sale event specific prediction models, namely, *sAllModel*, *sClustModel* and *NARXmodel* were used in this study to forecast the maximum player population during sale events of games. The outcomes of the evaluation process are depicted in Table 6.6 and Table 6.7 reporting the RMSE of models. Moreover, the regression plots of the *sAllModel* and the *sClustModel* related to test set 3 of the sale event dataset are also presented in Figures 6.11 and 6.12 which depicts the correlation between the actual and predicted values.

In Section 6.2.2 the evaluation of the three general population prediction models was presented (*nsAllModel*, *nsClustModel*, *NARmodel*). The evaluation revealed the prediction error that can be expected when using general population prediction models trained to predict population during non-sale periods, to predict population during sale event periods. Those models only used the population of the seven days prior to the sale event as input for generating predictions. The three sale event specific prediction models were introduced to investigate if the maximum population during sale events can be predicted more accurately using models that use sale event specific information. Hence, the three general population prediction models that are trained to predict population during non-sale periods are used as the baseline models. Their RMSE in predicting the maximum population during sale events is compared with the RMSE of the three sale event specific models.

As depicted in Table 6.7, both the *sAllModel* and the *sClustModel* has obtained a lower RMSE compared to the corresponding baseline models. Left-tailed two-sample t-tests revealed that the mean error of the *sAllModel* is statistically significantly lower than the mean error of the baseline model (*nsAllModel*) with a p-value of 4.7715e-20. A similar outcome was obtained for the *sClustModel* with a p-value

#### *6.4. Results and Discussion*

---

of 1.0028e-17. It can be seen that the RMSE of the sales models are now only approximately a half of the RMSE of the baseline models. Specifically, the RMSE of the *sAllModel* is only 0.6 times the RMSE of the *nsAllModel* and the RMSE of the *sClustModel* is only 0.7 times the RMSE of the *nsClustModel*. However, in the evaluation of the general population prediction models it was identified that the error values increase by roughly 6-fold when the general population prediction models are used to predict population during sale events. Hence, this indicates that the sale specific population prediction models have outperformed the general population prediction models in predicting population during sale events as the sale models have halved the error rates of the baseline model. This indicates that using sale event specific models that use sale event information along with past population can generate more accurate predictions regarding the population during sale events in both all-games-together and per-cluster approaches where sale event information of other games are also used in generating predictions.

Moreover, it was identified that the error of the *NARXmodel* is statistically significantly higher than the *NARmodel* with a p-value of 3.4808e-04. This indicates that when generating predictions about the population during a sale event in a per-game approach, more accurate predictions can be generated by using the game's past population history alone rather than using both population and sale event history of the game. However, this outcome is different from the outcome observed for the all-games-together and per-cluster approaches. Since the *NARXmodel* is a per-game approach, the higher error observed can be a result of a lack of sufficient past sale events in games to train the model accurately to predict population during future sale events. However, in the all-games-together and per-cluster approaches, this limitation is avoided as predictions are generated using the past sale event information of all games.

Both the *sAllModel* and the *sClustModel* not only use the past sale events of a given game but also the past events of other games in generating predictions about the population during a sale event of a game. As depicted in Figure 6.11, the re-

#### 6.4. Results and Discussion

---

Prediction Model	RMSE for Test sets		
	Set 1	Set 2	Set 3
<i>sAllModel</i>	0.1530	0.0933	0.1565
<i>sClustModel</i>	0.1543	0.0975	0.1658
<i>NARXmodel</i>	0.2159	0.2030	0.1995

Table 6.6: RMSE results from all prediction models for each test set in cross-validation; Set number represent the iteration number in cross-validation

Prediction Ap-proach	Sale Model Name	Baseline Model Name	RMSE	
			Sale Model	Baseline Model
All-Games-Together	<i>sAllModel</i>	<i>nsAllModel</i>	0.1367	0.1984
Per-Cluster	<i>sClustModel</i>	<i>nsClustModel</i>	0.1418	0.1986
Per-Game	<i>NARXmodel</i>	<i>NARmodel</i>	0.2062	0.1685

Table 6.7: RMSE results from all prediction models for all test sets combined: Baseline model results are the outcomes of predicting maximum player population during sale events of the sale event dataset using general population prediction models that were trained to predict maximum population during non-sale periods

gression correlation coefficient (R value) between the predicted and actual values for test set 3 by the *sAllModel* is 0.70 indicating an acceptable performance. Moreover, as depicted in Figure 6.12, the results of the *sClustModel* were somewhat different among the models of the three clusters. The regression correlation coefficient values for test set 3 are 0.60, 0.68 and 0.77 for Cluster 1, 2 and 3 respectively indicating an acceptable performance. Moreover, no model was created for Cluster 4 in the *sClustModel* approach as there were no games that belonged to Cluster 4 in the data set that matched with the initial game selection criteria. It was hypothesized that the *sClustModel* can generate predictions about sale events more accurately than the *sAllModel* as instead of a single model, one model per cluster of similar games based on life cycle shapes is utilized in the *sClustModel*. However, as per the RMSE values in Table 6.6, 6.7 and a left-tailed paired t-test conducted using the squared error values of the three test sets, the prediction error of the *sClustModel* was statistically significantly higher than the *sAllModel* with a p-value of 0.0078 at 0.05 significance level. On one hand, this indicates that there could be a negative consequence of generating separate models for games in each cluster rather than a

single model for all games as the number of games in each cluster are not similar. On the other hand, the results indicate that more accurate predictions about population during sale events can be generated using sale event information of all games together rather than only using sale event information of games that are similar based on the life cycle shape.

Moreover, it was identified that the error value of the *NARXmodel* is statistically significantly higher than the *sClustModel* and the *sAllModel* with  $p = 2.8200\text{e-}08$  and  $p = 3.1240\text{e-}09$ . The *NARXmodel* uses the past population history and sale event history of a game when generating predictions about population during sale events. The results indicate that, using sale event history of other games as in the *sClustModel* and the *sAllModel* without being limited to the considered game as in the *NARXmodel* can generate more accurate predictions about population during sale events.

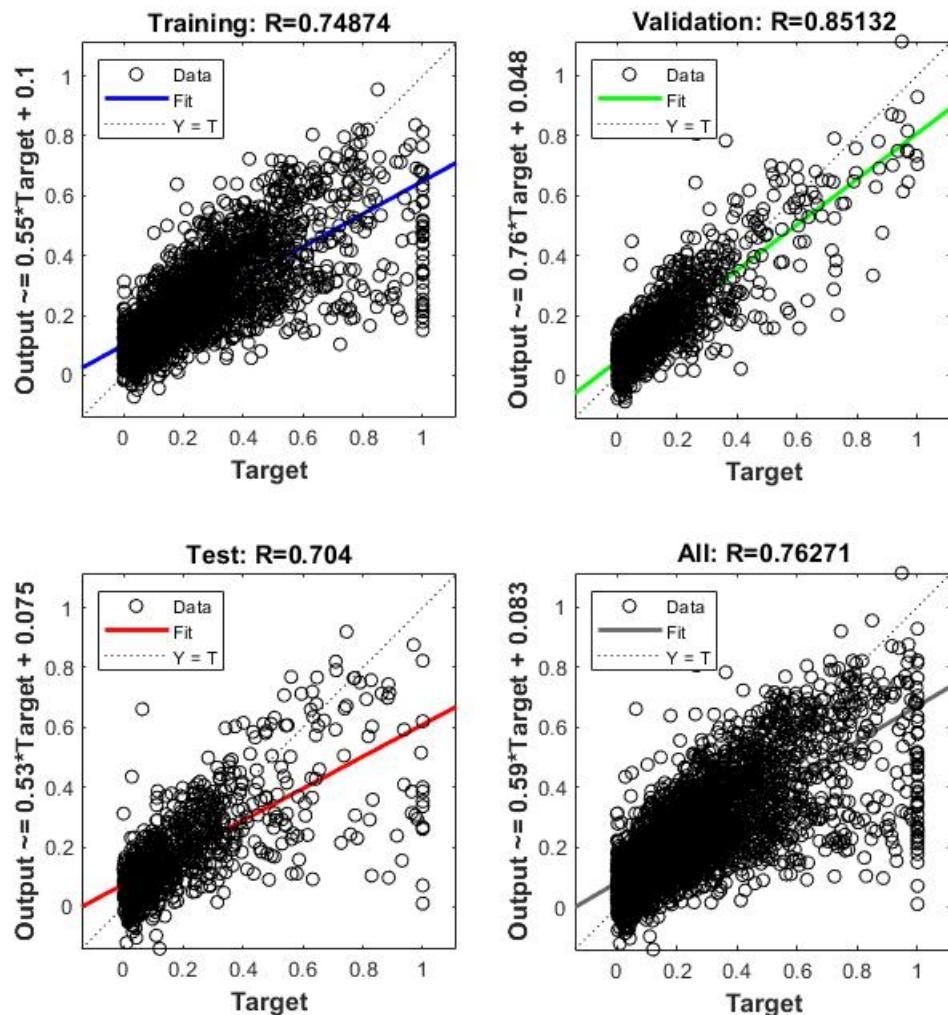
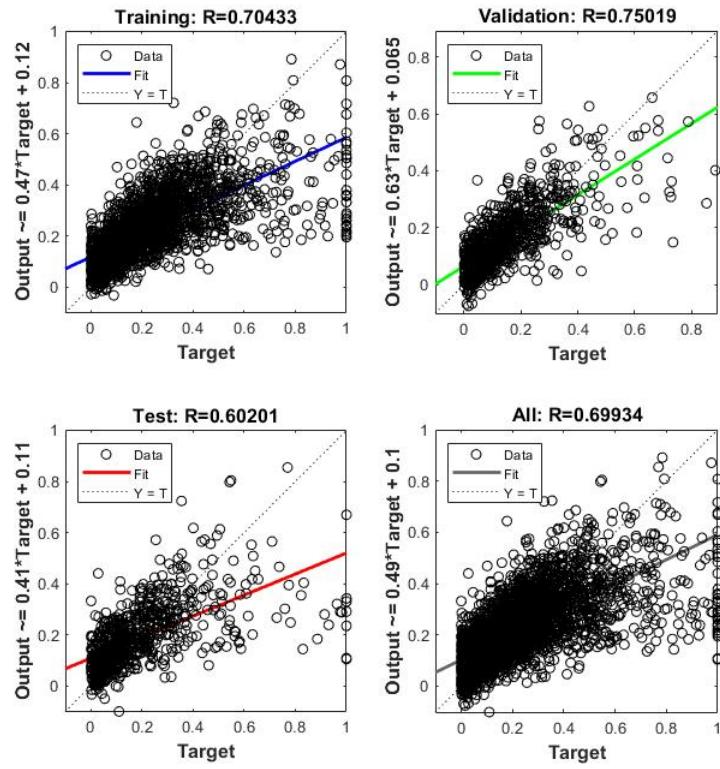
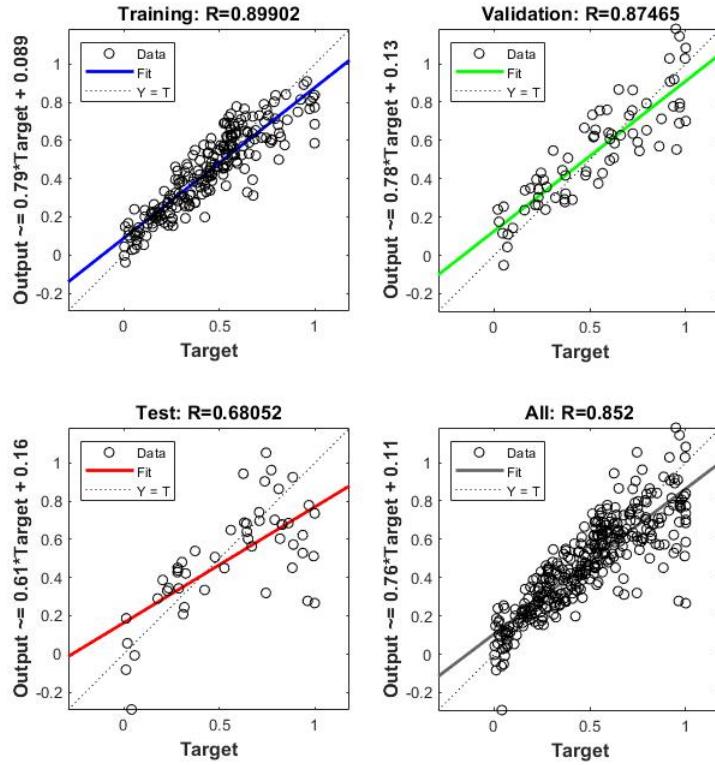


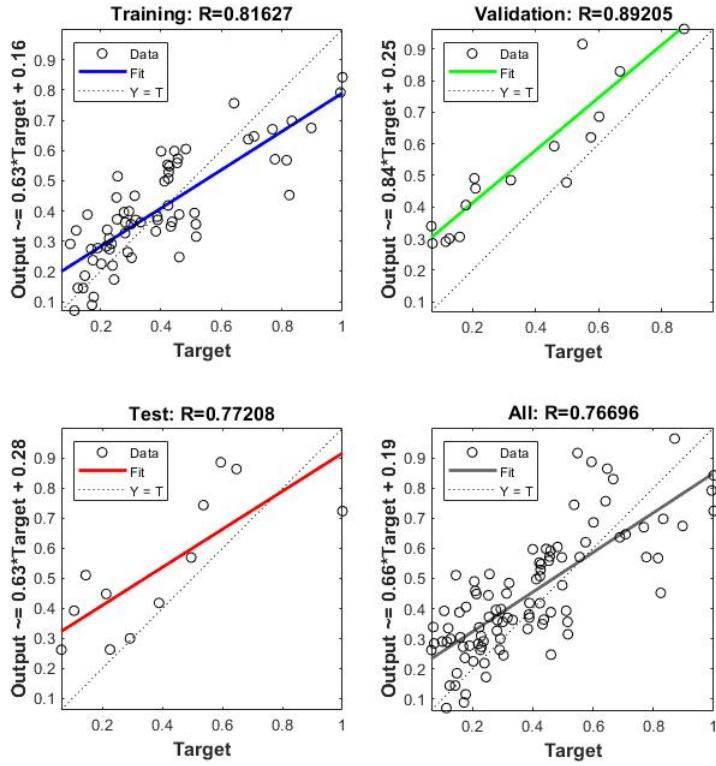
Figure 6.11: Regression plots of actual and predicted values resulting from *sAllModel* for test set 3: First plot depicts the regression plot of training dataset which has a R value of 0.74. Next plot depicts the regression plot for Validation dataset which has a R value of 0.85. It has the highest regression correlation coefficient out of the three datasets(training, validation and test). The next plot depicts the regression plot for test set which has a R value of 0.70. More under-predicted instances can be observed compared to over-predicted instances in each of the training, validation and test set regression plots. The final plot depicts the regression plot of all sets; training, validation and test, in a single plot which has a R value of 0.76



(a) *sClustModel* Cluster 1



(b) *sClustModel* Cluster 2



(c) *sClustModel* Cluster 3

Figure 6.12: Regression plot of actual and predicted values resulting from *sClustModel* for test set 3; Cluster 1, 2 and 3 plots are presented while Cluster 4 is not presented due to the unavailability of Cluster 4 games in the data set. Each sub figure depicts the regression plots for training, validation, test set and all those three sets together

The RMSE values reported in the study depicts the error in predicting population during either sale events or non-sale periods depending on the prediction model used. Hence, the reported RMSE values always correspond to either sale events or non-sale periods, but never to a combination of sale events and non-sale periods. The presented all-games-together and per-cluster based sale event specific population prediction models, namely, *sAllModel* and *sClustModel*, are trained to predict the maximum population during sale events. As explained in the chapter, the models use sale event specific information as input features. Hence, these models are sale-event specific and cannot be used to predict population during non-sale periods. Thus, the evaluation process also records RMSE values with regard to the prediction errors during sale events. Analysing the regression plots of actual and predicted values of

the models depicted in Figure 6.11 and Figure 6.12, reveals that there are more errors being made due to under-predicting of the population during sale events than to over-predicting. The study also used general population prediction models in all-games-together and per-cluster based approaches, namely, *nsAllModel* and *nsClustModel*, which only used past population to predict future population. The RMSE values of these models were higher in the scenario where these models were trained to predict population during sale events and used to predict population during future sale event periods, compared to the scenario where these models were trained to predict population during non-sale periods and used to predict population during future non-sale periods. Hence, it was identified that these general population prediction models that solely use the past population as input, predict the population during non-sale periods better than the population during sale events.

Overall, the results indicate that the introduced *sAllModel* has outperformed the other models, which are the proposed *sClustModel* and the time series forecasting approach *NARXmodel*. This indicates that for sale event related population forecasting, past sale event history of other games are also quite useful in addition to the game's own past events. The *NARXmodel* only used the game's own history but its performance is not better than the *sAllModel* or *sClustModel*. Moreover, both *NARXmodel* and *NARmodel* are per-game models. The prediction error of the *NARmodel* which only used the past population of the game was lower than the *NARXmodel* which used the price history of the game including daily initial price, final price and discount in addition to the game's past population history. This indicates that when generating per-game models for forecasting sale event population, only using the past population history of the game gives more prediction accuracy than when using price history as well. One reason for this could be the cold start problem in which the game has only a few past sale events for the model to accurately learn the behavior. To further validate if the lower performance of *NARXmodel* is due to a cold-start situation, further analysis was conducted through removing games that have a lower number of sale events. In the current dataset,

the number of sale events in each game has ranged between a minimum of 3 and a maximum of 42. The average number of sale events each game has is 17.42. Hence, to remove the games that have insufficient sale events, all the games that have less than 17 sale events were removed from the dataset. The experiments were conducted as before using the selected games that do not have the cold-start issue. The RMSE of the *NARXmodel* was 0.1667 and the RMSE of the *NARmodel* was 0.1581. The *NARXmodel* uses both past population and price history. Its prediction error has reduced from 0.2062 to 0.1667 when the games with insufficient sale event data were removed. This indicates the effect of the cold-start problem. The prediction error of the *NARmodel* has only changed slightly from 0.1685 to 0.1581. Unlike *NARXmodel*, the *NARmodel* only uses the past population. Hence, the removal of games with insufficient sale events has not introduced huge changes to the prediction error of the *NARmodel*. These outcomes validate that the cold-start problem could be a reason for the lower prediction accuracy of the per-game model that uses both past population and price history compared to the per-game model that solely uses the past population. Also, another reason for the lower prediction accuracy of the *NARXmodel* could be that the population fluctuations during the sale events of the game are highly varied. Hence, outcomes of sale events of the game's history are quite different from each other based on various other factors such as publicity provided to the event. However, when generating sale event population forecasting models, it is worth considering that a per game model approach is more resource-intensive than a single model for all games approach as when the number of games increases more models must be generated. Since the results indicated that the proposed single model for all games approach, *sAllModel*, outperformed the *NARXmodel* and the *NARmodel* it is possible to generate a less resource-intensive yet more accurate model for sale event related population forecasting.

## 6.5 Conclusion

This chapter presented a study to introduce a forecasting approach for accurately predicting the maximum player population during sale events of games; focusing on sale events in Steam. For this purpose, three prediction approaches were explored. The approaches are one model per game approach, one model for all games approach and one model per cluster approach which considers the similarity of games with respect to their life cycle shapes identified in Chapter 5. The study used 293 Steam games and their population and price history including the initial price, final price and discount to generate and evaluate prediction models.

The evaluation of the general population prediction models revealed that most accurate predictions related to the maximum player population during non-sale periods of a game can be generated by the *nsAllModel*. It indicates the effectiveness of a single model for all games approach for non-sale periods and the usefulness of an MLP model that uses population of the past seven days as input. Furthermore, it was identified that the maximum population during sale events can be predicted with more accuracy using sale event specific prediction models. Those models outperform the general population prediction models that are trained to predict population during non-sale periods, in both per cluster and all games together approaches. Specifically, it was observed that the prediction error increases by roughly 6-fold when a general population prediction model is applied to predict population during a sales period but, by using the sale event specific models that exploit sales information, that error roughly halves. This indicates the importance of using sale event specific information along with past population to generate predictions during sale events. However, the results were different for the per game approach. It was identified that for per game prediction models, using past population alone (*NARModel*) can provide lower error rates compared to using both past population and price history (*NARXModel*) to predict population during sale events. Moreover, the results also indicated that the maximum population during sale events can be predicted more accurately using past sale event information

of other games using the *sAllModel* compared to the other two sale event specific prediction models. Such a model is quite useful, as sale event related population predictions can be generated using both the game's sale event history and the history of other games as well. Thus, it is especially useful for games that do not have sufficient history of sale events. Furthermore, it was also identified that using the life cycle shape based similarity of games do not enhance the prediction accuracy in predicting the maximum population during both sale events and non-sale periods compared to having a single model for all games.

Predicting the player population of games during sales events is not a straightforward task and requires various data. The implications derived based on the outcomes of this study can be used by game developers and other game-related service providers to generate enhanced prediction models to forecast population during sale events. Since it was identified that the population prior to the sale event is the most important feature compared to the other features used in the all-games-together prediction approach, game companies can consider giving higher importance to the past population in generating prediction models that forecast population during sale events. However, based on the outcomes of the study it is suggested to game companies to generate sale event specific prediction models that use both past population and sale event details in all-games-together approaches, to get more accurate predictions regarding the population during sale events. This suggestion is made as it was identified that the general population prediction models that are trained to predict population during non-sale periods display higher error in predicting population during sale events compared to sale event specific models. Hence, in addition to the past population, it is suggested to use features such as end date of sale event, final price, discount, duration of event, days since last event, days since release, number of previous events, percentage of the final price relative to the last observed lowest price, month of the event and number of weekend days in the prediction models. Although game companies who develop multiple games can generate all-games-together prediction models, individual game developers would

not have data regarding other games to generate such models. Hence, individual game developers would have to create per-game prediction models to predict population during sale events. Based on the study, it is suggested to game developers to consider generating Nonlinear Autoregressive per-game models that solely use past population over Nonlinear Autoregressive Exogenous per-game models that use past population and price history to accurately predict population during sale events, especially, when there is insufficient past sale events in the game. Moreover, third-party game service providers can consider creating an all-games-together sale event specific prediction model that can predict the maximum population during sale events to provide prediction service to game developers, especially individual game developers.

The study conducted in this chapter was focused on addressing the research question “How accurately can we forecast player population of games during sale events?”. This was addressed by generating and evaluating three types of prediction models, namely, all-games-together, per-cluster and per-game models. It was identified that the generated sale event specific prediction models are capable of predicting the maximum player population during sale events more accurately compared to the generated general population prediction models. However, the main limitation of the study is that the prediction models were using a limited set of features. Not all features that could have relevance to sale events were captured due to the unavailability of data. Some of such features are the type of the sale event (eg:midweek madness, Lunar new year sale, developer scheduled sale), if the sale event was promoted in social media and if the game appeared on the first page of the relevant Steam sale page. Hence, one future work that arises from this study is to explore if the prediction accuracy of the models can be further enhanced by incorporating such external information. Moreover, since the life cycle shape based cluster approach did not provide better prediction accuracy than the all-games-together approach, it would be worth exploring other mechanisms to capture game similarity that can be used to obtain cluster-based prediction models for the purpose of sale

### *6.5. Conclusion*

---

event population prediction.

This chapter focused on the player population fluctuations of games in the presence of sale events of games. The next chapter presents a study conducted to investigate the player population fluctuations of games during the early period of the COVID-19 world crisis.

# Chapter 7

## Player Population Fluctuations during the Novel Coronavirus (COVID-19) Pandemic

Parts of the work reported in this chapter have been published in the following research paper;

1. **Dulakshi Wannigamage**, Michael Barlow, Erandi Lakshika and Kathryn Kasmarik, “Analysis and Prediction of Player Population Changes in Digital Games during the COVID-19 Pandemic,” 2020 in *Australasian Joint Conference on Artificial Intelligence*, Canberra, Australia, 2020, Springer, Cham.

The previous chapter was focused on player population changes during sale events of games. This chapter is focused on player population changes during a world event, specifically, the COVID-19 pandemic.

As explained in Chapter 2, the COVID-19 pandemic is a global health crisis that had massively impacted people’s normal day-to-day lifestyles. The digital game industry was mostly resilient to the impacts of the pandemic as demand for gaming increased in record numbers with people including children staying at home [151] and outdoor entertainment became inaccessible. However, since the COVID-19 pandemic is a very recent event, there are not many studies that have thoroughly investigated the changes in demand for games with respect to the potentially increased interest of people in gaming. Nonetheless, understanding player behaviour during

---

the pandemic, especially with respect to changes in player population size is quite useful to be better prepared for such future crises. It would aid in determining resource allocation and predicting demand and preference for various games helping the gaming industry to thrive during such crises. This chapter presents a study conducted to generate insights about the changes that occurred in the games industry with respect to player population during the onset of the pandemic. Since the pandemic is still continuing in the world at the time of writing, this study is focused only on the initial phase of the pandemic, specifically, the period prior to the 16th of April 2020. Using 500 popular Steam games, an empirical analysis is conducted to investigate the overall player population size changes and changes in daily and weekly population patterns. The analysis is conducted to understand the player behaviour and preferences during the initial phase of the pandemic. Moreover, several machine learning classification models are used along with an extended dataset of 1963 Steam games, to determine if accurate predictions about games that become highly popular during the early period of the pandemic can be generated.

