How contextual information affects the performance of predictive behavioural modelling in games.

Elizabeth Louise Watson

September 2017

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Big Data

Computing Science and Mathematics
University of Stirling

Abstract

Mobile gaming is increasingly becoming one of the largest areas of the games industry as the number of smartphone and tablet users is growing. Free-to-play games represent a significant proportion of the games available on mobile devices. Deriving insights from player behaviour is key to the success of free-to-play games. Modelling and predicting future player behaviour enables developers to implement data-driven decisions in development and player management strategies.

This paper considers approaches to estimating player retention and prediction of player churn. Player retention is analysed to discover the relationship between player metrics and retention. Furthermore, evaluation of the performance of retention prediction models gives insight into how contextual information about a player impacts the ability to estimate player retention.

Secondly, the study considers how contextual information about a game impacts the performance of churn modelling. The paper presents an approach to churn modelling which is game agnostic, hence, the model is applicable across all free-to-play games. Features of the model measure generic elements of player behaviour such as the time between events, playtime and the variability of event types. Comparisons are drawn between a decision tree classifier and a linear regression prediction in identifying those players about to churn.

Finally, a game specific model is developed as a case study to compare the performance of a game specific model against the game agnostic model. Performance measures suggest that there is little difference in the accuracy of churn prediction for a game specific and game agnostic approach to churn modelling.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following:

- The code discussed in Section 3.2 which runs threaded queries against the database was written by Nicholas Ross, an Assistant Professor at the University of San Francisco, who works closely with deltaDNA.

Signature **Date**

Acknowledgements

I would like to express my gratitude to deltaDNA for enabling me to undertake this project. Specifically, I would like to thank my supervisor at deltaDNA, Isaac Roseboom, for the useful comments, engagement, and guidance through the learning process of this thesis. Furthermore, I would like to thank Nicholas Ross for introducing me to this topic as well for support on the way.

I would like to thank my university supervisor, Dr Marwan Fayed, for his advice and guidance throughout this study. In addition, I would like to extend my thanks to Dr Gabriela Ochoa for her role as a second marker of this project.

Thank you to MBN Solutions for pairing me with my placement company. Finally, I would like to extend my gratitude to The Data Lab, not only for providing the funding of my placement on this course, but giving me the opportunity to attend multiple events and undertake an industrial placement.

Table of Contents

Abstract i
Attestation..... ii
Acknowledgements iii
Table of Contents..... iv
List of Figures..... vi
List of Tables vii
1 Introduction 1
1.1 Background and Context 1
1.2 Scope and Objectives..... 2
1.2.1 Retention..... 2
1.2.2 Churn 2
1.3 Achievements 3
1.4 Overview of Dissertation..... 4
2 State-of-The-Art 5
2.1 Literature Review 5
2.1.1 Behaviour Modelling..... 5
2.1.2 Games Industry..... 6
2.1.3 Churn Modelling in Games 7
2.2 Methodology..... 8
2.2.1 Nonlinear Least Squares..... 8
2.2.2 Huber Regressor 8
2.2.3 Decision Tree..... 10
2.2.4 Measurement of Model Accuracy..... 11
3 Data Acquisition and Access 14
3.1 Data Acquisition 14
3.2 Accessing Data 15
4 Player Retention Prediction..... 16
4.1 Data and Pre-processing 17
4.2 Player Metrics..... 17
4.3 Visualising Features..... 19
4.4 Modelling..... 21
4.4.1 Distribution of Retention Curve 22
4.4.2 Predicting Retention 22
4.5 Results 23

4.6 Evaluation & Next Steps 25
5 A Game Agnostic Approach to Churn Modelling 26
5.1 Game Agnostic Approach 26
5.2 Method 27
5.2.1 Decision Tree 29
5.2.2 Huber Loss Regression 29
5.3 Features 30
5.4 Data 32
5.5 Results 32
5.5.1 Decision Tree 32
5.5.2 Huber Loss Regression 33
5.6 Analysis 35
5.6.1 Feature Importance 35
5.6.2 Comparison of Models 36
6 A Game Specific Approach to Churn Modelling 38
6.1 Details of the Game 38
6.2 Method 38
6.3 Features 39
6.4 Data 40
6.5 Results 40
6.5.1 Game Specific Model 40
6.5.2 Game Agnostic Model 41
6.6 Analysis 41
7 Conclusion 43
7.1 Summary 43
7.2 Evaluation 