The main results of the study indicate that the game player population has increased by 33% during the onset of the pandemic, especially after the 16th of March 2020 when the US announced compulsory social distancing. The population during the onset of the pandemic was not only higher than the population during the same period in the previous year, but also during major Steam sale event periods. The games *Jackbox party pack 3* and *Tabletop simulator* were among the games that had the highest population increase. Moreover, it was revealed that the majority of games did not display recurring weekly population patterns during the initial phase of the pandemic. The daily population patterns were also slightly different from the pre-COVID period patterns. Moreover, it was identified that games that become highly popular during the initial phase of the pandemic can be predicted based on game-related and population features with at most 0.69 accuracy utilizing classification models (decision tree, random forest, bagging, boosting and SVM). The tag analysis revealed Adventure, Racing, Multiplayer and Boardgames are of

increasing popularity during the onset of the pandemic.

This chapter first presents an empirical analysis of player population changes during the onset of the pandemic. Then, the classification approach to predict games that are highly popular during the onset of the pandemic is presented. Finally, the chapter conclusion is provided.

## **7.1 Empirical Analysis of Player Populations during the early period of the COVID-19 pandemic**

This section presents the study conducted to investigate player population patterns during the initial phase of the COVID-19 pandemic. The study explores two main aspects of game player populations; namely, the size of the population and short term seasonality of the population. Analysis of these aspects are useful to generate insights on changes that have occurred in player populations as a consequence of the pandemic.

### **7.1.1 Data Collection**

Player population data of 500 Steam games were used for the study. Two main criteria were considered in selecting those 500 games. Firstly, the population data of the games should be of high granularity where population data are collected in shorter intervals. This is important for closer analysis of population fluctuations, especially for daily patterns. Secondly, the set of games should be a representative sample of the popular games available in Steam. This is important as the focus is on player population fluctuations. Adhering to these criteria, the first 500 games of the *Gameset1* introduced in Chapter 3, were chosen for the study. *Gameset1* consists of the top 1963 Steam games that had the highest player population during the last 24 hours on the 11th of December 2017. Hence, the first 500 games of *Gameset1*

contains the top 500 games with the highest player population. Player population data of the selected games were collected in 5 minutes intervals. Population data from the 16th March 2019 to 16th April 2020, which is a period of over a year is used in the study.

Once the population series were chosen, data preprocessing was conducted. The data preprocessing procedure involved missing data imputation. Player population series contained 1.12% of missing data on average within the period of year considered in this study. Median filtering with a window of size 7 was applied to handle missing data. Median filtering replaces a missing data point with the median of its neighboring data points within a window. A small window of size 7 was chosen to impute the missing data based on the values closest to a data point rather than using a bigger window.

The COVID-19 cases data were retrieved from the dataset compiled by the Johns Hopkins University Center for Systems Science and Engineering [152]. The retrieved dataset contained information about daily COVID-19 cases since 22nd January 2020 to 15th of May 2020 at the time of retrieval.

### **7.1.2 Changes in Player Population Size of games**

The first analysis of the study is focused on the changes in player population size. It is anticipated that the player population size of games would display a growth during the onset of the pandemic period due to the changes in the general lifestyle of people. Thus, the analysis is conducted to investigate the changes in player population size focusing on three aspects. Specifically, the correlation between COVID-19 cases and player population, change of population during the onset of the pandemic compared to normal time and change of population during the initial phase of the pandemic compared to Steam sale event periods.

### **7.1.2.1 Correlation of COVID-19 cases and Player Population**

The correlation between aggregate player population and COVID-19 cases was calculated to determine whether there is any correlation between the growth of COVID-19 cases and player population changes of games. Since the COVID-19 cases were available with daily frequency, the population data were converted to a daily frequency. This is performed by calculating the mean daily population of the 5-min interval population data. Once population data of each game were converted to daily frequency, the aggregate daily player population was extracted by calculating the daily total population of all 500 games. Furthermore, the population trend was also extracted by smoothing the weekly population fluctuations by applying a moving average with a window size of 7. Next, the correlation between aggregate player population and COVID-19 cases was calculated using Pearson Correlation Coefficient as per Equation 7.1. In the Equation 7.1,  $X, Y$  represent the two series and  $\mu$  and  $\sigma$  represent the mean and standard deviation of the series respectively.

$$\rho(X, Y) = \frac{\sum_{i=1}^N (X_i - \mu_X)(Y_i - \mu_Y)}{\sigma_X \sigma_Y} \quad (7.1)$$

Since the United States has the highest percentage of Steam users compared to other countries [153], the analysis not only considered global COVID-19 cases but also United States COVID-19 cases. Figure 7.1 presents the aggregate player population and global, US COVID-19 cases since 22nd January 2020.

As depicted in Figure 7.1 there is a clear correlation between the COVID-19 cases and player population. The Pearson correlation between population trend and global cases was 0.8718 and the same for US cases was 0.7903 indicating a high positive correlation. Moreover, it can be seen that the population has increased after the 16th of March. Even the minimum population each week during this growth period, as represented by troughs, is almost as high as the maximum population each week during the pre-growth period, as represented by peaks. Numerically, the mean population after the 16th of March shows a 33% increase compared to the

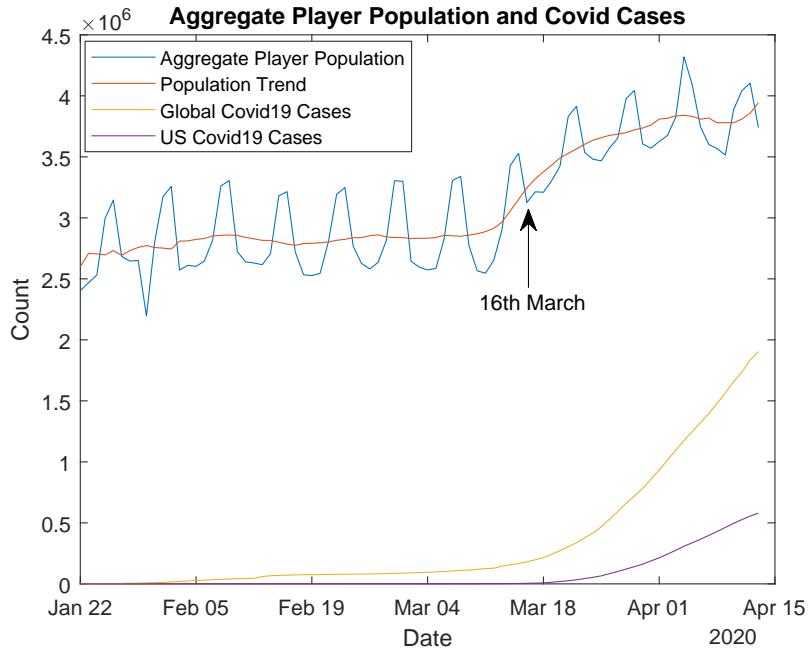


Figure 7.1: Data series of the daily aggregate player population, Global COVID-19 cases and US COVID-19 cases. 16th March is the date US president enforced restrictions on gatherings for up to 15 days. The Pearson correlation between the population trend and the global COVID-19 cases is 0.87. The Pearson correlation between the population trend and the US COVID-19 cases is 0.79.

mean prior to that period. It is noteworthy that the 16th of March is the date the US president announced social distancing restrictions for US residents. They have been advised to avoid gatherings of 10 or more people, to avoid eating and drinking at bars, restaurants and public food courts, to work or attend school from home whenever possible and to halt discretionary travel [154]. Overall, the results indicate that more people were turning to games as the severity of COVID-19 increases and social restrictions were in place.

Next, correlation analysis was conducted at the individual game level as well. Daily player population was extracted for each of the 500 games as before and Pearson correlation between population and global COVID-19 cases was calculated. Figure 7.2 presents the distribution of games with respect to the correlation value. It can be seen that most games display a positive correlation as the histogram is heavily skewed towards the right side. It was identified that 51% of games (254 games) display a positive correlation value which is higher than 0.5. Out of these the highest

correlation which is higher than 0.9 was displayed by 4 games, namely, *Tabletop Simulator*, *Knight Online*, *Motorsport Manager* and *Metin 2*. This indicates that the exponential growth pattern displayed by the COVID-19 cases series is quite similar to the growth pattern displayed by these four games during the early pandemic period.

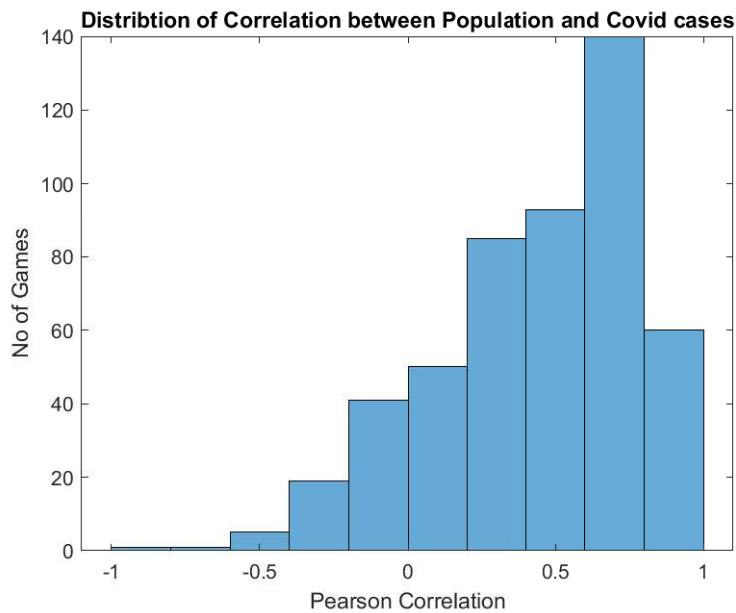


Figure 7.2: Distribution of correlation between daily global COVID-19 cases and population of individual games

### **7.1.2.2 Comparison of Population during the initial phase of COVID-19 and normal days**

In order to further investigate player population changes, a comparison between the player population during the onset of COVID-19 and prior to it was conducted. Population data of each game from 16th March to 16th April 2020 was used to represent the pandemic period. Population during the same time the previous year, 16th March - 16th April 2019, was used to represent the normal days or pre-pandemic period. This period was chosen to have no interference from daylight saving time changes. The mean player population during both periods were separately calculated for each game. The relative percentage of population change during the onset of the COVID-19 was calculated as per Equation 7.2. Figure 7.3 depicts the distribution of

mean population change percentage of games during the initial phase of COVID-19 pandemic. It can be observed that the mean population during the onset of the pandemic has increased in most games as most games have displayed a positive change percentage.

$$popChangePercent = \frac{(meanInPandemic - meanInPrePandemic)}{meanInPrePandemic} * 100 \quad (7.2)$$

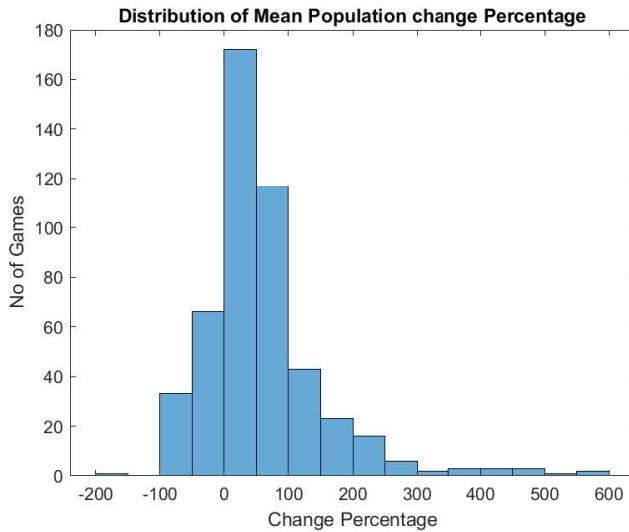


Figure 7.3: Histogram of mean population change percentage of games during COVID-19

Furthermore, it was identified that the games that had a positive correlation higher than 0.7 with COVID-19 global cases are also showing a positive population change percentage. In addition, the games that displayed a high positive change percentage were explored. 18 games were identified with a percentage higher than 300%. *The Jackbox Party Pack 3* and *Tabletop Simulator* displayed 520% and 480% respectively. *Tabletop Simulator* is a game where players can create and play tabletop games that also provide physics simulations just like playing in real life [155]. It includes classic games such as Chess, Dominoes, Poker and etc. *Jackbox Party Pack 3* is a local multiplayer game containing 5 party games such as Trivia Murder Party and Fakin it [156]. It is interesting that both of these games contain social games that people usually play when they are physically together. Two games, namely,

*Tabletop Simulator* and *Metin 2* appeared both in the set of games that displayed higher than 300% population increase and in the set of games that displayed the highest correlation with COVID-19 cases. *Half-life*, *Half-life 2*, *the Hunter Classic*, *the Hunter: Call of the Wild* are some other games that displayed a higher than 300% population increase. Another interesting game with a high percentage of population increase was *Plague Inc: Evolved* with a 245% increase. *Plague Inc: Evolved* is a Strategy and Simulation game where the player has to evolve a deadly plague to destroy human history [157]. This is somewhat relevant to the COVID-19 pandemic.

Additionally, the Wilcoxon signed rank test was conducted to statistically compare the mean population of games before and during the onset of the pandemic. The Wilcoxon signed rank test is a statistical hypothesis test for non-parametric paired data focused on differences of values before and after an intervention [158]. As opposed to paired t-test, the Wilcoxon signed rank test does not require the assumption of normal distribution to hold. The one-sample Kolmogorov-Smirnov test (K-S test), which is used to detect the distribution of a sample proved that the difference between mean population values before and during the onset of COVID-19 does not display a normal distribution. Hence the Wilcoxon signed rank test was selected. The right-sided test was conducted with a significance level of 0.05. This tests the null hypothesis that the median of the mean population differences during and before COVID-19 is zero against the alternate hypothesis that it is higher than zero. The null hypothesis was rejected with a p-value of 9.9686e-54. Thus, it can be claimed that the mean player population during the onset of COVID-19 has significantly increased compared to “normal” days.

#### **7.1.2.3 Comparison of Population during the onset of COVID-19 and Steam Sale event periods**

Steam sale events are widely popular among game players as games are sold for discounted prices. Analysis was carried out to identify if player population observed

during the onset of COVID-19 is higher than the population during major Steam sale event periods. The Steam Summer sale event which was held from 25th June 2019 to 9th July 2019 and Steam Winter sale held from 19th December 2019 to 2nd January 2020 were chosen for comparison. Mean player population during these sale periods and COVID-19 period were separately calculated as before for each individual game.

Median of the mean population values of games during the onset of the COVID-19 pandemic was 855.27. Median of the mean population values of games during Summer sale and Winter sale was 558.98 and 725.04 respectively, which are less than that during the onset of COVID-19. A right-tailed Wilcoxon signed rank test indicated that the mean population of games during the onset of COVID-19 pandemic is significantly higher than that of during Summer sales with a p-value of 6.8231e-41. Also, the same outcome was received for Winter sales with a p-value of 6.5660e-31. The Winter sale is held during December, a time where most people are enjoying holidays with more time available to spend on recreational activities including playing video games. It is interesting to observe that the population during the early days of the COVID-19 pandemic is even higher than that. Overall the results indicate that the player population during the onset of the COVID-19 pandemic is higher than the population during major Steam sale periods.

### **7.1.3 Weekly Patterns**

Game player populations display weekly seasonality in which player population fluctuation patterns repeat every week in a similar manner. In fact, it was identified in Chapter 4 that there exist 9 weekly player population patterns. However, due to the disruption to the normal living patterns of people due to COVID-19, it can be anticipated that the weekly population patterns of games could also be impacted. Thus, in this section, weekly player population patterns during the onset of COVID-19 are analysed.

Firstly, the games that display weekly seasonality during the initial phase of the

COVID-19 pandemic need to be recognized. For this purpose, the autocorrelation based weekly seasonality detection procedure discussed in Chapter 4 is applied. To recap, the process first removes trends from the population time series and then checks if the lag that has the highest autocorrelation falls within the window of lags representing a week.

Results revealed that out of the 500 games 76% of the games do not display weekly seasonality during the onset of COVID-19. However, 60% of the 500 games had displayed weekly seasonality during 16th March - 16th April 2019, a year prior to the COVID-19 period. Interestingly, out of the games that displayed weekly seasonality in the previous year 76% of games do not display weekly seasonality during the onset of COVID-19 pandemic. To illustrate, in Figure 7.4 it can be seen that the game *Garry's Mod* has the same player population fluctuation pattern recurring each week prior to COVID-19. In that, each week player population is almost similar for 4 days of the week and population increases in the last 3 days of the week. However, no recurring weekly pattern can be seen during the onset period of COVID-19 in that game. In the same way, *Rocket League* displays a weekly pattern where population slightly increases towards the end of each week prior to COVID-19 but no regular pattern during the early period of the COVID-19 pandemic as depicted in Figure 7.5. Such diminution of weekly seasonality in games during COVID-19 could have occurred due to several reasons. One reason is the disruption to a normal lifestyle in which people work from Monday to Friday and relax towards the end of the week. Furthermore, it can be understood that games are being played irrespective of the day of the week which could be a result of the stay at home advice put in place during COVID-19. Moreover, new players joining the games temporarily during the onset of the pandemic and only engaging in games in a casual way rather than consistently could be another reason for the diminishing of weekly seasonality. Figure 7.6 and 7.7 presents the non-normalized population versions of the previous Figure 7.4 and 7.5 indicating how population of those games has increased during the onset of the pandemic. In general, the

### *7.1. Empirical Analysis of Player Populations during the early period of the COVID-19 pandemic*

---

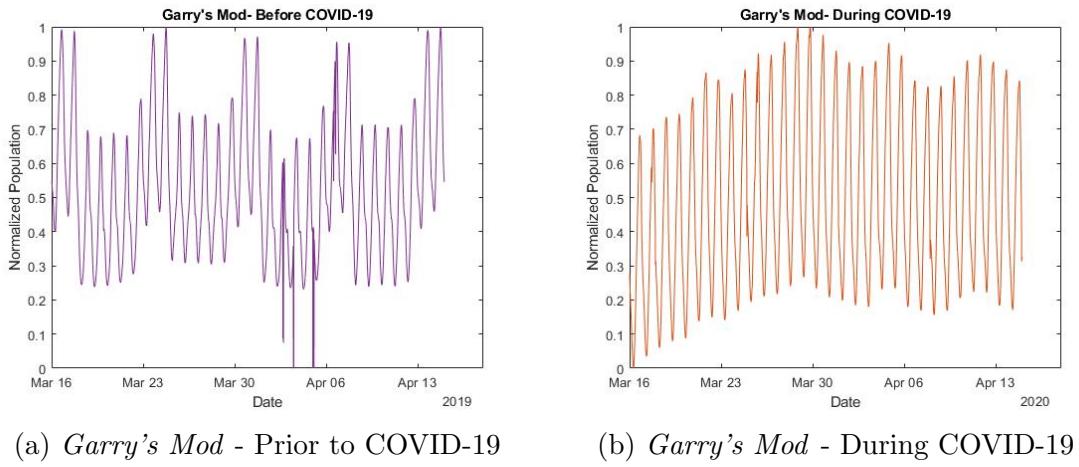


Figure 7.4: *Garry's Mod*- Normalized Player Population during and prior to COVID-19

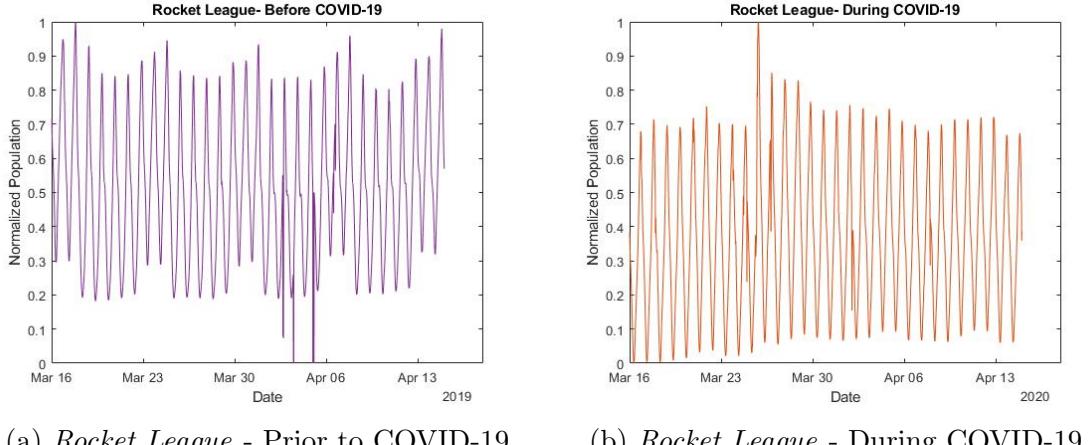


Figure 7.5: *Rocket League*- Normalized Player Population during and prior to COVID-19

results indicate that while player population has increased during the onset of the COVID-19 pandemic as seen in the previous section, player population fluctuations no longer display weekly seasonality.

#### **7.1.4 Daily Patterns**

Player population fluctuation of games display recurring daily cycles as recognized in Chapter 4. Such patterns could happen when most players tend to choose the same time of the day each day to play games. This could also be influenced by the natural day-night cycle and the work cycles (9-to-5 jobs). Since the COVID-19 restrictions have encouraged staying at home and working from home influencing normal daily

### *7.1. Empirical Analysis of Player Populations during the early period of the COVID-19 pandemic*

---

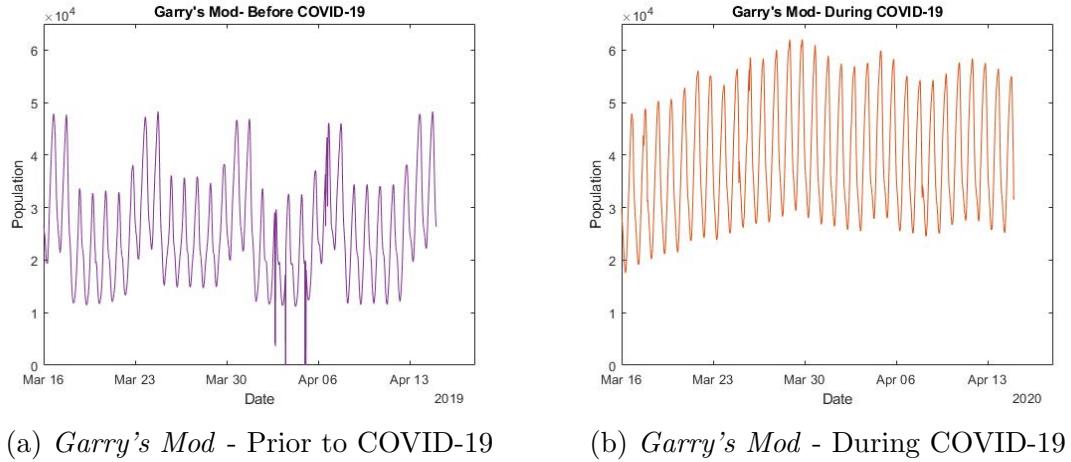


Figure 7.6: *Garry's Mod*- Player Population during and prior to COVID-19

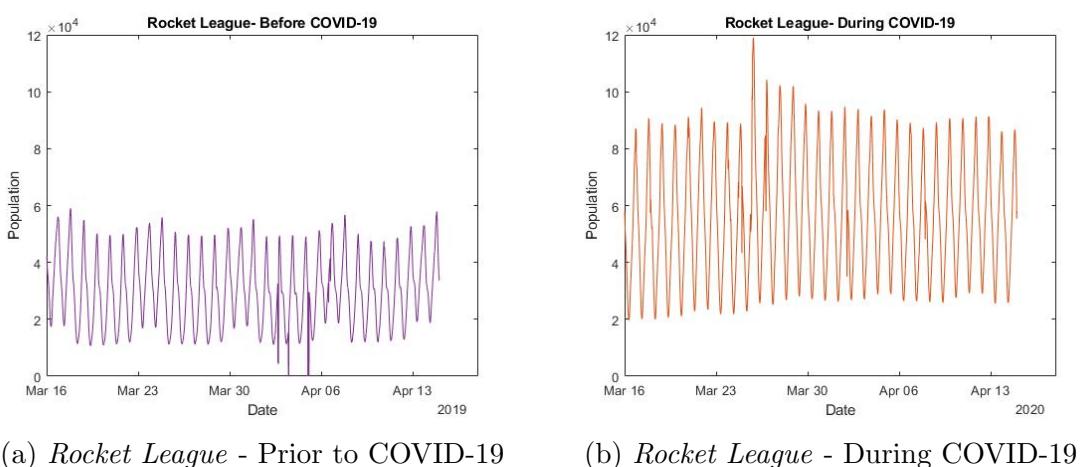


Figure 7.7: *Rocket League*- Player Population during and prior to COVID-19

routines it is hypothesised that the daily player population fluctuation patterns could have been impacted. Thus, in this section daily patterns are investigated.

The aggregate daily population pattern of all games is first generated for the selected pre-COVID-19 period and during the onset of COVID-19 period. Similar to previously the pre-COVID-19 period is a month from 16th March 2019 and the onset period of COVID-19 pandemic is a month from 16th March 2020. To generate the aggregate pattern the daily average player population pattern of each game is extracted by averaging all the per day data points of the game. Next, the total of the daily patterns of all games is calculated to generate the final aggregate pattern. However, due to time zone differences the aggregation cannot be performed directly and a realignment is required. For instance, it can be seen in Figure 7.8a and 7.8b that although the daily pattern shape is quite similar in both games a direct aggregation of the two patterns would result in a non-representative daily shape. However, if one pattern could be delayed appropriately a better alignment for aggregation can be found. Hence, a realignment is carried out for each game's extracted daily pattern. Figure 7.8c and 7.8d show the realigned daily population pattern of the previous two games given in Figure 7.8a and 7.8b.

The realignment process is presented in Algorithm 7.1. In the algorithm, the index of the minimum and maximum points of a given daily pattern is first identified. Based on these indexes the data series is broken into three subseries and those are rearranged to create the realigned pattern. The realignment process is slightly different between the situations where the maximum value is positioned at a location on the right side of the minimum value and, the maximum value is positioned at a location on the left side of the minimum value. If the maximum value is positioned at a location on the right side of the minimum value, data points from the index of the minimum point up to the index of the maximum point are extracted. Those data points are placed as the initial set of data points to create the realigned pattern. The remaining data points of the original daily pattern located at the right side of the index of the maximum point is extracted to use as the next set of data points for the

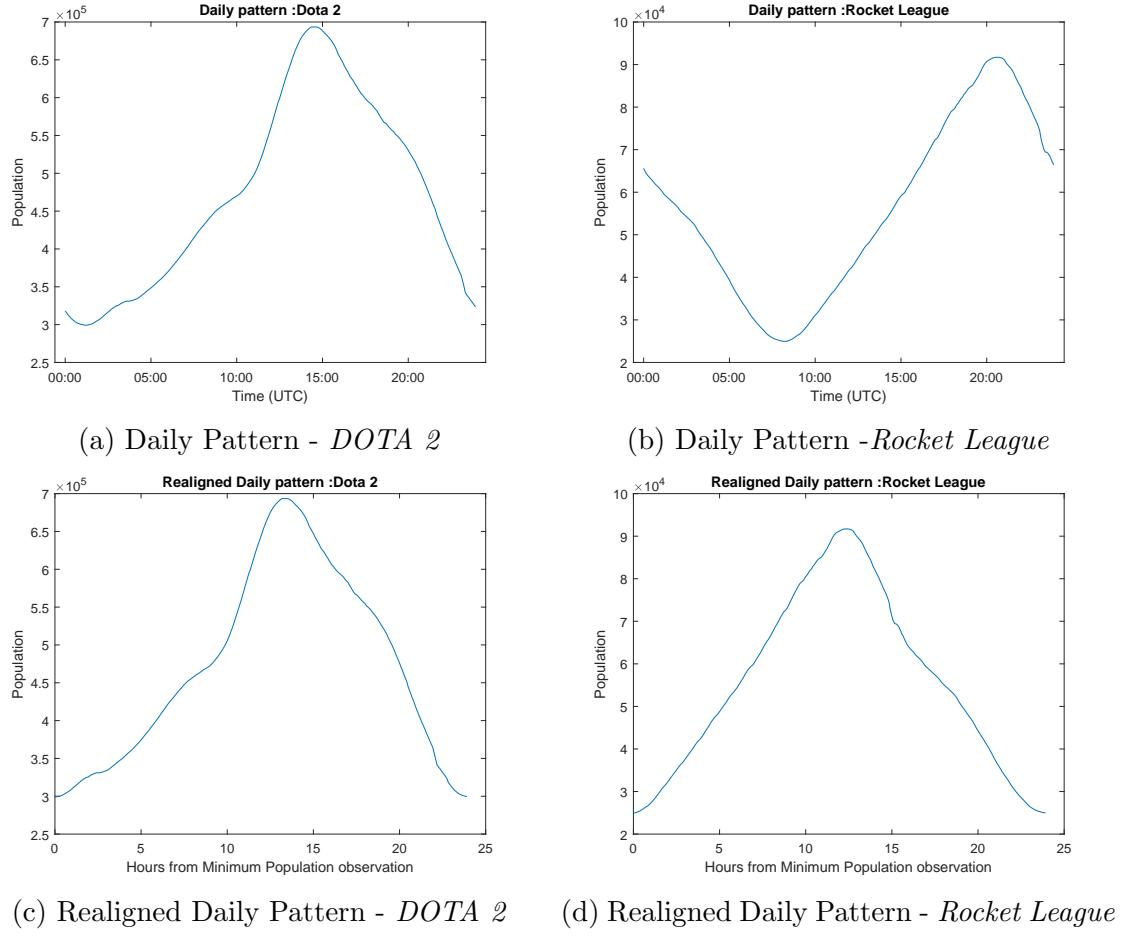


Figure 7.8: Daily Mean Population Pattern of *DOTA 2* and *Rocket League* during the onset of COVID-19 before and after realignment using Algorithm 7.1

realigned pattern. Finally, the remaining data points of the original daily pattern located at the left side of the index of the minimum point is extracted to use as the last set of data points for the realigned pattern. The realigned pattern contains these three subsets of data points of the daily pattern arranged in the order explained. The same method of realignment is repeated with changes to the ordering of the three subsets in the situation where the maximum value is positioned at a location on the left side of the minimum value in the original series. The complete process is presented in Algorithm 7.1. This process realigns the daily pattern of a game minimizing the impact from time zone influence. Also, the time is indicated relative to the minimum population observation point in the realigned pattern instead of UTC time due to the removal of time zone influence. The aggregate daily population pattern of games is generated by aggregating realigned patterns of each game.

**Algorithm 7.1** Daily Population Pattern Realignment

---

```
Inputs: dailyPopPattern
maxIndex = location of max(dailyPopPattern)
minIndex = location of min(dailyPopPattern)
if maxIndex > minIndex then
    leftVals = dailyPopPattern(minIndex:maxIndex)
    rightVals = [dailyPopPattern(maxIndex+1:end) , dailyPopPattern(1:minIndex-1)]
else if maxIndex < minIndex then
    rightVals = dailyPopPattern(maxIndex+1:minIndex-1)
    leftVals = [dailyPopPattern(minIndex:end),dailyPopPattern(1:maxIndex)];
end if
realignedDailyPopPattern = [leftVals,rightVals]
```

---

The aggregate daily population pattern during and prior to COVID-19 is presented in Figure 7.9. It can be seen that both patterns are quite similar in shape, but different in magnitude. This is due to the population increase observed during the onset of COVID-19 pandemic. In order to focus on the shape alone, a normalization process is executed. Once each game's mean daily pattern is extracted the values are rescaled to the 0 - 1 range prior to the realignment step. Per game rescaling is performed to obtain similar influence from each game to the final pattern, irrespective of the population size of each game. A final daily pattern is generated by calculating the total of the rescaled and realigned patterns of all games and dividing by the number of games. The normalized mean daily pattern of games is presented in Figure 7.10. It can be seen that the daily pattern during and prior to COVID-19 are quite similar in shape. However, the population during the first half of the pattern is higher during the onset of COVID-19 period. A right-tailed Wilcoxon signed rank test also indicated that the population during the first part of the daily pattern during the onset of COVID-19 is statistically significantly higher than that of before COVID-19 with a p-value of 4.48e-44. This indicates that more players can be observed during the section of the daily pattern where player population gradually increases from minimum to maximum, compared to the period where player population decrease from maximum to minimum during the day. Furthermore, the difference seems higher closer to the maximum population point. One reason for

### *7.1. Empirical Analysis of Player Populations during the early period of the COVID-19 pandemic*

---

this could be the stay at home restrictions. Rather than waiting for the evening or end of the day players have more freedom to play throughout the day, as they stay at home. This could increase the population observed during the first part of the daily pattern.

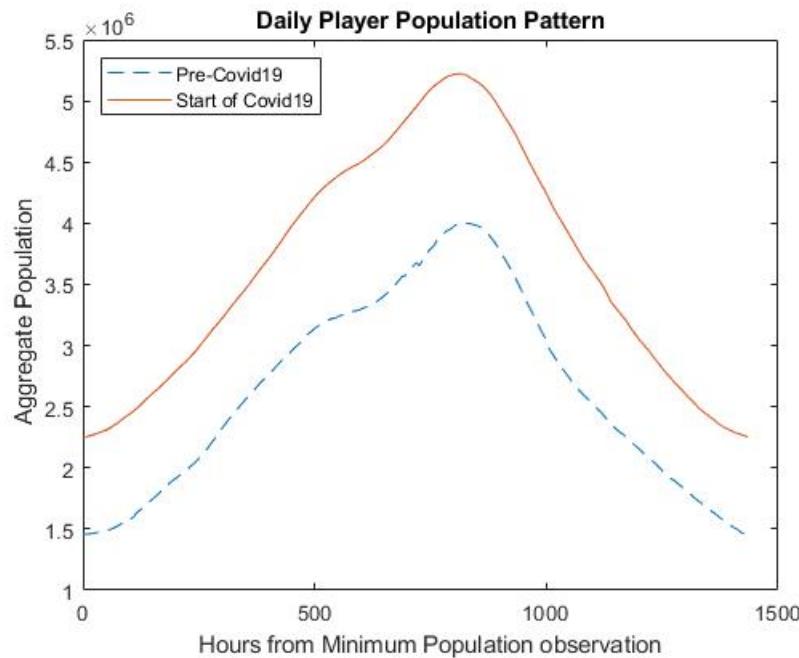


Figure 7.9: Aggregate Daily Player Population Pattern during and prior to COVID-19

Further analysis is conducted to explore the daily population pattern focusing on the duration between the time minimum daily population is observed and the time the maximum daily population is observed. The length of this duration observed during the onset of the COVID-19 and prior to COVID-19 is compared. The length of the duration between the time the minimum daily population is observed and the maximum daily population is observed is measured in minutes for each game. Then the Wilcoxon signed rank test is applied to analyse the duration difference. Results indicate that there is no statistically significant difference between the duration to reach the daily maximum population from the time the daily minimum population is observed at the onset of COVID-19 period and before COVID-19 period with a p-value of 0.1353. The median of the duration before and at the start of COVID-19 is similar and is 12 hours and 45 minutes. This indicates that the time it takes to

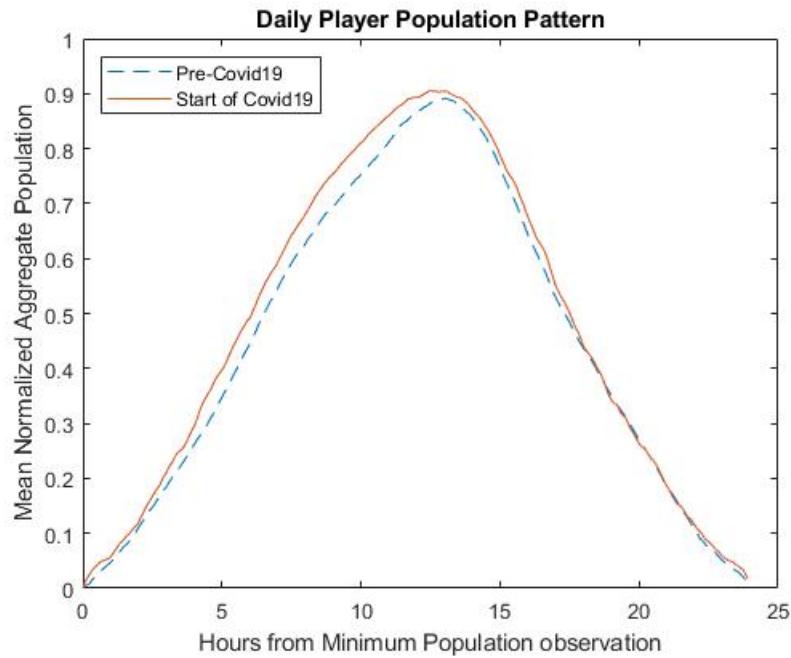


Figure 7.10: Mean normalized Daily Player Population Pattern of all games during and prior to COVID-19

reach daily maximum population since the time the daily minimum population is observed during a day is not significantly impacted by COVID-19. This could be because the day-night cycles are not impacted.

Another analysis is performed to investigate the exact time the maximum population and minimum population is reached. Although the time duration between minimum and maximum population observed points has not varied significantly, the absolute time in which the population reaches the daily minimum and maximum could change. To explore this aspect the mean daily population pattern of each game is used without the realignment phase. For each game, the number of minutes from the first data point of the day to the maximum population reaching point and to the minimum population reaching point is recorded separately. The measured length during the onset of COVID-19 and prior to COVID-19 period are compared. The Wilcoxon signed rank test revealed that there is no statistically significant difference between the time to reach maximum population prior to COVID-19 and during the onset of COVID-19 with a p-value of 0.072. Similarly, there was no statistically significant difference between time to reach the minimum population

in both situations with a p-value of 0.257. Hence, it can be seen that neither the duration between the time daily minimum population is reached and daily maximum population is reached changed nor the exact time at which daily minimum and maximum population is reached due to COVID-19.

Overall, the analysis of daily patterns indicate games having marginally more players during the first half of the day during the onset of COVID-19 period compared to pre COVID-19 period. However, the time to reach the daily maximum player population has not changed nor the duration from daily minimum to maximum population reaching times due to COVID-19.

In summary, the empirical analysis presented in this section revealed various changes that happened in games with respect to game player populations during the initial period of COVID-19. Not only did the player population of games increase but also the daily and weekly player population patterns were impacted. The study reveals an increased interest of people towards gaming during the onset of the pandemic. One reason for this increased interest could be people including children staying at home having more freedom to play at any preferred time. Furthermore, the lack of other outside entertainment options such as movie theatres and parks also could contribute towards the increased interest. Hence, it can be understood that growth in the number of game players could be expected during such pandemic situations.

The next section investigates if games that become popular during the onset of the pandemic can be predicted based on game-related features by generating several machine learning classification models.

## **7.2 Predicting the popularity of games during the onset of the pandemic**

This section presents a binary classification approach to predict what games become highly popular during the onset of the pandemic where popularity is determined

based on the population change percentage.

The COVID-19 pandemic is a world crisis event during which the video game industry gained increased interest from the public. As identified in the empirical analysis in the previous section, a majority but not all games observed an increase in player population during the onset of the pandemic. Hence, the ability to predict what games become popular during such a world crisis is useful for game companies to better plan for similar future crises. Especially, it would aid in planning ahead to serve more players and to understand what kind of games to promote and produce during such situations. Hence, several classification models to predict games that would become highly popular during the onset of the pandemic are generated and evaluated in this section.

Prior to presenting the classification approach the term *popular* needs to be defined in this context. The popularity of a game during the initial phase of COVID-19 pandemic is determined based on its percentage of population increase as the population of a game is indicative of the popularity. Hence, the top 20% of the games with the highest percentage of population increase were labelled as highly popular. The population increase percentage of these games was higher than 74%. The bottom 20% of the games with the lowest percentage of population increase were labelled as not highly popular. The population increase percentage of these games was less than 4.3%. The percentage of population increase is calculated as per Equation 7.2, where the pre-pandemic period is one month from 16th February 2020 and pandemic period is one month from 16th March 2020. Binary classification models are used to predict whether a game is significantly popular or not based on these labelling criteria.

### **7.2.1 Data Collection**

An extended version of the game dataset used for the empirical analysis in the previous section is used. Specifically, instead of using only the top 500 Steam games the top 1963 games of the *Gameset1* after removal of non-games are initially chosen.

The 1963 games contained 1.43% missing data on average. Hence, missing data of each game's population series were imputed by applying median filtering as before. Adhering to the labelling criterion mentioned earlier, the top 20% and the bottom 20% of the 1963 games based on their population increase percentage during the pandemic were chosen for the study. Moreover, game-related features are also used. Game-related feature collection process was introduced in Chapter 3.

### **7.2.2 Feature Extraction**

Features related to the games were extracted from the collected data to be used for the classification models. Table 7.1 depicts all the extracted features. The features are related to the state of the game prior to the pandemic where the 16th March 2020 is used as before to indicate the boundary date for the pandemic period.

The price was chosen as the price could have an influence over the purchasing decision of a game. Moreover, the days since release was chosen as it represents the age of a game. The first 3 most frequent tags of each game are chosen as they represent the type of the game. Moreover, tags were chosen instead of genres as tags are more descriptive. Since a positive review percentage can represent how games are embraced by players it was chosen. Each game in the dataset has one or many developers. However, due to the high diversity of developers where on average each developer is associated with around 1.4 games, it is not appropriate to use the developer as a categorical feature. Instead, whether the developer of the game is a top developer or not, decided based on the number of games in the dataset they have developed, is used as a binary categorical feature. In the dataset, the developer who has developed the highest number of games have developed 44 games, representing 2.2% of games in the dataset. Hence, several values less than 44 were explored to choose a threshold for the number of games developed to determine whether a developer is a top developer or not. Specifically, 30, 20 and 10 were explored which approximately represents 1.5%, 1% and 0.5% of games in the dataset. However, since there were no significant difference in the outcomes obtained when these values are

Feature	Description	Type
Price	The most frequent price within the two weeks prior to the pandemic	Continuous
Days since Release	The number of days since the game release by the 16th March 2020	Discrete
Tag1	The most commonly applied tag to the game	Nominal
Tag2	The second most commonly applied tag to the game	Nominal
Tag3	The third most commonly applied tag to the game	Nominal
Positive Review Percentage	The percentage of positive reviews	Continuous
Top Developer	Is the developer of the game one of the top developers	Nominal
Top Publisher	Is the publisher of the game one of the top publishers	Nominal
Number of Packages	The number of packages for the game in Steam	Discrete
Number of supported Languages	The number of languages the game supports	Discrete
Pre-COVID Mean Population 1 year	Mean population during the same period last year; 16th March - 16th April 2019	Continuous
Pre-COVID Mean Population 1 Month	Mean population during the month prior to the onset of pandemic; 16th February - 16th March 2020	Continuous

Table 7.1: Features used in the classification models

used, it was decided to use 10 as the threshold. Hence, a developer is considered as a top developer in this context if they have developed 10 or more games in the dataset. The top publisher feature is also extracted similarly.

The set of extracted features contain both numerical features and categorical features. Numerical features have either continuous values or discrete values. Categorical features have either nominal values or ordinal values where nominal values have no order while ordinal values have an order associated with them. Hence, the choice of the potential classification models depends on the ability to handle both numerical and categorical features.

Moreover, after analysing missing values in the dataset, several data instances that had missing feature values were removed. Hence, the final dataset used for the

classification model generation process contained 719 games. In the dataset, 49% of games belonged to the highly popular class and 51% of games belonged to the not highly popular class. Since the dataset was not highly imbalanced it was directly used without any imbalanced data handling approaches.

### **7.2.3 Classification Models**

Several machine learning classification models were trained to predict the games that are highly popular and not highly popular during the onset of the pandemic using the selected feature set. The classification models used for the task are Decision Trees, Decision Tree Ensembles (Boosting, Bagging and Random Forest) and Support Vector Machines (SVM). These classification models not only support both numerical and categorical features but also regarded as prominent classification models in the literature.

All four tree-based classification models were trained using the classical CART (Classification And Regression Tree) algorithm [159] in which the Gini index is used as the tree node splitting criterion. The AdaBoost algorithm [160] was used for the Boosting model. For the Random Forest model, the size of the random feature subset at each node was chosen to be the square root of the total number of features as it is the typical value used [161]. Moreover, each of the ensemble models (Boosting, Bagging and Random Forest) contained 100 trees. A radial basis function kernel was used for SVM and all the features were standardized prior to training it. Also, Bayesian optimization was used for hyperparameter optimization of all the models as it is less time consuming than the grid search.

### **7.2.4 Evaluation**

The following measures are used to evaluate the performance of the classification models.

$$Accuracy = \frac{Correct\,Predictions}{Total\,Predictions} \quad (7.3)$$

$$Precision = \frac{TruePositive}{(TruePositive + FalsePositive)} \quad (7.4)$$

$$Recall = \frac{TruePositive}{(TruePositive + FalseNegative)} \quad (7.5)$$

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \quad (7.6)$$

$$G - Mean = \sqrt{Sensitivity * Specificity}$$

$$Sensitivity = \frac{TruePositive}{(TruePositive + FalseNegative)} \quad (7.7)$$

$$Specificity = \frac{TrueNegative}{(FalsePositive + TrueNegative)}$$

In this study the positive class is the highly popular games while the negative class is the not highly popular games. Hence, *TruePositive* represents the number of instances predicted to be highly popular and are actually labelled as highly popular while *FalsePositive* represents the number of instances predicted to be highly popular but actually labelled as not highly popular. Also, *TrueNegative* is the number of games that are predicted as not highly popular and are actually not highly popular while *FalseNegative* is the number of games predicted as not highly popular but are actually highly popular.

Precision provides an indication of the likelihood that a game predicted as highly popular is actually highly popular. Recall indicates the likelihood that a game that is actually highly popular is predicted as highly popular. Hence, both precision and recall are relevant to the model's performance in predicting the highly popular games. Maximizing both precision and recall is difficult due to the existence of a trade-off between them. F-Measure represents the model's performance with respect to both precision and recall in a single measure. It is suitable for use cases where precision is not more important than recall and vice versa. Sensitivity, similar to recall, is a measure of the model's capability in predicting highly popular games while specificity indicates the capability of predicting not highly popular games. G-Mean, known as geometric mean is a combination of these two measures. Hence, it provides an indication of the overall performance of the model with respect to

predicting both highly popular and not highly popular games.

The evaluation of the classification models is performed using 10-fold cross-validation. It is performed by dividing the dataset into 10 equal-sized partitions and using each partition as the test set and the rest as the training set. A 10-fold cross-validation was performed for 30 runs and the mean value of each performance measure was recorded.

### 7.2.5 Results

The performance results of the classification models related to predicting popular games during the onset of the pandemic are depicted in Table 7.2. As per the table, highly popular games during the onset of the pandemic can be predicted with at most 0.69 accuracy with the Random Forest model. The confusion matrix of the Random Forest model is presented in Figure 7.11. As per Table 7.2, the SVM model has the closest similar accuracy to the Random Forest model. The lowest accuracy is displayed by the Decision Tree model. The SVM model has the highest precision in predicting the highly popular games during the onset of the pandemic, whilst the Random Forest model has the highest recall. Overall, results indicate that the Random Forest model has the highest overall performance in classification using game-related features. Random Forest is not only better at predicting highly popular games indicated by Fmeasure results but also at predicting both highly popular and not highly popular games indicated by Gmean results, compared to the other models.

Model	Accuracy	Precision	Recall	F measure	G mean
<b>Decision Tree</b>	0.62	0.62	0.59	0.60	0.62
<b>Boosted Trees</b>	0.65	0.66	0.62	0.64	0.65
<b>Bagged Trees</b>	0.65	0.64	0.66	0.65	0.65
<b>Random Forest</b>	0.69	0.69	0.67	0.68	0.68
<b>SVM</b>	0.67	0.72	0.57	0.64	0.67

Table 7.2: Performance of Classification Models

However, the performance of models are generally only above average as even the



Figure 7.11: Confusion Matrix of the Random Forest model: The confusion matrix of an evaluation run of the random forest model depicting true and predicted classes. The row-wise percentages represent the percentage of correctly and incorrectly classified instances for each true class out of the total instances of the respective true class. The column-wise percentages represent the percentage of correctly and incorrectly classified instances for each predicted class out of the total instances of the respective predicted class.

best performing model, Random Forest has an accuracy of only 0.69. It indicates that there is considerable room for improving the accuracy. Since COVID-19 is an unprecedented event it is likely that the game features used for the classification models alone might not be sufficient to predict the games that become highly popular. People's decisions on playing games and what games to play might also depend on various other circumstantial factors. For instance, the dominant country of the game and lockdown or any other constraints in the country are some external factors that might have an influence on what games become popular. Hence, it might be beneficial if access to such information is available to further improve the prediction accuracy of classification models.

### 7.2.5.1 Prominent tags of the highly popular games

A tag analysis was conducted to analyse the games that have become highly popular during the pandemic. In order to identify the most common tags that appear in

the highly popular games class compared to not highly popular class, the difference between the percentage of games related to each tag in the two classes was calculated. Specifically, for each tag, the percentage in not highly popular class is subtracted from the percentage in highly popular class. The percentage difference is used instead of the percentage in the highly popular class alone as some tags are quite common among games irrespective of the class, making a ranked list using only the highly popular class less meaningful. The top 3 tags of each game in the dataset were used for percentage calculation. Figure 7.12 depicts the prominent tags in the highly popular games class along with the difference between the percentages of games associated with each tag of highly popular class and not highly popular class. Only the tags that had a percentage difference higher than 0.5% are depicted in the Figure 7.12. The highest difference is only 7.7% and displayed by the Adventure tag. The differences displayed by the other tags are all less than 4.5%. Such a lower difference in percentage indicates that the individual level tag distributions of both classes are not highly different. However, there might be more differences among tag combinations.

It can be seen in Figure 7.12 that the Adventure tag has the highest difference making it the most prominent tag to distinguish highly popular games. The second most prominent tag appears to be Racing whilst the fourth most prominent tag is Automobile Simulator. Some of the highly popular games with Racing and Automobile Simulator tags were *Motorsport Manager*, *RaceRoom Racing Experience*, *Car Mechanic Simulator 2015*, *TrackMania Nations Forever* and *F1 2012* to *F1 2016*. The popularity of the *F1* games series is possibly a result of the start of Formula 1 season on the 15th of March 2020, when the severity of the pandemic increased, and the disruption caused to it by the pandemic resulting in the launching of a new F1 Esports Virtual Grand Prix series where F1 racers play the F1 2019 game which was streamed across social media [162]. The Multiplayer tag is the third most prominent tag. Moreover, the existence of Massively Multiplayer and Co-Op (Co-operative) along with Multiplayer tag indicates that Multiplayer games were highly

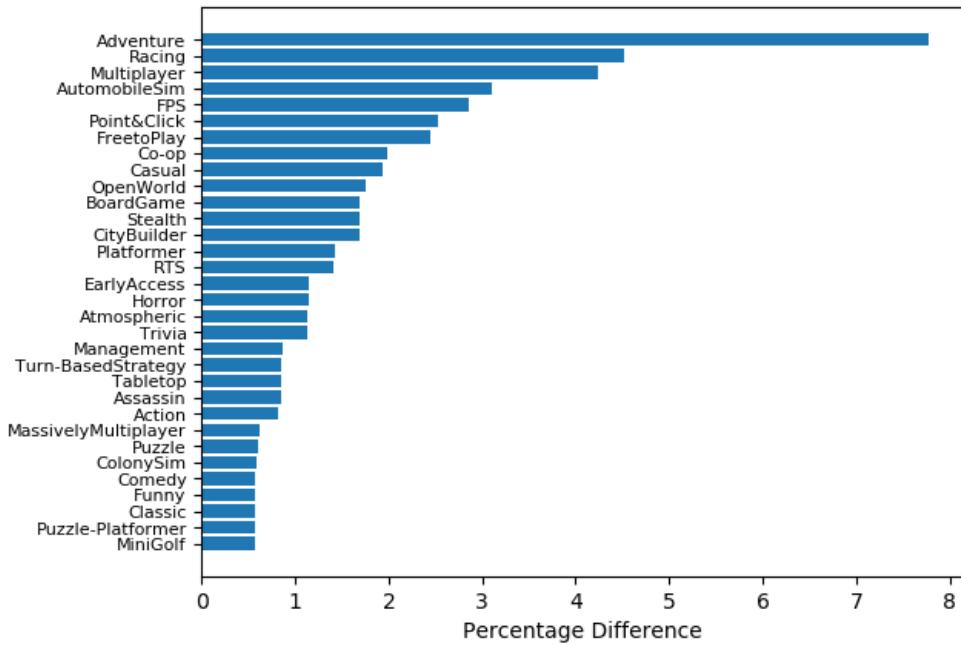


Figure 7.12: Prominent tags of the highly popular games; For each tag the percentage difference between the highly popular games class and not highly popular games class is presented. Only the tags in which the difference is higher than 0.5% are presented in the chart

popular during the onset of the pandemic, which must be because it allows players to interact with others through the game when they are physically separate. First Person Shooter and Point & Click appear next in the list, where Point & Click is a type of adventure game. Free-to-Play tag is also among the prominent tags of highly popular games during the onset of the pandemic indicating a preference for free games. Casual, Open World, Stealth and City Builder games also appear to be some common tags of the highly popular games. Moreover, Boardgame, Trivia, and Tabletop appear to be some of the other interesting tags in the highly popular games class. Some of the games with these tags in this class are *Monopoly Plus*, *Tabletop Simulator*, *UNO*, *Business Tour - Online Multiplayer Board Game* and *The Jackbox Party Pack*, *The Jackbox Party Pack 2 to 4*. *The Jackbox party pack* series contains various trivia type games and the *Tabletop simulator* contains simulations of various tabletop games. It can be seen that most of these games are digital versions of the common simple board games, card games and tabletop games people play when they are physically together such as UNO, monopoly and chess.

### *7.3. Conclusion*

---

Puzzle-platformer and platformer tags also seem to be common. Platformer games are a type of action games in which the gameplay mostly has jumping and running on platforms. Interestingly, there are games tagged as comedy and funny among the highly popular games indicating the popularity of such games during the pandemic.

Newzoo, a game analytics and market research company, has identified that the genres that had shown the highest growth in player share during December 2019 to March 2020 are Shooters, Gambling games, Deck-building games, Arcade games, Platformers and Battle Royale by analysing the games in their Game Development Solution<sup>1</sup> [163]. However, they mention that Deck-building, Platformers and Shooters have displayed growth not solely due to the COVID-19 situation, but due to the release of several major games of those genres. Moreover, they have identified growth in Adventure, Racing and Simulation racing. Interestingly, our analysis also depicts Adventure, Racing and Automobile Sim among the top tags. Also, the social arcade games, one of their custom genres, have also displayed an increase according to their analysis. As per their definition, social arcade games are a combination of arcade, puzzle and online multiplayer genres and contain easy to learn or digitized versions of games players already know [163]. The tags that closest match with this definition in Figure 7.12 are Multiplayer, Puzzle games, Board games, Tabletop and Trivia. It is interesting, that the results observed in our study also has some similarities with their outcomes even though they have used a customized genre taxonomy and a different dataset. It depicts that, irrespective of the dataset, the tags of games that are popular during the onset of the pandemic remain quite similar.

## **7.3 Conclusion**

This chapter presented a study conducted to investigate the changes that occurred in digital games during the initial phase of the COVID-19 pandemic with respect to player population. The study not only generated insights about player population changes but also explored how classification models can be used to predict

---

<sup>1</sup><https://newzoo.com/solutions/game-development/>

### *7.3. Conclusion*

---

games that become highly popular during the onset of the pandemic based on the population changes.

First, an empirical analysis was conducted using 500 popular Steam games to analyse population changes that occurred during the onset of the pandemic. It was identified that there is a high correlation between the aggregate daily player population of all games and the global COVID-19 cases since the 22nd of January 2020. Also, a 33% aggregate player population increase was observed after the 16th of March 2020, which was the same date social distancing and stay at home advice was announced in the United States. Moreover, it was identified that the mean player population of games during the onset of COVID-19 period of 16th March - 16th April 2020 was significantly higher than the mean player population during the same period in the previous year and also higher than the player population during popular Steam Winter and Summer sale events. All the outcomes of the analysis indicate that more and more people have turned to digital games during the onset of the pandemic. It is to be anticipated as people had to stay at home and accessing other outside entertainment options such as movie theatres were all prohibited during the onset of the pandemic. Additionally, it was revealed that out of the games that had displayed weekly seasonality during the pre-pandemic period last year (16th March - 16th April 2019), 76% of games did not display weekly player population patterns during the onset of COVID-19 period (16th March - 16th April 2020). This is expected as more people are staying at home and working from home during the pandemic allowing them to have more freedom in playing on any day and at any time they prefer, disrupting the possibility of having the same population patterns repeated every week. Also, daily player population cycles have displayed changes during the initial phase of COVID-19 period in which there were more players during the first half of the day compared to that of pre-COVID19 period. Moreover, it was identified that *The Jackbox Party Pack 3* and *Tabletop Simulator* were among the games that displayed a population increase of more than 300%. Interestingly, these are games suited for social gatherings as they contain various

### *7.3. Conclusion*

---

casual and fun games such as tabletop games and quizzes. Moreover, the *Plague Inc: Evolved* game which has a storyline relevant to the COVID-19 pandemic had also displayed a 245% population increase. Thus, the study also revealed the variety of game preferences people displayed during the onset of the pandemic. Overall, the empirical analysis revealed that people's interest in digital games has increased during the onset of the pandemic.

The study also investigated how machine learning classification models can be utilized to predict games that become highly popular during the onset of the pandemic based on game-related features, so that game companies can be better prepared for an increased demand. An extended dataset of 1963 Steam games was used. Games were labelled as highly popular if they were among the top 20% of games with the highest percentage of population increase during the onset of the pandemic. The 20% of games with the lowest percentage of population increase were labelled as not highly popular games. Support Vector Machine, Decision tree and tree ensembles, namely, bagging, boosting and random forest were used as classification models. Evaluation of the models revealed that the games that become highly popular during the onset of the pandemic can be predicted better by the Random Forest model compared to other models, however, with an accuracy of 0.69. Additionally, analysis of the tags of the highly popular games revealed that Adventure, Racing and Multiplayer games are the most popular games during the early pandemic. Also, Point & click, Free to Play, Casual, Boardgames and Platformer are some of the other tags of the highly popular games.

The outcomes of this study can be used by game developers to learn what to expect with respect to player population and game preferences during similar pandemics. Specifically, since it was identified that the player population of games highly increased during the early pandemic period, game developers need to be prepared to expect high demand for their games during such periods. It should be anticipated that the demand will be even higher than the demand observed during Steam winter sale period during December. Moreover, since it was identified

### *7.3. Conclusion*

---

that games tagged as Multiplayer, Adventure, First Person Shooter, Free to Play, Casual, and Boardgames were highly popular during the early pandemic period, game developers whose games belong to such game categories can expect higher demand in similar future crises periods. Also, game developers can consider what games became highly popular during the early period of the COVID-19 pandemic as identified by the popular tags, if they consider releasing or promoting games during future pandemics.

The study conducted in this chapter was focused on addressing the research question “How does player population of games fluctuate during pandemics?”. This was addressed by revealing the player population changes that can be observed during the pandemic such as population increases and changes in daily and weekly population patterns. Also, to address the research question, investigations were conducted to explore the possibility of predicting games that become highly popular based on population changes during the early pandemic. It was identified that games that become highly popular can be predicted with at most 0.69 accuracy using game-related features in the prediction model. Furthermore, tags of the games that become highly popular were also identified. One limitation of the study is that only game-related features were considered in generating the prediction models. However, since the pandemic is a highly unprecedented event it is quite likely that not only the game-related features but also other circumstantial factors might have an impact on game selection choices during this period. For instance, what country most players of the game belong to, the severity of the pandemic in that country, and whether the game is family-friendly to play if stay at home rules are imposed. Thus, it might be possible to further improve the performance of classification models if access to such information is available. A direct future work arising from this study is to investigate the player population fluctuations of games beyond the early pandemic period. Such an investigation would be helpful to understand if the patterns observed during the early pandemic period continues throughout the rest of the pandemic. For instance, if population increase observed during early pandemic further increases or returns

### *7.3. Conclusion*

---

back to normal.

This chapter presented insights on player population changes observed in digital games during the early period of the COVID-19 pandemic, which is a world crisis event. The next chapter provides the conclusion of the thesis, summarizing the work conducted and the results obtained.

# **Chapter 8**

## **Conclusion**

This chapter provides the conclusion of the thesis by demonstrating how the research questions of the thesis were addressed and the implications of the findings, presenting the limitations of the research and the future work arising from the thesis.

### **8.1 General Discussion**

This thesis presented a study of player population changes in digital games considering the player population changes in the presence of temporal and event related external factors. The number of players in games changes frequently due to various external factors. Studying the changes of player population size of games in the presence of external factors are important to enhance the understanding of games with respect to player behaviour.

Research literature in the domain of digital games and game data analytics reports studies conducted to enhance the understanding of the behaviour of game players. However, most game studies are based on a single or a few games limiting the applicability of generated insights about game players to the wider game domain, few exceptions being [9] [32] [19]. Furthermore, there still exist various game data analytic related research problems that are not thoroughly investigated. One such less explored area is player population fluctuations of games in the presence of external factors. Although player population changes in games are monitored

### *8.1. General Discussion*

---

and analysed using metrics such as daily active users and peak concurrent users, insights generated from such per-game level analysis are not made widely available. Moreover, since such per-game level analysis are limited to single games, it limits the applicability of the outcomes across other games. Hence there is a lack of knowledge regarding common player population changing patterns in games, especially in the presence of external factors. However, analysis of player population changes at a many-game level is important for game developers to become aware of the current digital game industry trends based on population, especially for indie developers who do not have their data to conduct investigations [34] and to learn common player population changing patterns in various games in the presence of various factors and apply that knowledge appropriately to their games. Hence, this thesis studied player population fluctuations of games in the presence of temporal and event related external factors. Namely, time of the day, day of the week, time since game release, sale events and global pandemic event. The thesis attempts to generate predictions regarding player population changes and to identify the common player population changing patterns displayed by games in the presence of the mentioned external factors, to provide insights and suggestions for game developers and game-related stakeholders.

The main research question of this thesis was “**How does player population of games change in the presence of various external factors?**”. The external factors considered were temporal factors, namely, time of the day, day of the week and time since the release of the game and event related factors, namely, sale events and the onset of the COVID-19 pandemic. Several sub research questions were formulated to investigate the main research question. This section describes how the findings of this thesis address these sub research questions and provide implications.

- How does player population of games fluctuate during a day and a week?

The study presented in Chapter 4 was conducted to address this research question. Using a comprehensive dataset of 1963 popular Steam games, it was

identified that in a majority of games player population changing pattern during a day and a week are recurring. Specifically, 68% of games display recurring daily player population changing patterns and 77% of games display recurring weekly patterns. It was identified that the average daily population pattern depicts a 24 hour cycle in which the population increases for 12 hours and decreases for another 12 hours within a day. Moreover, it was identified that there are nine archetypal weekly player population patterns games display. Whilst the most common pattern displayed by 50% of the games represents an increased player population from Friday to Sunday, several other patterns were also identified. Among them, one weekly pattern displayed nearly the same number of players across all seven days of the week, some patterns had the highest number of players within the week on Sundays while some other patterns had the highest number of players on Saturdays. Further, the features, namely, tags, age requirement and population size, corresponding to the games displaying each pattern were identified.

Several studies in the literature have also explored daily and weekly game playing patterns of players based on player counts and game session information [17] [26] [82] [32]. Most of these studies are focused on patterns in individual games such as *World of Warcraft* [17] [26] and *Everquest II* [82]. The study of Chambers et al. [32], is based on multiple games, however, it is also limited to few popular games. Hence, the applicability of the outcomes to other games seems constrained. However, the study presented in this thesis used an extensive dataset of 1963 popular games. The outcomes of this study are comparative to the outcomes of previous studies with respect to the finding that a majority of games display daily and weekly population patterns. However, the outcomes also indicate that there are some games where the number of players changes irregularly without any daily or weekly pattern which has not been recognized in previous studies due to the limited number of games used. Moreover, previous studies that have focused on daily and weekly pat-

terns have not thoroughly investigated the shapes of weekly patterns games display due to the limited number of games used in the studies. Mahmassani et al., [82] have observed a higher number of players during Friday, Saturday and Sunday in the *Everquest II game*. The study presented in this thesis used an extensive dataset of games and identified different shapes of weekly population patterns games display without being limited to a single or a few games. Hence this finding significantly extends the current knowledge about weekly player population patterns of games.

There are several practical implications of the findings related to this research question. Firstly, the revealed weekly population patterns and the characteristics of games that display each pattern, such as the tags and mean population, can assist upcoming game developers to learn how player population of various types of games fluctuate during a week. They can also understand what type of a weekly pattern they can expect from their games based on game tags. Also, game server infrastructure providers can use the findings regarding the weekly patterns displayed by different types of games in their shared game hosting services. For instance, since it was identified that the pattern in which the player population highly increases on Friday, Saturday and Sunday where Saturday is the highest is displayed mostly by Action and VR games with a mean population of around 126, the hosting providers can consider not placing such games in the same shared server to minimize server outages due to possible high demands on Saturday. Moreover, when running promotional campaigns targeted at different types of games, marketers can use the identified weekly patterns to learn when to expect more players within the week and run the campaigns on such days to reach more players.

- **How does player population of games fluctuate during the first year and first three years after game release displaying life cycle shape approximations?**

### *8.1. General Discussion*

---

The study presented in Chapter 5 addressed this research question. It was identified from the study that there are four archetypal life cycle shapes that games display during the first year after game release and four archetypal life cycle shapes games display during the first three years after game release. Also, the life cycle shapes during the first year and first three years are similar in shape and each shape during the first three years are a possible extension of the first year shape. The most common pattern displayed by a majority of games, specifically 74.2% of games, indicated that player population decreases rapidly during the first few months after release and then the decline in player population slows down. The other life cycle shapes during the first year indicated a steady decrease of population (12% of games), an increase of population (9.8% of games), and a constantly high population that quickly drops at the end of the first year (3.6% of games). The characteristics of games that display these shapes were also explored to generate archetypal life cycle shapes of games.

Currently, there are no studies in the literature we are aware of that have explored life cycle shapes of games based on the player population changes. The closest relevant work conducted on this endeavour is the study by Sifa et al., [9] that have identified four prototypical total playtime profiles of games. The profiles depict the frequency distribution of the total number of hours of playtime. For instance, one identified profile is a short playtime profile where the total playtime is 1 hour for most of the players and less than 10% of players exist that have playtimes longer than 3 hours. The identified playtime profiles are mostly indicative of retention patterns rather than life cycle patterns. Also, the study by Cook has focused on game genre life cycles based on the number of games released from each genre [2] which are different from the life cycles of games based on player population revealed in this study. Hence, the archetypal life cycle shapes revealed in this thesis contribute to the knowledge about player population changes during the first year and first three years after

game release providing approximations regarding life cycle shapes of a game.

The findings provide several implications for game developers and game-related stakeholders. The identified life cycle shapes provide an indication regarding the longevity that can be expected from games. Specifically, since it was observed that in a majority of games the player population starts decreasing soon after release, it can be understood that in most games the popularity that the game received soon after release does not survive in the long run at the same level. However, there are some games in which the player population keeps on increasing indicating that some games have higher longevity. Game developers, especially, upcoming game developers can use the findings of this thesis to become aware of these life cycle shapes and the characteristics of games displaying each life cycle shape. Based on the characteristics game developers can determine what games are more likely to display each life cycle shape. For instance, games that display a growth pattern are mostly released during November, December, February and March where December is the Steam winter sales month. Games that display a fast population decrease after release have been mostly released during August, October and November and least released in June, July, December and January. The popular Steam summer sales and winter sales are also held during June, July, December and January months indicating the games that display a fast population decrease pattern are released avoiding Steam winter and summer sale periods. Game developers can use this insight to select a month to release their games considering if they prefer high popularity soon after release that decreases afterwards or long term growth of population. However, it should be noted that the investigated game characteristics such as release month and tags only provide some indication regarding the life cycle shape of the game. Other features such as the storyline of the game, promotion strategies could also contribute to the longevity of the game which is not explored in this study. Furthermore, since it was identified that the life cycle shape in which player population stays high during the first

year and drops at the end of the year is displayed by mostly sports games, game developers, especially, sports games developers can consider providing annual releases to their games during major sports events to maintain a high population during each year. Additionally, since it was identified that in three out of the four life cycle shapes player population decreases during the first year either soon after release or at the year-end, game developers need to consider applying strategies to grow the player population during the first year for better longevity. It is important as it was identified that the life cycle shape observed during the first three years is a continuation of the first year shape.

- **How accurately can we forecast player population of games during sale events?**

The study presented in Chapter 6 was conducted to address this research question while also using the outcomes of Chapter 5. It was identified that the maximum player population during a sale event of a game can be predicted using past population and sale event related information of the considered game and other games with an RMSE of 0.1367. Moreover, it was identified that such sale event specific models outperform the general population prediction approaches which are trained to predict population during non-sale periods using past population, in predicting population during sale events. Specifically, while the prediction error increased by 6-folds when the general population prediction models are used to predict population during sale periods, the sale event specific prediction models halve that error. Additionally, it was revealed that using life cycle shape based similarity of games to generate separate prediction models for clusters of games does not enhance the accuracy of predicting population during sale events.

Prior studies in the literature have focused on forecasting sales of games [91] [93] [94]. Even among those studies, only one study has also used sale event related information in the forecasting process [94]. But, that study is also lim-

ited to two games. However, not much attention has been given to forecasting population during sale events in the existing literature.

The study conducted in thesis identified that accurately predicting the maximum player population during sale events is challenging compared to predicting the maximum population during non-sale periods. However, since it was identified that by using the past population and sale event information of all games (using a Multi Layer Perceptron model), the maximum player population during a sale event can be predicted more accurately than using general prediction models trained to predict population during non-sale periods based on past population alone, game developers can consider using sale event related information of other games as well in their prediction approaches. In fact, third-party game service providers can consider creating an all-games-together sale event specific prediction model to predict the maximum population during sale events which can be used to provide prediction services to game developers, especially, individual game developers who do not own multiple games to generate such models. Also, since it was identified that the population prior to the sale event is the most important feature compared to the other features used in the all-games-together prediction approach, game companies can consider giving higher importance to the past population in generating prediction models that forecast population during sale events. Furthermore, it was identified that for per-game prediction approaches, more accurate predictions on population during sale events can be generated by solely using the past population history of the game in a Nonlinear Autoregressive model rather than using both population and price history in a Nonlinear Autoregressive Exogenous model. Hence, game developers who only have data regarding their games can consider generating such per-game forecasting models that solely use the past population for sale event population prediction, especially, when there is insufficient past sale event history. Additionally, the capability to predict the expected number of players during a sale event can aid game companies in

### *8.1. General Discussion*

---

understanding the demand for games during sale events and to schedule sale events effectively to attract more players based on the predictions.

- **How does player population of games fluctuate during pandemics?**

The study presented in Chapter 7 addressed this research question. It was identified that the player population of digital games has significantly increased during the onset of the COVID-19 pandemic. Specifically, a 33% of population increase was observed after the 16th of March 2020, the date the US announced social distancing advice. Moreover, the player population of games during the early period of the pandemic was higher than the population during the same period in the previous year and even during popular Steam summer and winter sales. There were even changes in weekly player population patterns in games where games that had previously displayed weekly seasonality failed to display weekly patterns during the early period of the pandemic. These findings imply that more people have turned to video games during the early days of the pandemic. Furthermore, changes in weekly patterns imply that people had more freedom to play on any day and at any time preferred. This is likely attributable to the fact that people were forced to stay at home, even during the weekdays. Moreover, it was identified that a random forest classification model can predict games that become highly popular during the onset of the pandemic; however, with only a 69% of accuracy. The performance of the random forest model was better than the decision tree, tree bagging, tree boosting, and SVM models. The classification models used game related characteristics such as release date, tags, price and mean player population prior to the pandemic as features of the model. Since this is an initial step towards predicting games that become popular during such a unique global pandemic, the prediction performance results are promising, yet there is considerable room for improvement.

Since the COVID-19 pandemic is a very recent event there are only a few

### *8.1. General Discussion*

---

studies that have investigated the changes that have occurred in the gaming industry. These studies have identified an increase in sales of video games [35], the time spent playing video games [36], viewership in Twitch [98] and viewership in Youtube Gaming [99]. However, the study conducted in this thesis not only investigated the changes in player population size during the onset of the pandemic but also investigated the changes in daily and weekly population patterns and generated models to predict games that become popular during the pandemic onset.

The outcomes of the study conducted to address this research question implies that game developers should expect a higher demand for their games during pandemic onsets. Especially, games tagged as Multiplayer, Adventure, First Person Shooter, Free to Play, Casual, and Boardgames can anticipate a higher demand during similar future crises as it was observed that the games that became highly popular during the COVID-19 pandemic onset mostly had those tags. Hence, game developers can use the insights regarding the tags of games that become highly popular during the pandemic to assist in selecting what games to release, promote, or to expect a higher demand during similar pandemics or world crisis.

The research conducted in this thesis used *Gameset1* and *Gameset2* introduced in Chapter 3. *Gameset1* contained player population data collected in 5-minutes and 1-hour intervals. *Gameset2* contained daily player population data. As mentioned in Chapter 3, player population data of *Gameset1* are made publicly available for anyone to use. To the best of our knowledge currently, no player population dataset collected in such high frequency is publicly available. Collecting data at such high frequencies can be a burden to both the collector and the data provider due to the storage requirements and the high number of requests that have to be made to collect data. However, based on the studies conducted in this thesis, it can be understood that data collected in such frequent intervals are useful for some situations while it is not required for some other situations. To investigate frequent player population

fluctuations data collected in shorter intervals are useful. For instance, both hourly and per 5-minute player population data were initially used to investigate daily and weekly patterns. However, 5-minute interval data were also converted to hourly data later to reveal daily and weekly patterns as it was observed that hourly data is sufficient for that purpose. To investigate long-term player population changes daily player population data is sufficient. For instance, life cycle shape extraction was conducted using daily player population data. However, to calculate the daily average or daily maximum population which is used to create a daily population dataset, it is still necessary to collect data at a high frequency such as hourly. Hence, it can be understood that at least hourly population data is required to investigate player population fluctuations in several situations.

## 8.2 Limitations

The research presented in this thesis has several limitations. One key limitation is the limited number of game-related features and other external features considered in characterizing the games that display each archetypal weekly pattern and life cycle pattern. Specifically, when investigating the characteristics of games that display the identified weekly population patterns, only the game tags, age requirements and mean population were explored. Also, when investigating the characteristics of games corresponding to each player population-based life cycle archetypes, only game-related features such as tags, price and release month were considered. However, features related to strategies game companies use to attract players such as pre-release marketing budget and the number of promotional events during the first year were not considered due to the unavailability of such data. The limitations related to the limited number of features used are applicable to the sale event population forecasting model generation as well. Some features that are related to sale events such as the type of the sale event (developer-scheduled or Steam-scheduled) and whether the game appeared in the first page of the sale event or on a later page were not used in the all-games-together model due to the unavailability of

data regarding that. Moreover, when generating sale event-related population prediction models in a per-game approach only non-linear autoregressive (NAR) and non-linear autoregressive exogenous (NARX) models were used. Although some other time series forecasting approaches, such as Autoregressive Integrated Moving Average (ARIMA) and Holt Winter’s Exponential Smoothing (HWES) exist, those were not used in the study due to the time and effort required to generate such forecasting models for each game. Hence, the outcomes of the per-game models are limited to the NAR and NARX models.

### **8.3 Future Work**

This section presents several future research directions arising from the work presented in this thesis.

- A study investigating the games that did not display recurring daily and weekly player population patterns.

The study conducted in this thesis revealed that a majority of games display daily (68% of games) and weekly patterns (77% of games) implying that there are some games that do not display recurring daily and weekly patterns. It would be interesting to further investigate the type of these games and what causes those games to differ from the common behaviour displayed by other games. While having a lower number of players could be a probable cause for the lack of regularity in player population changes, there could be other reasons as well which would be worth investigating.

- Explore game life cycle shapes beyond three years.

This thesis revealed four archetypes of game life cycle shapes based on player population fluctuations during the first year and first three years after game

### *8.3. Future Work*

---

release. However, there are games that successfully live beyond three years. Hence it would be quite useful to understand what life cycle shapes games that enjoy viable player populations that endure display. This would however require gathering player population data of games for a longer period of time.

- Explore the possibility of enhancing the process of predicting maximum player population during sale events using external information.

The prediction approach proposed in this thesis used sale event related information and game related information as features. However, there are clearly other factors that influence the player population that can be observed during a sale event of a game. Some of those factors could be whether the sale event was prominently promoted, whether pre-marketing strategies were conducted about the sale event, and whether competitor games would also be conducting sale events at the same time. Hence, collecting such information and exploring whether those could enhance the prediction accuracy would be useful to investigate.

- Investigate the possibility of enhancing the accuracy of predicting games that become popular during the pandemic using information related to the pandemic

The classification models used in this thesis to predict popular games during the pandemic used game related features and past population information of the game as features of the model. However, using external information related to the pandemic, such as social distancing rules imposed in the country where a game that is most commonly played might further aid in predicting the popular games during the pandemic.

- Determine whether the changes in player populations associated with COVID-19 are sustained.

This thesis investigated the player population changes in games during the early period of the pandemic, which is prior to the 16th of April 2020. The study identified that player population has significantly increased during the onset of the pandemic and revealed changes observed in the daily and weekly population patterns. However, since the pandemic is still continuing, it is interesting to investigate whether the changes observed during the early pandemic period are sustained. Especially, whether the popularity of games continues to increase and if and when the population increase obtained during the early pandemic period starts to decrease are some aspects that can be explored.

- Exploring how the changes in one game are tied to changes in the population of another.

Every year thousands of digital games are released to the public. Hence, the game players have the opportunity to select games from a wide range of options. Thus, the players can leave a game permanently or migrate to another game. Player population of games could change due to such migrations. Apart from the release of new games, these migrations could be influenced by the changes introduced in existing games, such as updates and promotional activities. Hence, each game could have various types of influence on one another impacting their popularity in the competitive digital game industry. Hence, it is interesting to explore how changes in one game such as updates, promotional activities, and price changes can introduce changes in the population of other games. Furthermore, the release of a new game could also introduce changes in the population of existing games. Hence, exploring how the changes in one game could introduce changes in the popularity of other games is useful to model player migrations across games.

- Generating a prediction model to forecast player population of a game whilst

considering the influence from various temporal and event related external factors.

This thesis explored how player populations change in the presence of various external factors such as time of the day, day of the week, age of the game, sale events and pandemic events. The next major research pathway that can be explored utilizing the insights generated from this thesis is to model player population changes of a game incorporating such external factors in order to accurately forecast the future population of a game based on its history. Such a model would be quite useful to better plan required resources to serve the expected demand and to plan marketing and sales strategies at different stages of the game to strengthen the longevity of the game.

## **8.4 Concluding Remarks**

The digital games industry is massively popular and generates multi-billions in revenue. While many games are frequently released, the player numbers in particular games also fluctuate frequently due to various factors. This thesis investigated the player population fluctuations of games in the presence of temporal and event related external factors, namely, time of the day, day of the week, age of the game, sale events and the onset of COVID-19 pandemic. The thesis presented insights regarding the daily and weekly player population patterns and life cycle shapes of games. Furthermore, several models that aid in revealing population patterns and generating predictions related to games and player population were presented. This included predicting player population during sale events and predicting popular games during the onset of the COVID-19 pandemic. It is expected that the outcomes of this thesis would be helpful for stakeholders in the digital game industry, such as game companies and indie game developers to be more aware of the player population changes in various types of games. Moreover, it is also expected that this

#### *8.4. Concluding Remarks*

---

thesis would serve as a stepping stone for further research related to the exploration of the changes in game player populations in the presence of various external factors. Thus, ultimately providing a better experience to game players. Additionally, the frameworks of analysis and methodologies employed in this thesis are not only limited to the digital game industry but also applicable to other fields, as influence from external factors are not restricted to the digital game industry. Ultimately, the contributions of this thesis pave the way towards further advancement in thriving digital game industry everyone enjoys.

# Appendix A

## Analysis of Game Characteristics of the Life Cycle Archetypes

### A.1 Introduction

In Chapter 5, the life cycle shape archetypes games display during the first year after game release  $1Yclust$  and during the first three years after game release  $3Yclust$  were identified. A summary of the characteristics of games displaying each of those archetypes was presented in that chapter. This Appendix is a comprehensive version of the summary provided in Chapter 5. In this appendix, first, the analysis of characteristics of games displaying  $1Yclust$  archetypes is presented. Then the analysis of characteristics of games displaying  $3Yclust$  archetypes is presented.

### A.2 Characteristics of games displaying archetypes of first year ( $1Yclust$ )

In this section tags, release year, release month, mean population, publishers, developers, price and reviews of games displaying  $1Yclust$  archetypes are analysed.

#### Tags

Analysing the tags of games in each cluster it was observed that some tags are

common across all clusters while some tags are unique to only a certain cluster. The tags that are common among all 4 clusters are presented in Figure A.1 along with the percentage of games they represent in each cluster for ease of visualization. In addition to that, some tags such as Free-to-Play, Indie, First-Person, Atmospheric, Great Soundtrack, FPS, Massively Multiplayer, Online Co-Op, Story Rich and Sandbox are common among 2 or 3 clusters only as depicted in Figure A.2. Apart from these, there are tags that are unique to only a single cluster. *1YfastDec-slowInc Cluster* consist of Fantasy, Third Person, Difficult, Funny tags. While Realistic is unique to *1YslowDec-inc Cluster*, Survival is unique to *1Ydecrease Cluster*. Also, *1Ydec-inc-fastDec Cluster* contains Tactical, Management, PvP, Sports, Moddable, Soccer, Football, Competitive, War tags as unique tags.

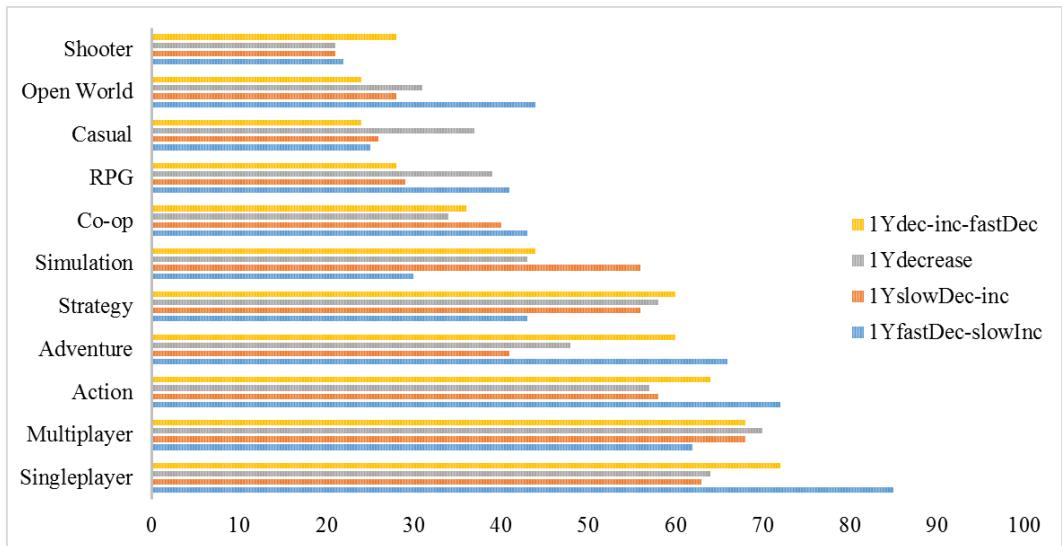


Figure A.1: Common tags among clusters; first year archetypes

While some popular tags are more commonly appearing among games, some tags may appear only in some games [164]. Hence, it is interesting to analyse the percentage difference between tags in the dataset and tags in each cluster. This would provide an understanding of the tags in each cluster that appear not solely due to the popularity of the tag. It is presented in Figure A.3, A.4 and Table A.5. *1YfastDec-slowInc Cluster* is the largest cluster containing 74% of the games in the dataset. Hence, the tag distribution of that cluster cannot be heavily different from the tag distribution of the dataset. However, it can be noted that the

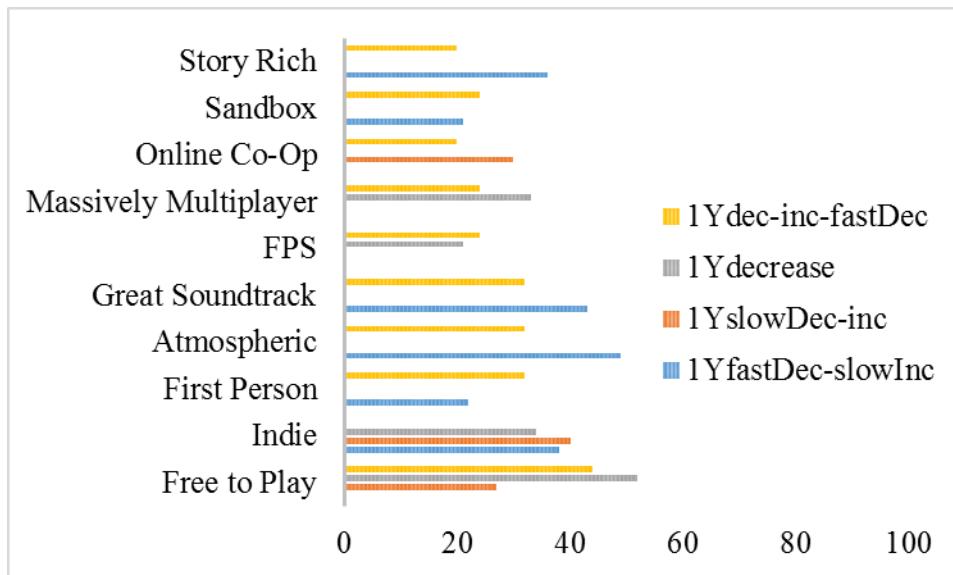


Figure A.2: Partially common tags among clusters; first year archetypes

tag percentage of Singleplayer, Action, Adventure, OpenWorld, Atmospheric, Great Soundtrack, Story Rich, Fantasy, Third Person, Difficult, Funny have a higher tag percentage in *1YfastDec-slowInc Cluster* compared to that of the dataset by 2-5%. When *1YslowDec-inc Cluster* is considered Simulation, Strategy, Online Co-Op, Realistic tags seem to have higher percentages of appearance. *1Ydecrease Cluster* seem to contain more games of Strategy, Simulation, Casual, Free-to-Play, Massively Multiplayer and Survival as they have higher percentages compared to the dataset. *1Ydec-inc-fastDec Cluster* could be associated with Strategy, Simulation, Shooter, Free-to-Play, First-Person, FPS, Massively Multiplayer, Sandbox, Tactical, Management, PvP, Sports, Moddable, Soccer, Football, Competitive, War.

Based on the analysis it seems that each cluster is associated with a multiple number of tags instead of a single Tag. A simplified set of tags associated with each cluster is chosen by comparing the tag percentages between clusters, identifying unique tags from clusters, and considering the difference from the dataset.

*1YfastDec-slowInc Cluster*: Action, Adventure, Open World

*1YslowDec-inc Cluster*: Strategy, Simulation

*1Ydecrease Cluster*: Survival, Casual

*1Ydec-inc-fastDec Cluster*: Sports, Football, Shooter

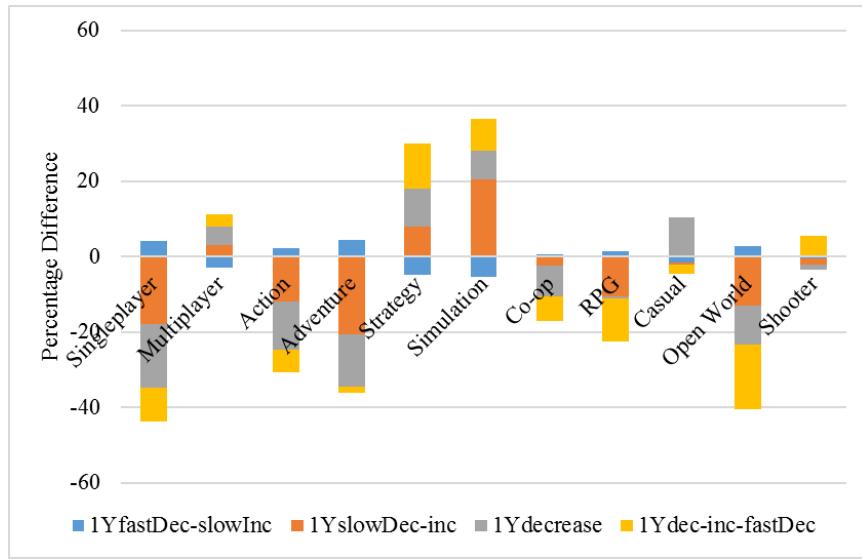


Figure A.3: Common tags and their percentage difference between clusters and dataset ; first year archetypes

### Release Year

Histograms of the release year of games in each cluster are depicted in Figure A.6.

As per Figure A.6, all the games in *1YfastDec-slowInc Cluster* are released during or after 2010. *1YslowDec-inc Cluster* has few games released during 2007 - 2012, but most of the games are released after 2012. Most games in *1Ydecrease Cluster* are released after 2014. The released year of games in *1Ydec-inc-fastDec Cluster* seems to be equally distributed. This also agrees with the patterns observed in *1Ydec-inc-fastDec Cluster* which included some sports games which are released every year. However, the released years' distribution of *1YfastDec-slowInc Cluster*, *1YslowDec-inc Cluster* and *1Ydecrease Cluster* does not seem to indicate any relationship with the pattern the cluster represents.

### Released Month

Release month distribution of games in clusters is presented in Figure A.7. As per Figure A.7, many games in *1Ydecrease Cluster* seem to have been released during June, and September to December. Although the major Steam sale event dates change slightly each year, Summer sale tends to start by the end of June and Halloween, Autumn(Black Friday) and Winter sales tend to take place during October to December. Hence, the release month of games in *1Ydecrease Cluster*

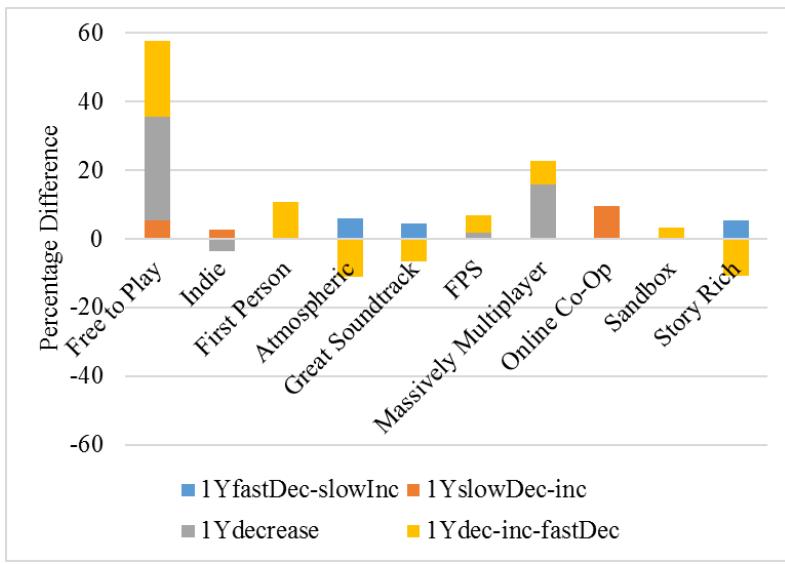


Figure A.4: Partially common tags and their percentage difference between clusters and dataset ; first year archetypes

seem to coincide with the major sale events in Steam. However, this alone can not directly imply the existence of any relationship between Steam sale events and the decreasing population pattern of games in this cluster. However, information about the games that actually participated in the sale events may provide further insights. Also, the rules imposed by Steam on discounting games need to be considered. January, April, June and October seem to be the months the least number of games in *1YslowDec-inc Cluster* were released. However, most games in *1Ydec-inc-fastDec Cluster* have been released in November. The last 4 months of the year which is September to December period seems to be the popular month of release for games in *1Ydec-inc-fastDec Cluster* which coincides with the National Football League (NFL) season.

### Mean Population

The mean population of games during the first year of each cluster is analysed using the box plots in Figure A.8. As per Figure A.8, median of the mean population values of *1Ydec-inc-fastDec Cluster* is 2101. This is the highest among the four clusters. The pattern of *1Ydec-inc-fastDec Cluster* displays that the population initially decreases and increases slightly while staying at a high value and decreases suddenly at the end of the year. Hence, it is understood that *1Ydec-inc-fastDec Cluster* could

Percentage Difference				
Tag	1Yfast Dec- slowInc	1Yslow Dec-inc	1Ydecre ase	1Ydec- inc- fastDec
Fantasy	2.19			
Third Person	3.55			
Difficult	2.12			
Funny	2.47			
Realistic		11.97		
Survival			4.65	
Tactical				19.9
Management				22.16
PvP				19.65
Sports				21.44
Moddable				13.57
Soccer				22.36
Football				21.91
Competitive				13.59
War				7.33

Figure A.5: Unique Tags and their percentage difference between clusters and dataset; first year archetypes

display a high median in mean population compared to other clusters. Furthermore, this also indicates that the *1Ydec-inc-fastDec Cluster* contains popular games. The median value of *1Ydecrease Cluster* is 1247, almost half of the median of *1Ydec-inc-fastDec Cluster*. However, this is higher than that of *1YfastDec-slowInc Cluster* and *1YslowDec-inc Cluster*. Although *1Ydecrease Cluster* displays a decreasing pattern, the decreasing process happens slowly throughout the year. Hence, the mean population of most of the games is not quite low. The median of the mean population values is lowest in *1YslowDec-inc Cluster*, which is 909. *1YfastDec-slowInc Cluster* also has values closer to *1YslowDec-inc Cluster* where the median is 961. This is lower compared to *1Ydecrease Cluster* and *1Ydec-inc-fastDec Cluster*. Games in *1YfastDec-slowInc Cluster* displayed a pattern where population decreases for a while and slowly increases maintaining a low population. As per the 25th percentile, more than 75% of the games that display *1YfastDec-slowInc Cluster* pattern have

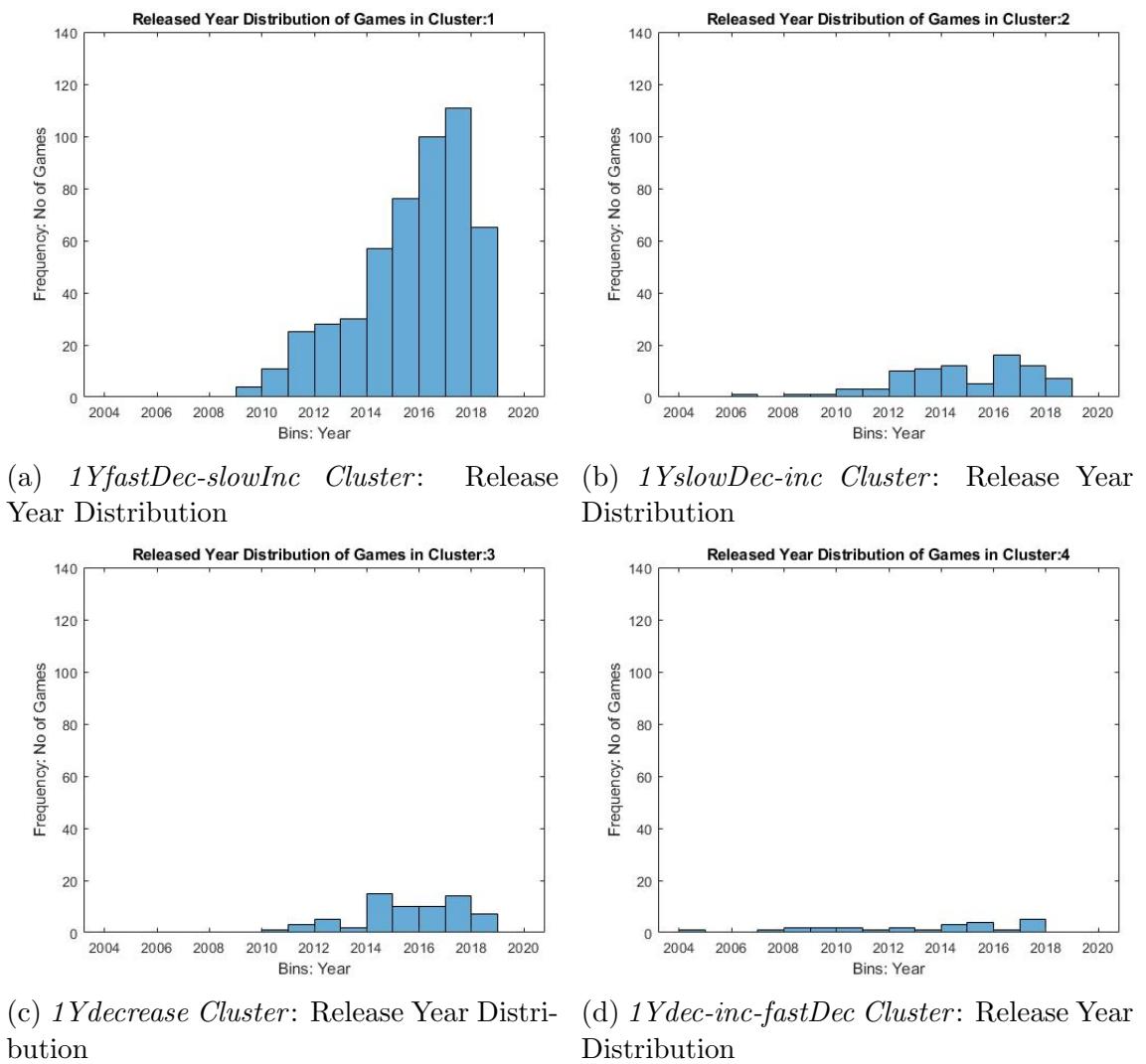


Figure A.6: Release year distribution of games displaying first year archetypes

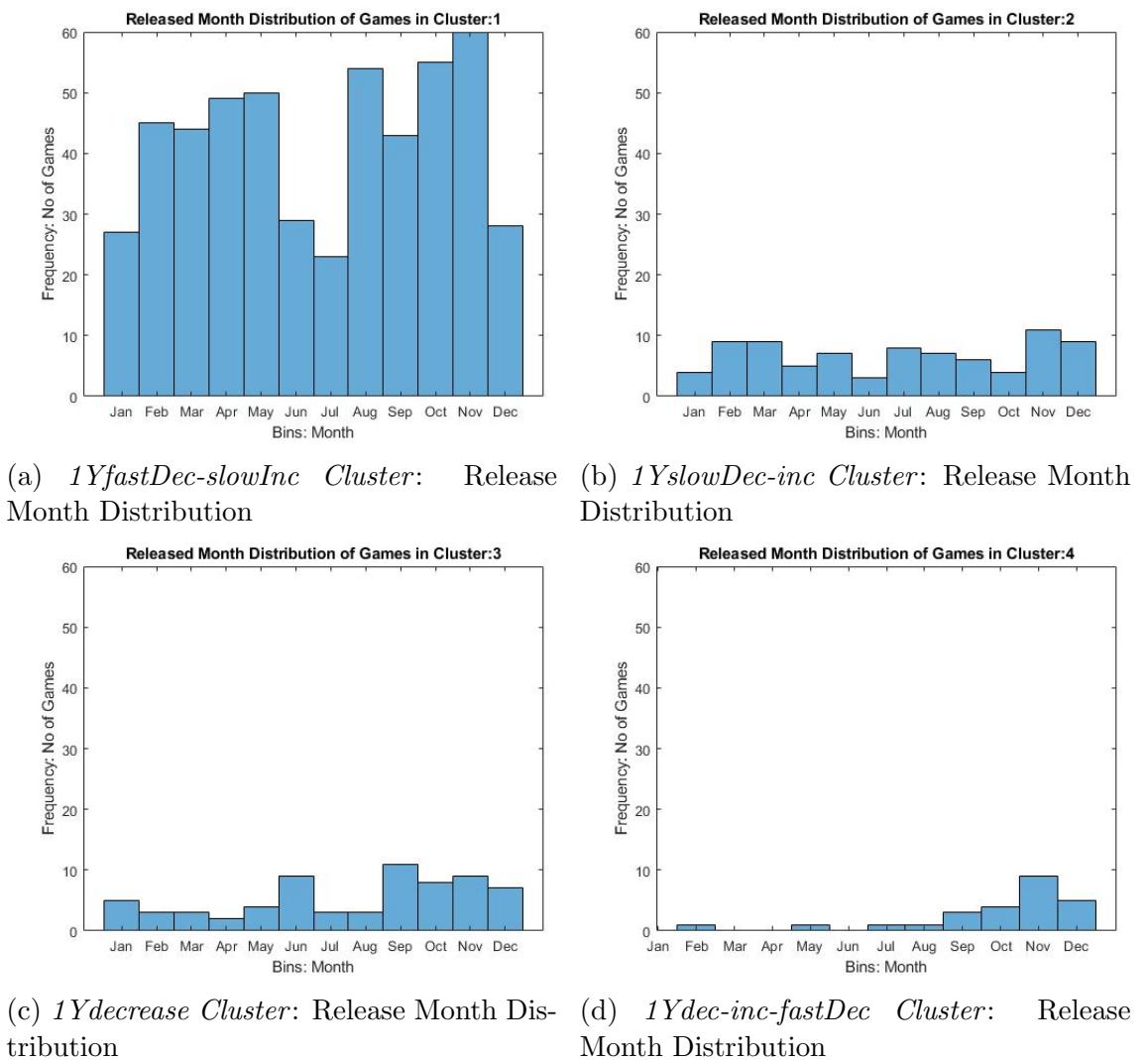


Figure A.7: Release month distribution of games displaying first year archetypes

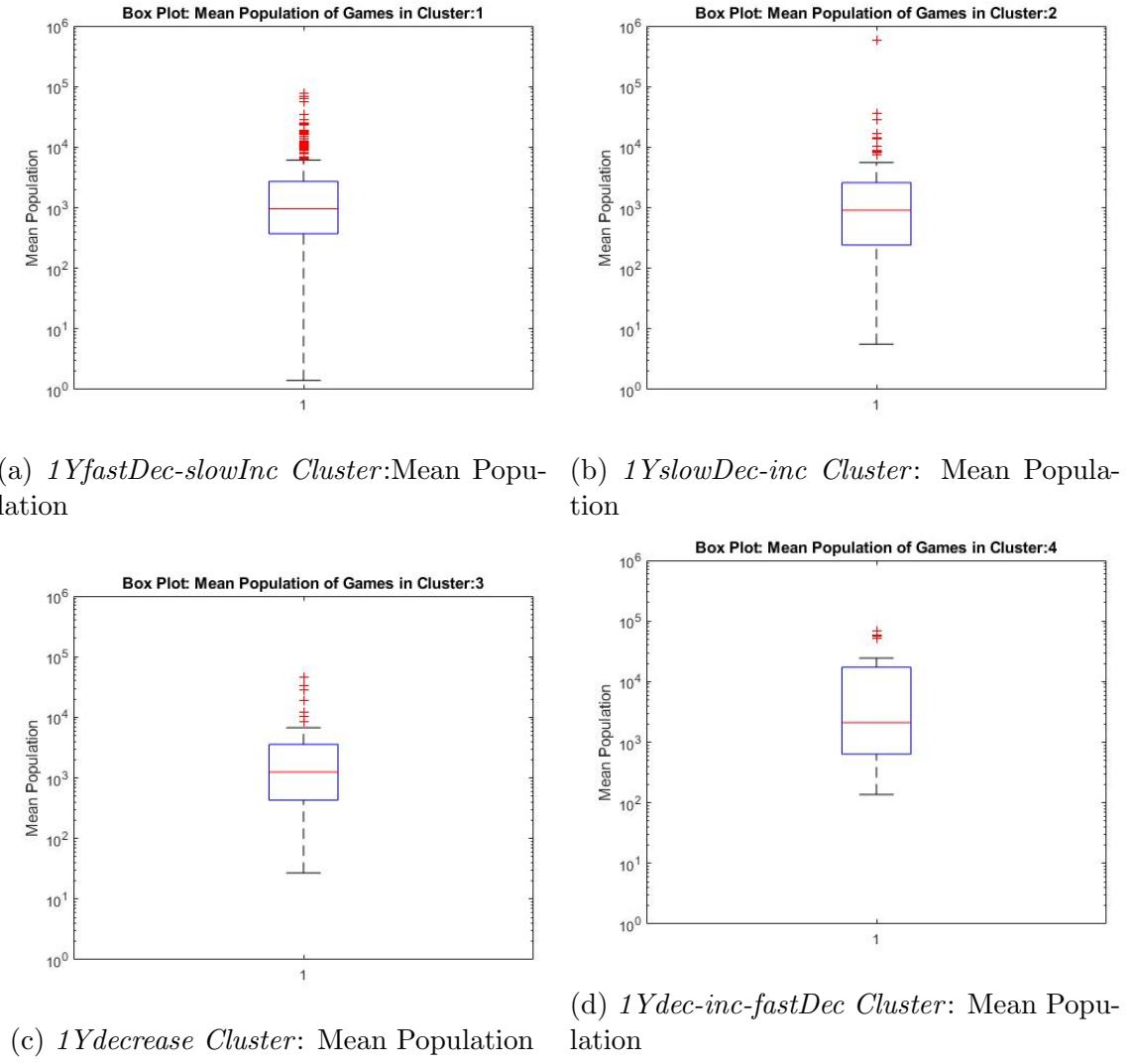


Figure A.8: Mean population statistics of games displaying first year archetypes

a mean population of 373 or more during the first year.

### Publishers

Analysis of publishers revealed that each cluster is associated with a large number of publishers. Considering the number of games in each cluster, the distribution of publishers is too diverse to determine any relationship between the cluster and publishers. For instance, although *1YslowDec-inc Cluster* has 82 games the top publisher of the cluster, *2K* has only published 5 games of the cluster, which represents only 6% of the games. Most of the other publishers have published only 1 game in all clusters. *SEGA* is at the top in *1Ydec-inc-fastDec Cluster* representing 20% of the games. Football Manager games in *1Ydec-inc-fastDec Cluster* are published by *SEGA*. Hence, it can be seen that publisher information cannot be readily used

to determine the types of games associated with each cluster except for *SEGA* for *1Ydec-inc-fastDec Cluster*.

### Developers

Developers in each cluster are too diverse and only 1 or a few games are developed by a given developer in each cluster. Thus, developers can not be used to distinguish between games in each cluster.

### Free-to-Play Games

Free-to-Play game percentages of each cluster are presented in Table A.1. As per Table A.1 *1YfastDec-slowInc Cluster* has a higher percentage of non-Free-to-Play games compared to Free-to-Play games. *1YslowDec-inc Cluster* also has a higher percentage of non-Free-to-Play games. *1YslowDec-inc Cluster* pattern is an increasing population pattern. However, *1Ydecrease Cluster* has more Free-to-Play games than non-Free-to-Play games. *1Ydecrease Cluster* shows a declining population pattern. But the percentage difference is not significant. *1Ydec-inc-fastDec Cluster* seems to have the opposite ratio of *1Ydecrease Cluster*. *1Ydec-inc-fastDec Cluster* contains more non-Free-to-Play games, however with close percentage differences.

Cluster	F2P Percentage	Non-F2P Percentage
<i>1YfastDec-slowInc</i>	16.17	81.65
<i>1YslowDec-inc</i>	29.26	70.73
<i>1Ydecrease</i>	56.71	43.28
<i>1Ydec-inc-fastDec</i>	44	56

Table A.1: Free to Play games percentage of games displaying first year archetypes

### Price of Games

Price distribution of games in each cluster is provided in Figure A.9. *1YfastDec-slowInc Cluster* seems to have a large number of games belonging to the \$10-20 price range. It is similar for *1Ydecrease Cluster* and *1Ydec-inc-fastDec Cluster* as well. *1YslowDec-inc Cluster* has more games in \$0-20 price range.

### Reviews

The distribution of the positive review percentage of games in clusters are depicted in Figure A.10. Interestingly, *1Ydecrease Cluster* seem to have slightly more

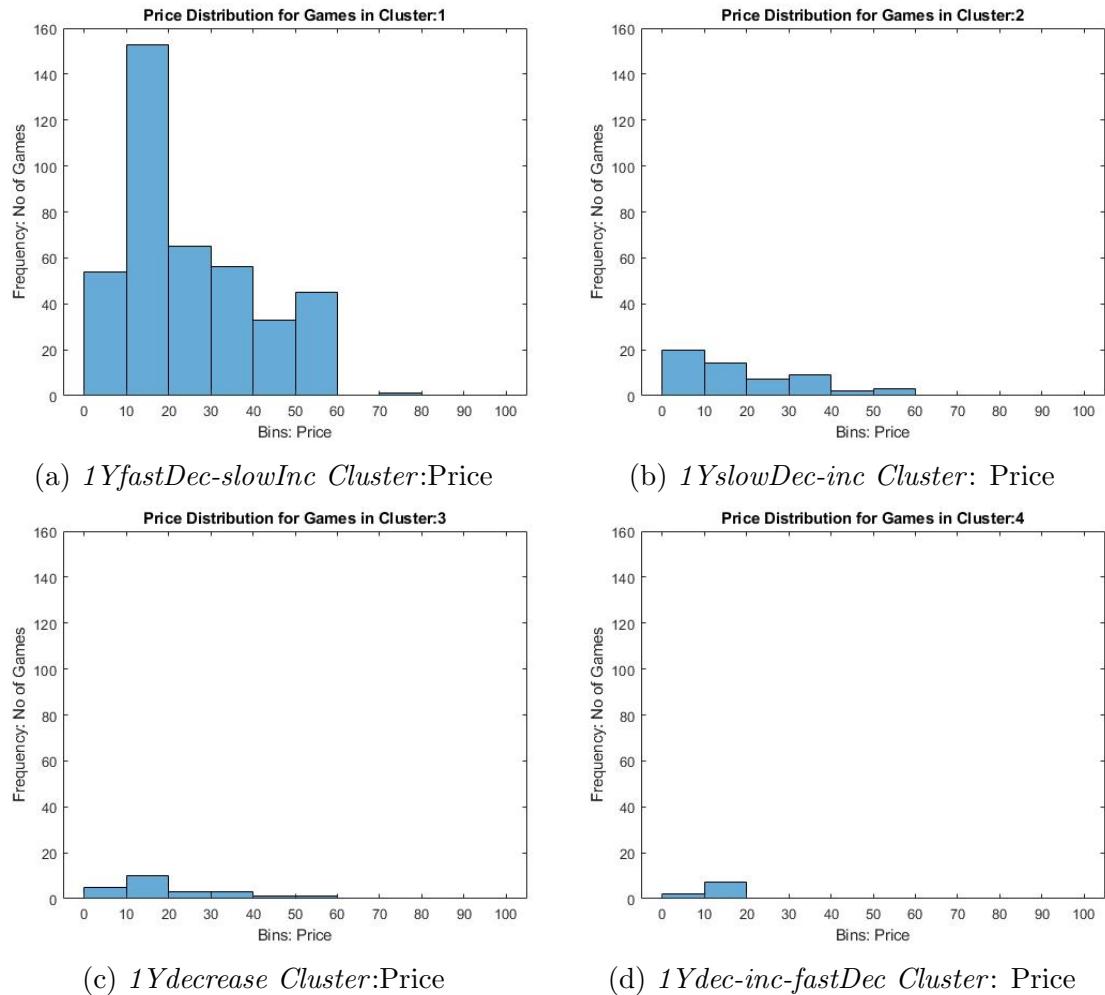


Figure A.9: Price distribution of games displaying first year archetypes

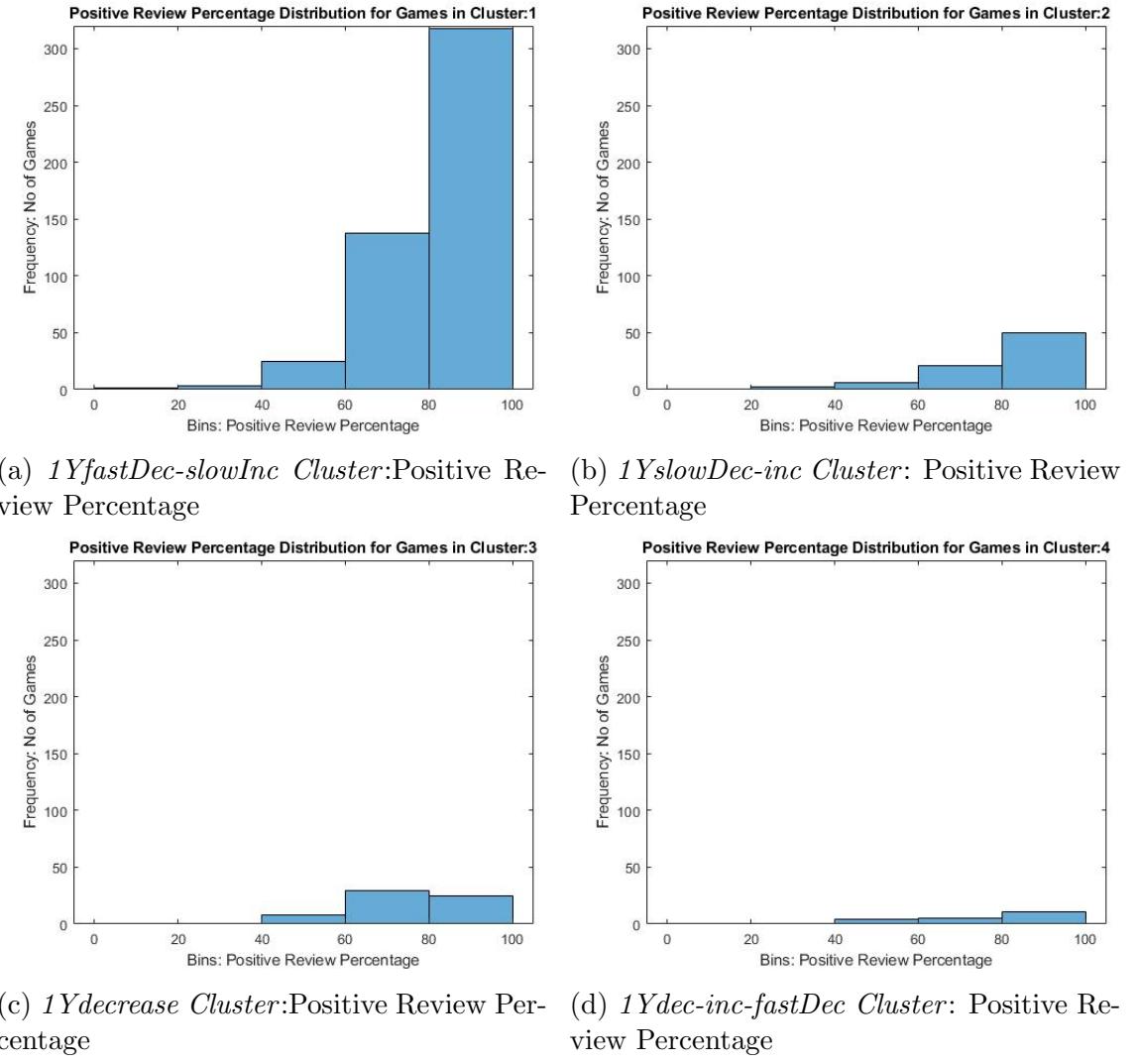


Figure A.10: Positive review percentage distribution of games displaying first year archetypes

games in the 60-80% range compared to the 80-100% positive review percentage range. *1Ydecrease Cluster* showed a decreasing population pattern. *1YslowDec-inc Cluster* seems to have many games in the 80-100% range and it showed an increasing population pattern. *1Ydec-inc-fastDec Cluster* has most games in the 80-100% range compared to the other bins.

## **A.3 Characteristics of games displaying archetypes of first three years ( $3Yclust$ )**

In this section tags, release year, release month, mean population, price and reviews of games displaying  $3Yclust$  archetypes are analysed.

### **Tags**

Tags that were common across all 4 clusters are depicted in Figure A.11 along with the percentage of games those represent. Tags that are partially common are presented in Figure A.12. Moreover, there were tags that are unique to each cluster as well. The unique tags and the percentage difference of common tags between clusters can be used to determine the tags that represent games in each cluster.  $3YfastDec-slowDec$  Cluster seems to contain more games of Single Player, Action, Adventure, RPG, Atmospheric, Open World and Indie tags. Also, the tags that are unique can be named as Story Rich, Fantasy, Third Person, Difficult, Sci-fi, Funny and 2D. When it comes to common tags,  $3Yincrease$  Cluster seems to contain more games of Strategy and Simulation. Moreover, it contains unique tags such as Realistic, Moddable, Tactical, Military, War, Replay Value, Historical and Team based.  $3Ydecrease$  Cluster contains more games belonging to Action, Adventure, RPG, Open World, Massively Multiplayer and Sandbox. Also, unique tags it contains are PvP, MMORPG, Gore, Medieval and Classic.  $3Y3stageDec$  Cluster seems to contain Single Player, Multi Player, Adventure, Strategy, Simulation, Co-op, Casual, Atmospheric and Free-to-Play as common tags. However, a higher percentage of these tags could be appearing due to the low number of games in the cluster as well. Moreover, the unique tags it contains are Sports, Management, Family Friendly, Soccer, Football, Controller, Clicker, 2D, Funny, Addictive, Touch Friendly, Point and Click, Memes, Basketball and Realistic.

Based on the unique tags and tag percentages and games, the tags associated with each cluster can be simplified as follows.

$3YfastDec-slowDec$  Cluster : Action, Adventure, Indie

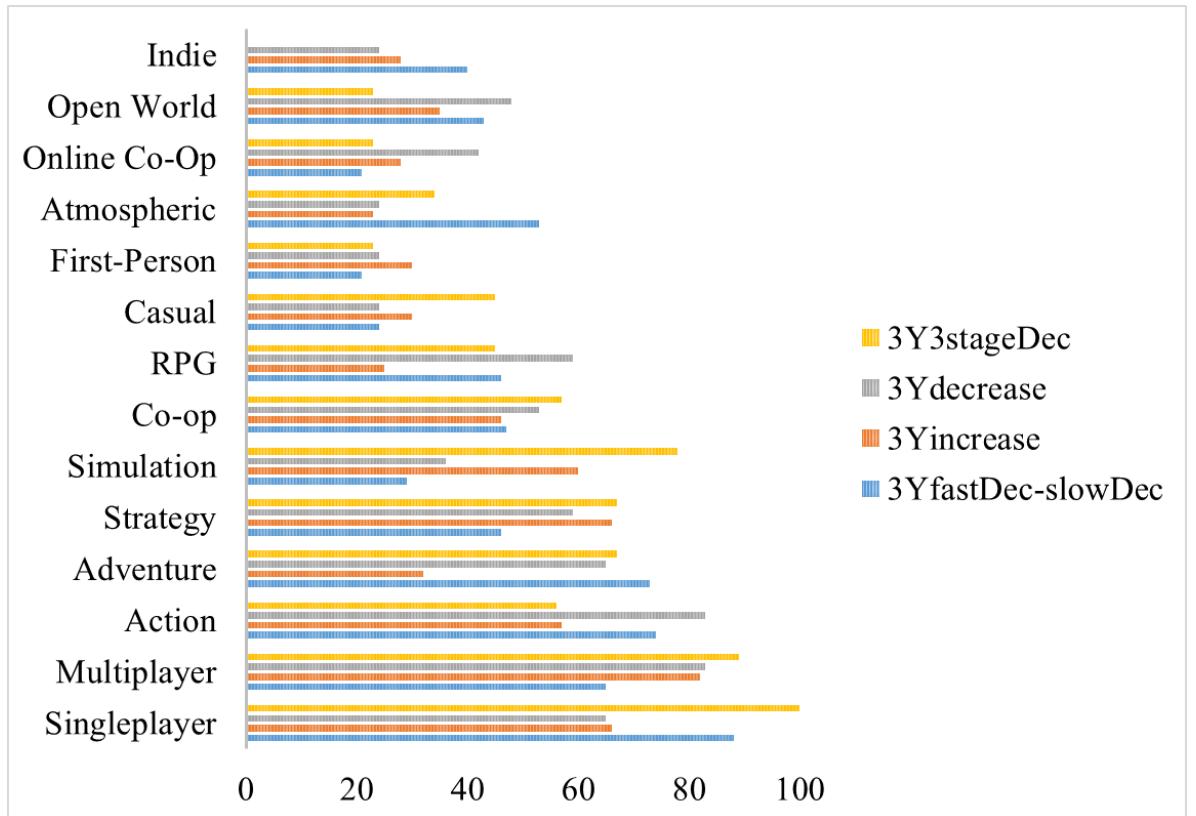


Figure A.11: Common tags among clusters; first three years archetypes

*3Yincrease Cluster:* Simulation, Strategy, War

*3Ydecrease Cluster:* Action, RPG, Open world, Free to Play, Survival

*3Y3stageDec Cluster:* Simulation, Casual, Sports, Football, Basketball

### Release Year

The release year distribution of games in clusters is depicted in Figure A.13.

*3YfastDec-slowDec Cluster* has most games released after 2014 and the majority is released in 2016. *3Yincrease Cluster* also has a higher number of releases after 2014.

*3Ydecrease Cluster* seems to have more games released in 2013 and 2016 however the difference between years is not quite significant. *3Y3stageDec Cluster* has most games released between 2014 and 2016.

### Release Month

Release month distribution of games in clusters is presented in Figure A.14. The least number of games have been released in July and December months in *3YfastDec-slowDec Cluster*. There is a low number of game released in June and January also. However, popular Steam winter and summer sales are held during

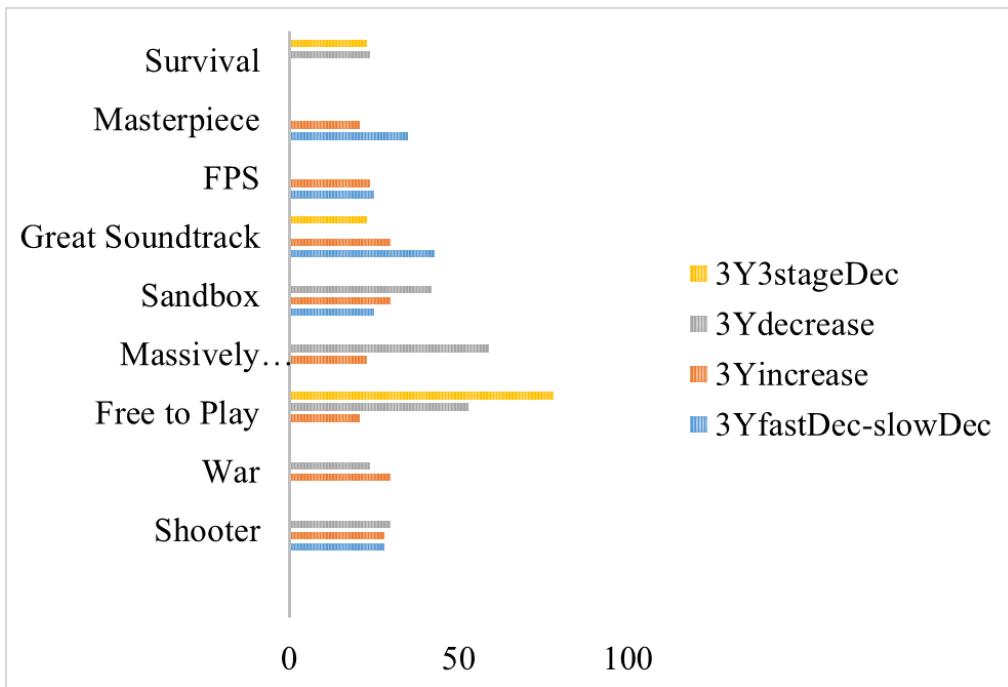


Figure A.12: Partially common tags among clusters; first three years archetypes

these months. This indicates that most games in *3YfastDec-slowDec Cluster* are released outside popular sale event periods. *3Yincrease Cluster* has games released in each of the 12 months of the year. August appears to be the most common followed by May and June. Games in *3Ydecrease Cluster* are released in months from June to December and also during February. However, the most common months are July, November and December. *3Y3stageDec Cluster* contains games mostly released in September and November.

### Mean Population

Statistics related to the mean population of games during the first three years after release are presented in Figure A.15. It can be seen that the median of the mean population is quite similar between *3YfastDec-slowDec Cluster*, *3Yincrease Cluster* and *3Ydecrease Cluster*. However, it is higher in *3Y3stageDec Cluster*.

### Free-to-Play Games

Free-to-Play game percentages are given in Table A.2. It can be seen that *3YfastDec-slowDec Cluster* and *3Yincrease Cluster* has a higher percentage of non-Free-to-Play games. *3Ydecrease Cluster* has a higher percentage of Free-to-Play games while *3Y3stageDec Cluster* has an almost similar percentage of both Free-to-

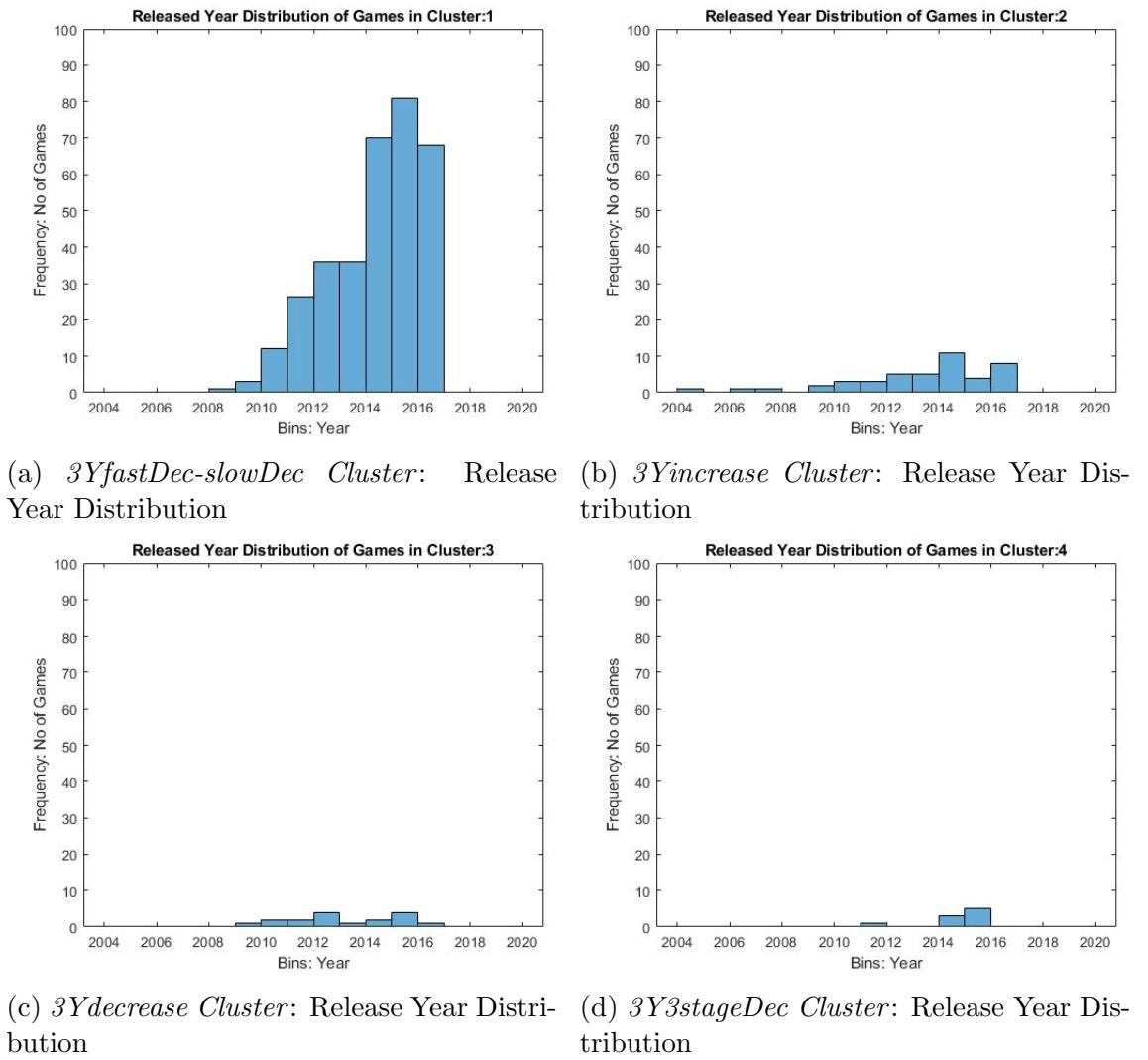


Figure A.13: Release year distribution of games displaying first three years archetypes

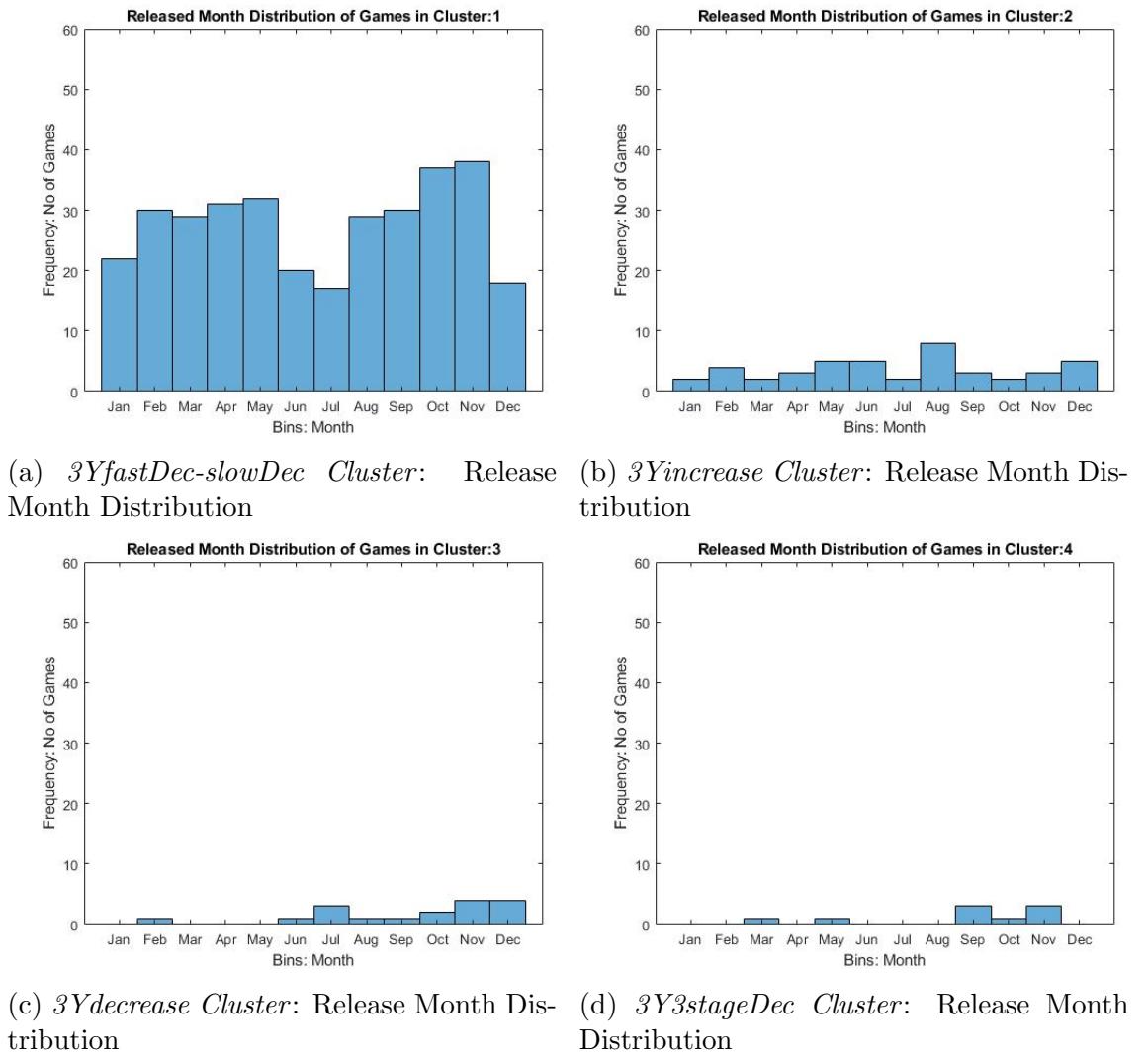


Figure A.14: Release month distribution of games displaying first three years archetypes

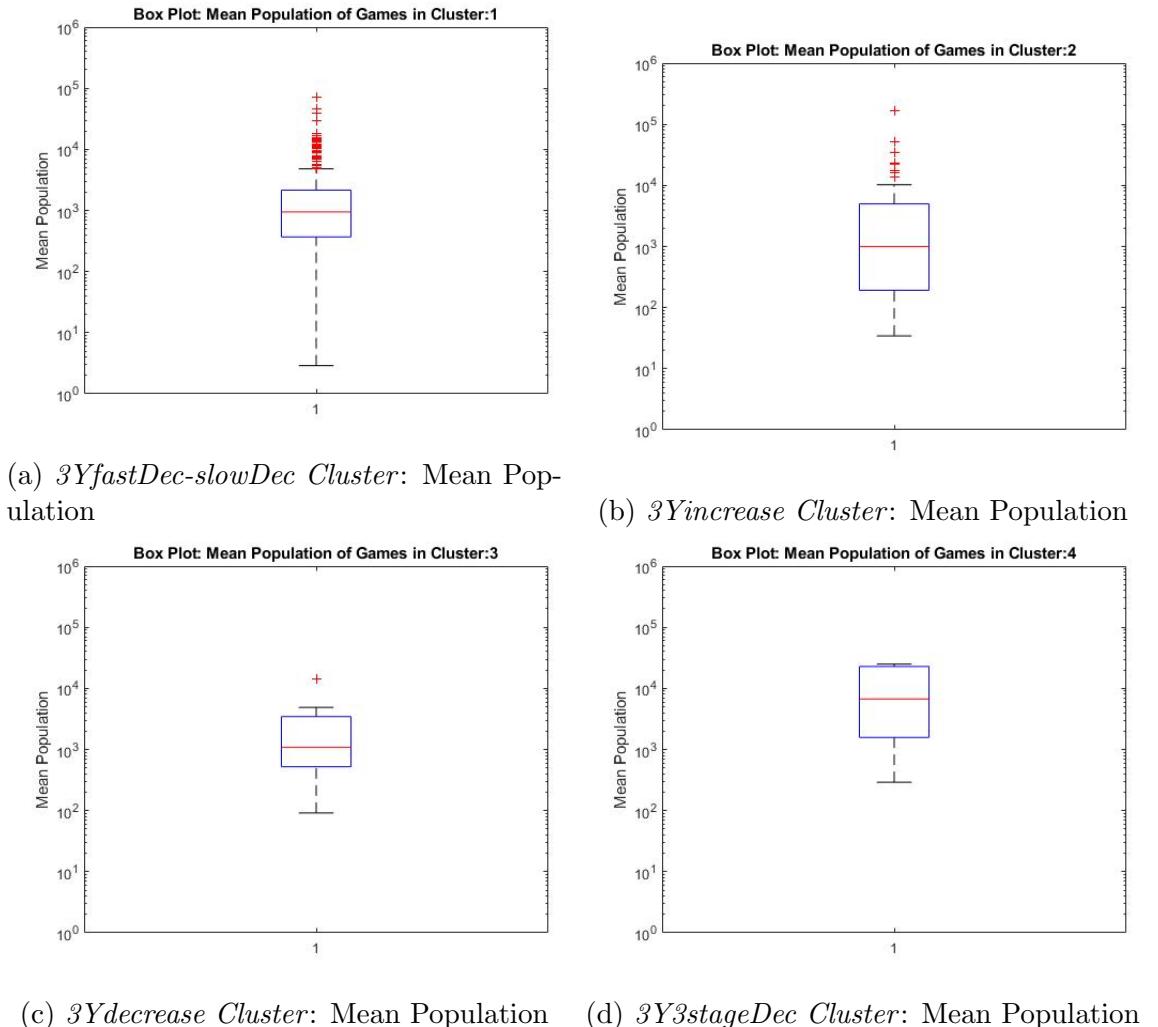


Figure A.15: Mean population statistics of games displaying first three years archetypes

### *A.3. Characteristics of games displaying archetypes of first three years (3Yclust)*

---

Play and Non-Free-to-Play games.

Cluster	F2P Percentage	Non-F2P Percentage
<i>3YfastDec-slowDec</i>	16.21	81.68
<i>3Yincrease</i>	27.27	72.72
<i>3Ydecrease</i>	64.70	35.29
<i>3Y3stageDec</i>	22	22

Table A.2: Free to Play games percentage of games displaying first three year archetypes

#### **Price of Games**

Figure A.16 depicts the price distribution of games. *3YfastDec-slowDec Cluster* has most games in the \$10-20 price range. *3Yincrease Cluster* has more games less than \$20 and \$30-40. *3Ydecrease Cluster* also has less than \$20 games and \$30-40 games. *3Y3stageDec Cluster* has games in the \$50-60 range and one game in less than \$10 range. However, it was identified that 66% of games in *3Y3stageDec Cluster* has become unavailable for purchase after first year, for instance Football Manager 2016 game. This is due to the yearly release of new versions of the game. These games are not depicted in the histogram as the most common price status of those games during the first three years are unavailable.

#### **Reviews**

The distribution of positive review percentages are depicted in Figure A.17. *3YfastDec-slowDec Cluster* has most games that have received 80-100% positive review percentages. The same can be observed in *3Yincrease Cluster*. *3Ydecrease Cluster* has games that has received 60-80% of positive reviews. *3Y3stageDec Cluster* has games in 40-60% and 80-100% percentage ranges.

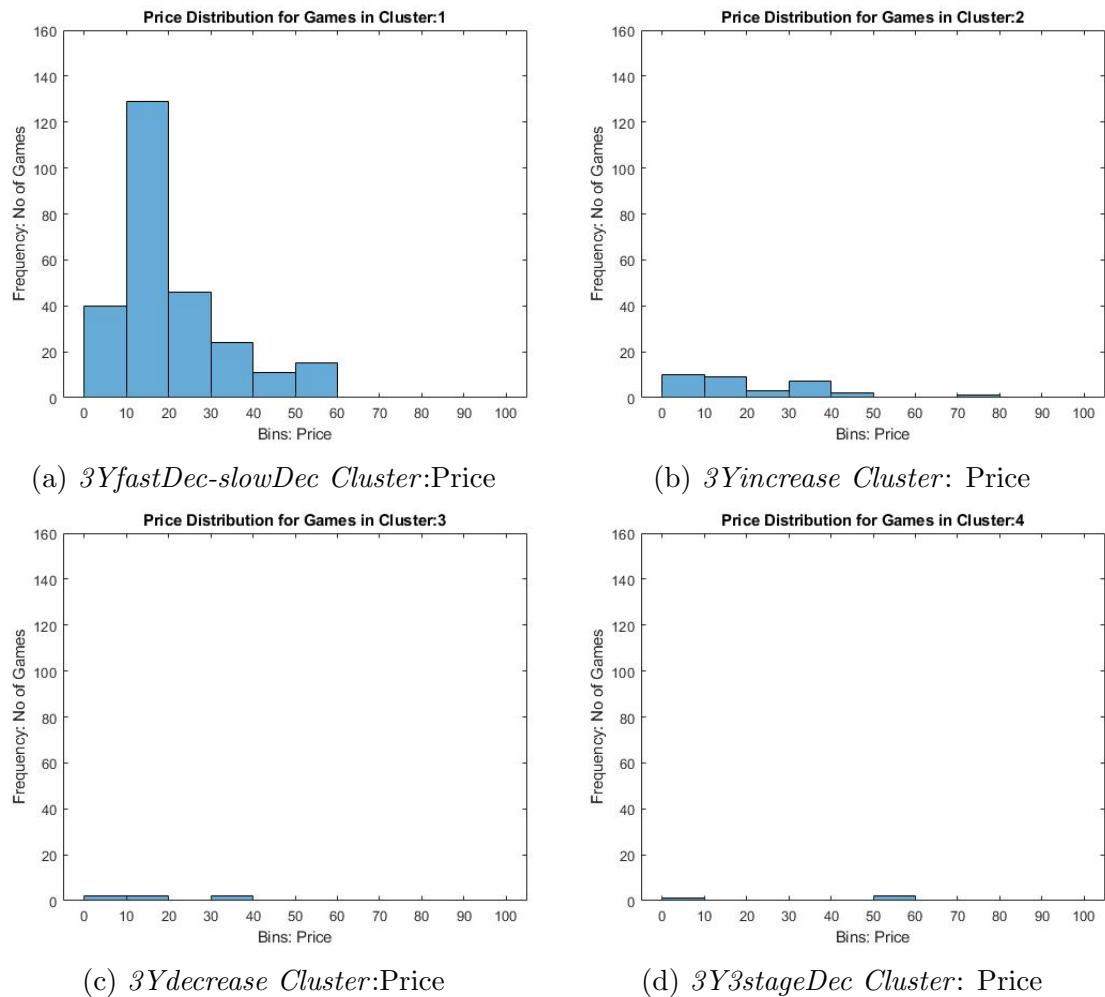


Figure A.16: Price distribution of games displaying first three years archetypes

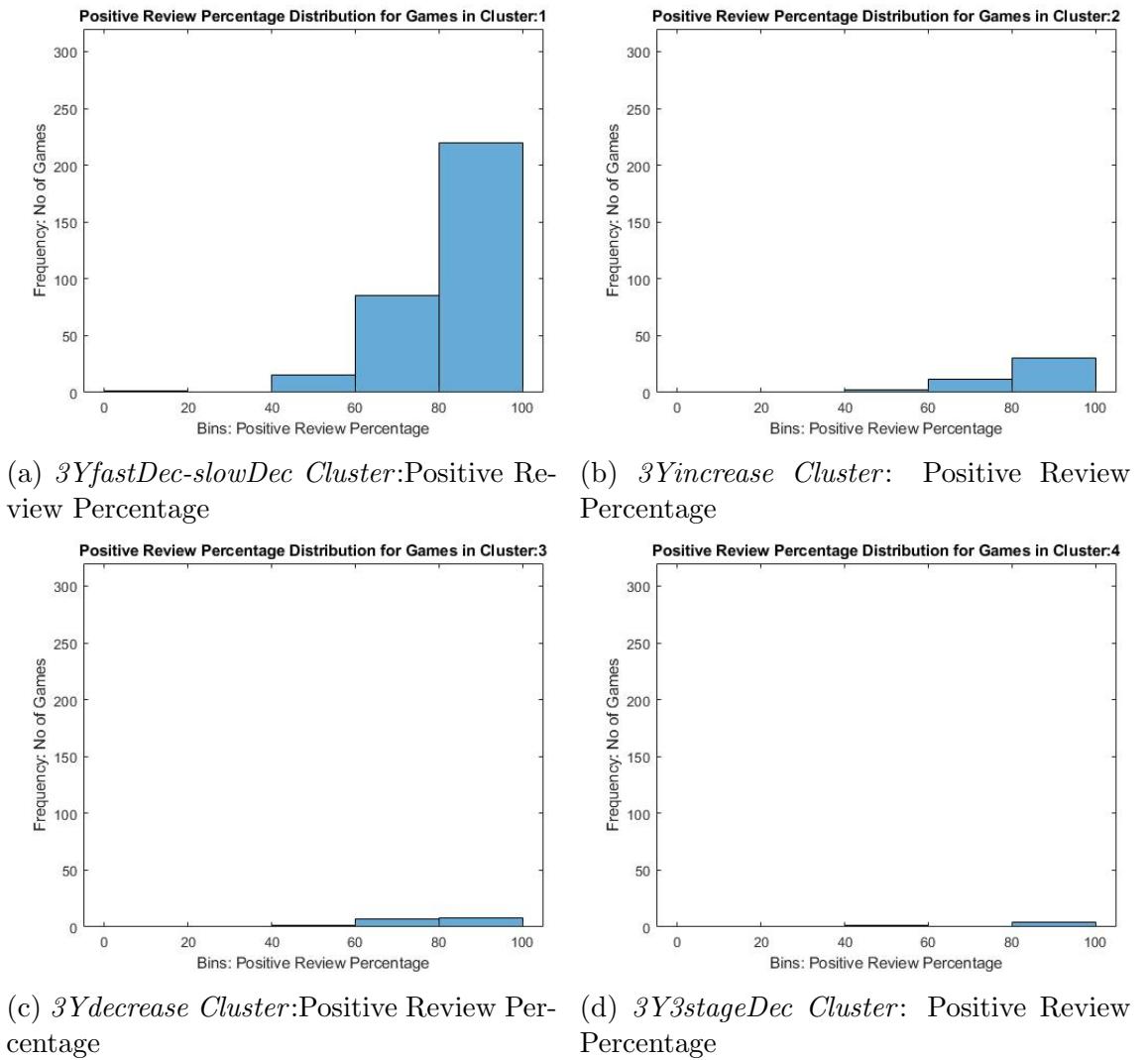


Figure A.17: Positive review percentage distribution of games displaying first year archetypes

# Bibliography

- [1] D. R. Rink and J. E. Swan, “Product life cycle research: A literature review,” *Journal of Business Research*, vol. 7, no. 3, pp. 219–242, Sep. 1979.
- [2] D. Cook, “Gamasutra - The Circle of Life: An Analysis of the Game Product Lifecycle,” [https://www.gamasutra.com/view/feature/129880/the\\_circle\\_of\\_life\\_an\\_analysis\\_of\\_.php?](https://www.gamasutra.com/view/feature/129880/the_circle_of_life_an_analysis_of_.php?) May 2007.
- [3] M. Gazecki, “The Game Life Cycle & Game Analytics: What metrics matter when?” Hamburg, 2012.
- [4] P. S. P. Cowpertwait and A. V. Metcalfe, *Introductory Time Series with R*, ser. Use R! Dordrecht ; New York: Springer, 2009.
- [5] NewZoo, “GLOBAL GAMES MARKET REPORT 2019,” 2019.
- [6] Statista, “Number of gamers worldwide 2023,” <https://www.statista.com/statistics/748044/number-video-gamers-world/>, Jan. 2021.
- [7] SuperData, “2019 Year In Review,” <https://www.superdataresearch.com/2019-year-in-review>.
- [8] A. Drachen and S. Connor, “Game Analytics for Games User Research,” in *Games User Research*. Oxford University Press, Mar. 2018, ch. Games User Research, pp. 333–353.

- [9] R. Sifa, A. Drachen, and C. Bauckhage, “Large-Scale Cross-Game Player Behavior Analysis on Steam,” in *Proceedings of the Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-15)*. AAAI Press, 2015, pp. 198–204.
- [10] M. S. El-Nasr, *Game Analytics: Maximizing the Value of Player Data*. New York: Springer, 2013.
- [11] A. Vivekanandarajah, “How data analytics software is changing the video game industry,” <https://seleritysas.com/blog/2018/12/14/data-analytics-software-video-game-industry/>, Dec. 2018.
- [12] J. Runge, P. Gao, F. Garcin, and B. Faltings, “Churn prediction for high-value players in casual social games,” in *2014 IEEE Conference on Computational Intelligence and Games*, Aug. 2014, pp. 1–8.
- [13] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, “Predicting Purchase Decisions in Mobile Free-to-Play Games,” in *AIIDE*, 2015.
- [14] A. Tyack, P. Wyeth, and D. Johnson, “The Appeal of MOBA Games: What Makes People Start, Stay, and Stop,” in *Annual Symposium on Computer-Human Interaction in Play*. ACM Press, 2016, pp. 313–325.
- [15] R. Sifa, C. Bauckhage, and A. Drachen, “The Playtime Principle: Large-scale cross-games interest modeling,” in *2014 IEEE Conference on Computational Intelligence and Games*, Aug. 2014, pp. 1–8.
- [16] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore, ““Alone together?”: Exploring the social dynamics of massively multiplayer online games,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, 2006, p. 407.
- [17] D. Pittman and C. GauthierDickey, “Characterizing Virtual Populations in Massively Multiplayer Online Role-Playing Games,” in *Advances in Multime-*

- dia Modeling*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Matern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. Boll, Q. Tian, L. Zhang, Z. Zhang, and Y.-P. P. Chen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 5916, pp. 87–97.
- [18] J. Kawale, A. Pal, and J. Srivastava, “Churn Prediction in MMORPGs: A Social Influence Based Approach,” in *2009 International Conference on Computational Science and Engineering*. IEEE, 2009, pp. 423–428.
- [19] M. O’Neill, E. Vaziripour, J. Wu, and D. Zappala, “Condensing Steam: Distilling the Diversity of Gamer Behavior,” in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC ’16. New York, NY, USA: ACM, 2016, pp. 81–95.
- [20] A. Drachen, M. Seif El-Nasr, and A. Canossa, “Game Analytics – The Basics,” in *Game Analytics: Maximizing the Value of Player Data*, M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds. London: Springer, 2013, pp. 13–40.
- [21] T. V. Fields, “Game Industry Metrics Terminology and Analytics Case Study,” in *Game Analytics: Maximizing the Value of Player Data*, M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds. London: Springer, 2013, pp. 53–71.
- [22] J. Hellemans, K. Willems, and M. Brengman, “Daily Active Users of Social Network Sites: Facebook, Twitter, and Instagram-Use Compared to General Social Network Site Use,” in *Advances in Digital Marketing and eCommerce*, ser. Springer Proceedings in Business and Economics, F. J. Martínez-López and S. D’Alessandro, Eds. Cham: Springer International Publishing, 2020, pp. 194–202.

- [23] D. Schneider, “Daily Active Users: Understand and increase your user engagement,” <https://www.similarweb.com/corp/blog/daily-active-users/>, Jul. 2020.
- [24] A. F. del Río, A. Guitart, and Á. Periáñez, “A Time Series Approach To Player Churn and Conversion in Videogames,” *arXiv:2003.10287 [cs, stat]*, Mar. 2020.
- [25] J. Junaidi, A. Julianto, N. Anwar, Safrizal, H. L. H. Spits Warnars, and K. Hashimoto, “Perfecting A Video Game with Game Metrics,” *TELKOM-NIKA Indonesian Journal of Electrical Engineering*, vol. 16, pp. 1324–1331, Jun. 2018.
- [26] X. Zhuang, J. Pang, A. Bharambe, and S. Seshan, “Player Dynamics in Massively Multiplayer Online Games,” *School of Computer Science, Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-CS-07-158*, p. 30, Oct. 2007.
- [27] S. Triberti, L. Milani, D. Villani, S. Grumi, S. Peracchia, G. Curcio, and G. Riva, “What matters is when you play: Investigating the relationship between online video games addiction and time spent playing over specific day phases,” *Addictive Behaviors Reports*, vol. 8, pp. 185–188, Dec. 2018.
- [28] R. King and T. de La Hera, “Gamer perception of endorsements from Fortnite Streamers on YouTube,” in *International Conference on the Foundations of Digital Games*, ser. FDG ’20. New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 1–3.
- [29] M. R. Johnson and J. Woodcock, “The impacts of live streaming and Twitch.tv on the video game industry,” *Media, Culture & Society*, vol. 41, no. 5, pp. 670–688, Jul. 2019.
- [30] H. Choi, D. Medlin, and S. Hunsinger, “The Effects of Discount Pricing Strategy on Sales of Software-as-a-Service (SaaS): Online Video Game Market Con-

text,” *Journal of Information Systems Applied Research*, vol. 10, no. 1, p. 55, Apr. 2017.

- [31] R. Flunger, A. Mladenow, and C. Strauss, “Game Analytics on Free to Play,” in *Big Data Innovations and Applications*, ser. Communications in Computer and Information Science, M. Younas, I. Awan, and S. Benbernou, Eds. Cham: Springer International Publishing, 2019, pp. 133–141.
- [32] C. Chambers, W.-c. Feng, S. Sahu, and D. Saha, “Measurement-based characterization of a collection of on-line games,” in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. ACM Press, 2005, p. 1.
- [33] D. Lin, “How Can Game Developers Leverage Data from Online Distribution Platforms? A Case Study of the Steam Platform,” Ph.D. dissertation, Queen’s University, Canada, 2019.
- [34] N. Y. Prathama, R. Asmara, and A. R. Barakbah, “Game Data Analytics using Descriptive and Predictive Mining,” in *2020 International Electronics Symposium (IES)*, Sep. 2020, pp. 398–405.
- [35] C. Dring, “What is happening with video game sales during coronavirus,” <https://www.gamesindustry.biz/articles/2020-03-28-what-is-happening-with-video-game-sales-during-coronavirus>, Mar. 2020.
- [36] P. Shanley, “Nielsen Reports 45 Percent Spike in U.S. Video Game Usage,” <https://www.hollywoodreporter.com/news/us-video-game-usage-up-45-percent-nielsen-reports-1288738>, Apr. 2020.
- [37] “GameAnalytics — Player Tracking & Analytics - 100% Free,” <https://gameanalytics.com/>.
- [38] “A Complete Solution for Game Analytics,” <https://www.cooladata.com/solutions/game-analytics>.

- [39] D. Choi and J. Kim, “Why People Continue to Play Online Games: In Search of Critical Design Factors to Increase Customer Loyalty to Online Contents,” *CyberPsychology & Behavior*, vol. 7, no. 1, pp. 11–24, Feb. 2004.
- [40] J. Moon, M. D. Hossain, G. L. Sanders, E. J. Garrity, and S. Jo, “Player Commitment to Massively Multiplayer Online Role-Playing Games (MMORPGs): An Integrated Model,” *International Journal of Electronic Commerce*, vol. 17, no. 4, pp. 7–38, Jul. 2013.
- [41] T. Debeauvais, B. Nardi, D. J. Schiano, N. Ducheneaut, and N. Yee, “If You Build It They Might Stay: Retention Mechanisms in World of Warcraft,” in *Proceedings of the 6th International Conference on Foundations of Digital Games*, ser. FDG ’11. New York, NY, USA: ACM, 2011, pp. 180–187.
- [42] B. E. Harrison and D. Roberts, “Analytics-Driven Dynamic Game Adaption for Player Retention in a 2-Dimensional Adventure Game,” in *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, Sep. 2014.
- [43] R. Sifa, S. Srikanth, A. Drachen, C. Ojeda, and C. Bauckhage, “Predicting Retention in Sandbox Games with Tensor Factorization-based Representation Learning.” IEEE, Sep. 2016, pp. 1–8.
- [44] A. Drachen, E. T. Lundquist, Y. Kung, P. S. Rao, D. Klabjan, R. Sifa, and J. Runge, “Rapid Prediction of Player Retention in Free-to-Play Mobile Games,” in *arXiv:1607.03202 [Cs, Stat]*, Jul. 2016.
- [45] M. Tamassia, W. Raffe, R. Sifa, A. Drachen, F. Zambetta, and M. Hitchens, “Predicting player churn in destiny: A Hidden Markov models approach to predicting player departure in a major online game,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, Sep. 2016, pp. 1–8.
- [46] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting, and C. Bauckhage, “Predicting player churn in the wild,” in *2014 IEEE Conference on Computational Intelligence and Games*, Aug. 2014, pp. 1–8.

- [47] K. Savetratanakaree, K. Sookhanaphibarn, S. Intakosum, R. Thawonmas, and K. T. Chen, “Departure Prediction of Online Game Players,” *Advanced Materials Research*, vol. 931-932, pp. 1370–1374, May 2014.
- [48] S. Demediuk, A. Murrin, D. Bulger, M. Hitchens, A. Drachen, W. L. Raffe, and M. Tamassia, “Player retention in league of legends: A study using survival analysis,” in *Proceedings of the Australasian Computer Science Week Multi-conference*, ser. ACSW ’18. Brisband, Queensland, Australia: Association for Computing Machinery, Jan. 2018, pp. 1–9.
- [49] A. Drachen, A. Canossa, and G. N. Yannakakis, “Player modeling using self-organization in Tomb Raider: Underworld,” in *2009 IEEE Symposium on Computational Intelligence and Games*, Sep. 2009, pp. 1–8.
- [50] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, “Guns, swords and data: Clustering of player behavior in computer games in the wild,” in *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, Sep. 2012, pp. 163–170.
- [51] J. Valls-Vargas, S. Ontañón, and J. Zhu, “Exploring Player Trace Segmentation for Dynamic Play Style Prediction,” in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, Sep. 2015.
- [52] E. S. Siqueira, C. D. Castanho, G. N. Rodrigues, and R. P. Jacobi, “A Data Analysis of Player in World of Warcraft Using Game Data Mining,” in *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, Nov. 2017, pp. 1–9.
- [53] N. Hanner and R. Zarnekow, “Purchasing Behavior in Free to Play Games: Concepts and Empirical Validation,” in *2015 48th Hawaii International Conference on System Sciences*, Jan. 2015, pp. 3326–3335.
- [54] W. Yang, G. Yang, T. Huang, L. Chen, and Y. E. Liu, “Whales, Dolphins, or Minnows? Towards the Player Clustering in Free Online Games Based on

- Purchasing Behavior via Data Mining Technique,” in *2018 IEEE International Conference on Big Data (Big Data)*, Dec. 2018, pp. 4101–4108.
- [55] H. Xie, S. Devlin, D. Kudenko, and P. Cowling, “Predicting player disengagement and first purchase with event-frequency based data representation,” in *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, Aug. 2015, pp. 230–237.
- [56] X. Fu, X. Chen, Y.-T. Shi, I. Bose, and S. Cai, “User segmentation for retention management in online social games,” *Decision Support Systems*, vol. 101, pp. 51–68, Sep. 2017.
- [57] R. Sifa, C. Bauckhage, and A. Drachen, “Archetypal Game Recommender Systems,” in *LWA*, 2014.
- [58] M. O. Riedl and A. Zook, “AI for game production,” in *2013 IEEE Conference on Computational Intelligence in Games (CIG)*, Aug. 2013, pp. 1–8.
- [59] A. Vítek, “Cross-Game Modeling of Player’s Behaviour in Free-To-Play Games,” in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP ’20. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 384–387.
- [60] N. Shaker and M. Abou-Zleikha, “Transfer learning for cross-game prediction of player experience,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, Sep. 2016, pp. 1–8.
- [61] C. Pedersen, J. Togelius, and G. N. Yannakakis, “Modeling Player Experience for Content Creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 54–67, Mar. 2010.
- [62] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis, “Predicting player behavior in Tomb Raider: Underworld,” in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, Aug. 2010, pp. 178–185.

- [63] J. Blackburn, R. Simha, N. Kourtellis, X. Zuo, C. Long, M. Ripeanu, J. Skvoretz, and A. Iamnitchi, “Cheaters in the Steam Community Gaming Social Network,” *arXiv:1112.4915 [physics]*, Dec. 2011.
- [64] F. Baumann, D. Emmert, H. Baumgartl, and R. Buettner, “Hardcore Gamer Profiling: Results from an unsupervised learning approach to playing behavior on the Steam platform,” *Procedia Computer Science*, vol. 126, pp. 1289–1297, Jan. 2018.
- [65] R. Sifa, C. Ojeda, and C. Bauckhage, “User Churn Migration Analysis with DEDICOM,” in *RecSys*, Vienna,Austria, Sep. 2015, p. 4.
- [66] C. Bauckhage, K. Kersting, R. Sifa, C. Thurau, A. Drachen, and A. Canossa, “How players lose interest in playing a game: An empirical study based on distributions of total playing times,” in *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, Sep. 2012, pp. 139–146.
- [67] M. Trněný, “Machine Learning for Predicting Success of Video Games,” Ph.D. dissertation, 2017.
- [68] A. Pathak, K. Gupta, and J. McAuley, “Generating and Personalizing Bundle Recommendations on Steam,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Shinjuku Tokyo Japan: ACM, Aug. 2017, pp. 1073–1076.
- [69] N. Yee, “Visualizing How Steam Tags Are Related,” <https://quanticfoundry.com/2018/01/24/visualizing-steam-tags-related/>, Jan. 2018.
- [70] T. W. Windleharth, J. Jett, M. Schmalz, and J. H. Lee, “Full Steam Ahead: A Conceptual Analysis of User-Supplied Tags on Steam,” *Cataloging & Classification Quarterly*, vol. 54, no. 7, pp. 418–441, Oct. 2016.

- [71] D. Lin, C.-P. Bezemer, Y. Zou, and A. E. Hassan, “An empirical study of game reviews on the Steam platform,” *Empirical Software Engineering*, Jun. 2018.
- [72] H.-N. Kang, H.-R. Yong, and H.-S. Hwang, “A Study of Analyzing on Online Game Reviews using a Data Mining Approach: STEAM Community Data,” *International Journal of Innovation*, vol. 8, no. 2, p. 6, 2017.
- [73] D. Lin, C.-P. Bezemer, and A. E. Hassan, “Studying the urgent updates of popular games on the Steam platform,” *Empirical Software Engineering*, vol. 22, no. 4, pp. 2095–2126, Aug. 2017.
- [74] ——, “An empirical study of early access games on the Steam platform,” *Empirical Software Engineering*, vol. 23, no. 2, pp. 771–799, Apr. 2018.
- [75] R. Becker, Y. Chernihov, Y. Shavitt, and N. Zilberman, “An analysis of the Steam community network evolution.” IEEE, Nov. 2012, pp. 1–5.
- [76] S. Galyonkin, “Steam Spy: The missing manual,” <https://galyonk.in/steam-spy-the-missing-manual-cc22ef6eebe1>, Jul. 2015.
- [77] “Steam Charts - Tracking What’s Played,” <https://steamcharts.com/>.
- [78] “Steam Database,” <https://steamdb.info/>.
- [79] L. Mellon, “Applying metrics driven development to MMO costs and risks,” *Versant Corporation Tech. Rep.*, 2009.
- [80] A. Koskenvoima and M. Mäntymäki, “Why Do Small and Medium-Size Freemium Game Developers Use Game Analytics?” in *Open and Big Data Management and Innovation*, ser. Lecture Notes in Computer Science, M. Janssen, M. Mäntymäki, J. Hidders, B. Klievink, W. Lamersdorf, B. van Loenen, and A. Zuiderwijk, Eds. Cham: Springer International Publishing, 2015, pp. 326–337.

- [81] L. Doucet, “Steam Traffic Patterns Deep Dive,” <https://www.fortressofdoors.com/steam-traffic-patterns-and-discoverability/>, Nov. 2014.
- [82] H. S. Mahmassani, R. B. Chen, Y. Huang, D. Williams, and N. Contractor, “Time to Play?: Activity Engagement in Multiplayer Online Role-Playing Games,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2157, no. 1, pp. 129–137, Jan. 2010.
- [83] D. Cook, “Game Genre Lifecycle: Part I,” <https://lostgarden.home.blog/2005/05/06/game-genre-lifecycle-part-i/>, May 2005.
- [84] X. Liu, C. Guo, and H. Jia, “Mobile application life cycle characterization via apple app store rank,” *Proceedings of the American Society for Information Science and Technology*, vol. 51, no. 1, pp. 1–4, 2014.
- [85] N. Drašković, M. Marković, and K. Žnidar, “Product lifecycle strategies in digital world,” *International journal of management cases*, vol. 20, no. 3, pp. 29–43, Aug. 2018.
- [86] G. J. Tellis and C. M. Crawford, “An Evolutionary Approach to Product Growth Theory,” *Journal of Marketing*, vol. 45, no. 4, pp. 125–132, 1981.
- [87] “Discounting (Steamworks Documentation),” <https://partner.steamgames.com/doc/marketing/discounts>.
- [88] M. Consalvo and C. Paul, “If you are feeling bold, ask for \$3’: Value Crafting and Indie Game Developers,” in *Digital Games Research Association DiGRA*, 2017, p. 14.
- [89] M. Wu, “Case Study the Indie Game Industry and Help Indie Developers Achieve Their Success in Digital Marketing,” M.S., Northeastern University, United States – Massachusetts, 2017.

- [90] M. Hackett, “A Wizard’s Lizard by the numbers: Our HTML5 game on Steam,” [https://www.gamasutra.com/blogs/MattHackett/20141009/227341/A\\_Wizards\\_Lizard\\_b](https://www.gamasutra.com/blogs/MattHackett/20141009/227341/A_Wizards_Lizard_b), Sep. 2014.
- [91] J. Ruohonen and S. Hyrynsalmi, “Evaluating the use of internet search volumes for time series modeling of sales in the video game industry,” *Electronic Markets*, vol. 27, no. 4, pp. 351–370, Nov. 2017.
- [92] O. Schaer, N. Kourentzes, and R. Fildes, “Demand forecasting with user-generated online information,” *International Journal of Forecasting*, vol. 35, no. 1, pp. 197–212, Jan. 2019.
- [93] R. Rossetti, “Forecasting the sales of console games for the Italian market,” *Ekonometria*, no. vol. 23 no. 3, pp. 76–96, 2019.
- [94] A. Guitart, P. P. Chen, P. Bertens, and Á. Periáñez, “Forecasting Player Behavioral Data and Simulating In-Game Events,” in *Advances in Information and Communication Networks*, ser. Advances in Intelligent Systems and Computing, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2019, pp. 274–293.
- [95] A. G. Department of Health, “What you need to know about coronavirus (COVID-19),” <https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/what-you-need-to-know-about-coronavirus-covid-19>, Mar. 2020.
- [96] L. Di Renzo, P. Gualtieri, F. Pivari, L. Soldati, A. Attinà, G. Cinelli, C. Leggeri, G. Caparello, L. Barrea, F. Scerbo, E. Esposito, and A. De Lorenzo, “Eating habits and lifestyle changes during COVID-19 lockdown: An Italian survey,” *Journal of Translational Medicine*, vol. 18, no. 1, p. 229, Jun. 2020.
- [97] SteamDB, “Steam Database on Twitter: ”#Steam has just reached a new concurrent online user record of 20 million, with 6.2 million cur-

- rently in-game, likely due to many people staying at home due to the #coronavirus. <https://t.co/bzLMfMOJvD> #COVID19” / Twitter,” <https://twitter.com/steamdb/status/1239180882826715136>, Mar. 2020.
- [98] Nielsen, “3, 2, 1 Go! Video Gaming is at an All-Time High During COVID-19,” <https://www.nielsen.com/us/en/insights/article/2020/3-2-1-go-video-gaming-is-at-an-all-time-high-during-covid-19>, Mar. 2020.
- [99] H. Taylor, “Gaming and live streaming rise globally amid COVID-19 crisis,” <https://www.gamesindustry.biz/articles/2020-03-18-gaming-and-live-streaming-rise-globally-amid-covid-19-crisis>, Mar. 2020.
- [100] D. Takahashi, “WHO and game companies launch #PlayApartTogether to promote physical distancing,” Mar. 2020.
- [101] R. Wiseman and M. Jacob, “Can You Save The World ? by Martin Jacob,” <https://martin-jacob.itch.io/can-you-save-the-world>, 2020.
- [102] D. L. King, P. H. Delfabbro, J. Billieux, and M. N. Potenza, “Problematic online gaming and the COVID-19 pandemic,” *Journal of Behavioral Addictions*, vol. 9, no. 2, pp. 184–186, Apr. 2020.
- [103] S. Laato, A. K. M. N. Islam, and T. H. Laine, “Did location-based games motivate players to socialize during COVID-19?” *Telematics and Informatics*, p. 101458, Jun. 2020.
- [104] N. Smith, “The giants of the video game industry have thrived in the pandemic. Can the success continue?” *Washington Post*, May 2020.
- [105] P. J. Brockwell and R. A. Davis, “Introduction,” in *Introduction to Time Series and Forecasting*, ser. Springer Texts in Statistics, P. J. Brockwell and R. A. Davis, Eds. Cham: Springer International Publishing, 2016, pp. 1–37.
- [106] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2018.

- [107] K. Bandara, C. Bergmeir, and S. Smyl, “Forecasting Across Time Series Databases using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach,” *arXiv:1710.03222 [cs, econ, stat]*, Oct. 2017.
- [108] G. Mahalakshmi, S. Sridevi, and S. Rajaram, “A survey on forecasting of time series data,” in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE’16)*, Jan. 2016, pp. 1–8.
- [109] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, Mar. 2020.
- [110] Matlab, “Nonlinear autoregressive neural network - MATLAB narnet - MathWorks Australia,” <https://au.mathworks.com/help/deeplearning/ref/narnet.html>.
- [111] ——, “Nonlinear autoregressive neural network with external input - MATLAB narxnet - MathWorks Australia,” <https://au.mathworks.com/help/deeplearning/ref/narxnet.html>.
- [112] J. Han and M. Kamber, *Data Mining : Concepts and Techniques*, 2006.
- [113] T. Warren Liao, “Clustering of time series data—a survey,” *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005.
- [114] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, “Time-series clustering – A decade review,” *Information Systems*, vol. 53, pp. 16–38, Oct. 2015.
- [115] A. Mueen and E. Keogh, “Extracting Optimal Performance from Dynamic Time Warping,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 2129–2130.
- [116] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, “Towards parameter-free data mining,” in *Proceedings of the 2004 ACM SIGKDD International Con-*

- ference on Knowledge Discovery and Data Mining - KDD '04.* Seattle, WA, USA: ACM Press, 2004, p. 206.
- [117] O. Maimon and L. Rokach, Eds., *Data Mining and Knowledge Discovery Handbook*, 2nd ed. New York: Springer, 2010.
- [118] D. Wannigamage, M. Barlow, E. Lakshika, and K. Kasmarik, “Steam Games Dataset : Player count history, Price history and data about games,” vol. 1, Aug. 2020.
- [119] Steam, “Steam :: Steamworks Development :: Steam - 2018 Year in Review,” <https://store.steampowered.com/news/group/4145017/view/>, Jan. 2019.
- [120] N. Höglund, “Digital distribution of video games for PC : A SWOT analysis,” Ph.D. dissertation, 2014.
- [121] C. Gough, “Steam demographics: Users by country,” <https://www.statista.com/statistics/826870/steam-distribution-country/>, 2018.
- [122] I. Pitas and A. N. Venetsanopoulos, “Median Filters,” in *Nonlinear Digital Filters*, ser. The Springer International Series in Engineering and Computer Science. Springer, Boston, MA, 1990, pp. 63–116.
- [123] “How To Identify Patterns in Time Series Data: Time Series Analysis,” <http://www.statsoft.com/Textbook/Time-Series-Analysis>.
- [124] H. Musbah, M. El-Hawary, and H. Aly, “Identifying Seasonality in Time Series by Applying Fast Fourier Transform,” in *2019 IEEE Electrical Power and Energy Conference (EPEC)*, Oct. 2019, pp. 1–4.
- [125] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control, Revised Ed.* Holden-Day, 1976.

- [126] Z. Wu, N. E. Huang, S. R. Long, and C.-K. Peng, “On the trend, detrending, and variability of nonlinear and nonstationary time series,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 38, pp. 14 889–14 894, Sep. 2007.
- [127] G. Kirchgässner, J. Wolters, and U. Hassler, *Introduction to Modern Time Series Analysis*. Springer Science & Business Media, Oct. 2012.
- [128] A. Saas, A. Guitart, and Á. Periáñez, “Discovering Playing Patterns: Time Series Clustering of Free-To-Play Game Data,” *arXiv:1710.02268 [cs, stat]*, pp. 1–8, Sep. 2016.
- [129] P. Montero and J. A. Vilar, “**TSclust** : An *R* Package for Time Series Clustering,” *Journal of Statistical Software*, vol. 62, no. 1, 2014.
- [130] X. Wang, K. Smith, and R. Hyndman, “Characteristic-Based Clustering for Time Series Data,” *Data Mining and Knowledge Discovery*, vol. 13, no. 3, pp. 335–364, Sep. 2006.
- [131] R. R. Sokal and F. J. Rohlf, “The Comparison of Dendograms by Objective Methods,” *Taxon*, vol. 11, no. 2, pp. 33–40, 1962.
- [132] T. Levitt, “Exploit the Product Life Cycle,” *Harvard Business Review*, no. November 1965, Nov. 1965.
- [133] R. Vernon, “International Investment and International Trade in the Product Cycle,” *The Quarterly Journal of Economics*, vol. 80, no. 2, pp. 190–207, May 1966.
- [134] E. Camponogara and L. F. Nazari, “Models and Algorithms for Optimal Piecewise-Linear Function Approximation,” *Mathematical Problems in Engineering*, 2015.
- [135] P. Djundik, “How and why we moved graph storage to InfluxDB,” <https://steamdb.info/blog/graph-storage-influxdb/>, Dec. 2017.

- [136] W. S. Cleveland, “Robust Locally Weighted Regression and Smoothing Scatterplots,” *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, Dec. 1979.
- [137] Matlab, “Filtering and Smoothing Data - MATLAB & Simulink - MathWorks Australia,” <https://au.mathworks.com/help/curvefit/smoothing-data.html>.
- [138] E. Vieth, “Fitting piecewise linear regression functions to biological responses,” *Journal of Applied Physiology (Bethesda, Md.: 1985)*, vol. 67, no. 1, pp. 390–396, Jul. 1989.
- [139] T. O’Brien, “Steam sales data shows that you don’t play the games you buy,” <https://www.engadget.com/2014-04-16-steam-sales.html>, Apr. 2014.
- [140] J. Adnan, N. G. N. Daud, M. T. Ishak, Z. I. Rizman, and M. I. A. Rahaman, “Tansig activation function (of MLP network) for cardiac abnormality detection,” in *AIP Conference Proceedings*, 2018, p. 020006.
- [141] S. Tamura and M. Tateishi, “Capabilities of a four-layered feedforward neural network: Four layers versus three,” *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 251–255, Mar. 1997.
- [142] Q. Chen, Z. Meng, X. Liu, Q. Jin, and R. Su, “Decision Variants for the Automatic Determination of Optimal Feature Subset in RF-RFE,” *Genes*, vol. 9, no. 6, Jun. 2018.
- [143] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.
- [144] E. Romero and J. M. Sopena, “Performing Feature Selection With Multilayer Perceptrons,” *IEEE Transactions on Neural Networks*, vol. 19, no. 3, pp. 431–441, Mar. 2008.
- [145] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, Jan. 1991.

## *Bibliography*

---

- [146] J. Heaton, *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks*, 1st ed. Heaton Research, Inc., Nov. 2015.
- [147] H. P. Gavin, “The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems,” 2016.
- [148] “Levenberg-Marquardt backpropagation - MATLAB trainlm - MathWorks Australia,” <https://au.mathworks.com/help/deeplearning/ref/trainlm.html>.
- [149] Matlab, “Design Time Series NARX Feedback Neural Networks - MATLAB & Simulink - MathWorks Australia,” <https://au.mathworks.com/help/deeplearning/ug/design-time-series-narx-feedback-neural-networks.html>.
- [150] G. P. Zhang, “Neural Networks for Time-Series Forecasting,” in *Handbook of Natural Computing*, G. Rozenberg, T. Bäck, and J. N. Kok, Eds. Berlin, Heidelberg: Springer, 2012, pp. 461–477.
- [151] D. Howley, “The world is turning to video games amid coronavirus outbreak,” <https://finance.yahoo.com/news/coronavirus-world-turning-video-games-150704969.html>, Mar. 2020.
- [152] “Novel Coronavirus (COVID-19) Cases Data by Humanitarian Data Exchange,” <https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases>.
- [153] “Steam demographics: Users by country,” <https://www.statista.com/statistics/826870/steam-distribution-country/>.
- [154] R. Muccari and D. Chow, “Coronavirus timeline: Tracking the critical moments of COVID-19,” <https://www.nbcnews.com/health/health-news/coronavirus-timeline-tracking-critical-moments-covid-19-n1154341>.
- [155] “Tabletop Simulator on Steam,” [https://store.steampowered.com/app/286160/Tabletop\\_Simulator](https://store.steampowered.com/app/286160/Tabletop_Simulator).

## *Bibliography*

---

- [156] “The Jackbox Party Pack 3 on Steam,” [https://store.steampowered.com/app/434170/The\\_Jackbox\\_Party\\_Pack\\_3/](https://store.steampowered.com/app/434170/The_Jackbox_Party_Pack_3/).
- [157] “Plague Inc: Evolved on Steam,” [https://store.steampowered.com/app/246620/Plague\\_Inc\\_Evolved/](https://store.steampowered.com/app/246620/Plague_Inc_Evolved/).
- [158] D. Rey and M. Neuhäuser, “Wilcoxon-Signed-Rank Test,” in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg: Springer, 2011, pp. 1658–1659.
- [159] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. CRC press, 1984.
- [160] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
- [161] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, 2nd ed., ser. Springer Series in Statistics. New York: Springer-Verlag, 2009.
- [162] “Formula 1 launches Virtual Grand Prix Series to replace postponed races,” <https://www.formula1.com/en/latest/article.formula-1-launches-virtual-grand-prix-series-to-replace-postponed-races.1znLAbPzBbCQPj1IDMeiOi.html>, Mar. 2020.
- [163] J. Jackson, “What Gamers Are Playing & Watching During the Coronavirus Lockdown: Player Share & Viewership Spikes for Games & Genres,” <https://newzoo.com/insights/articles/games-gamers-are-playing-watching-during-coronavirus-covid19-lockdown-quarantine/>, Apr. 2020.
- [164] N. Yee, “Visualizing How Steam Tags Are Related,” <https://quanticfoundry.com/2018/01/24/visualizing-steam-tags-related/>, Jan. 2018.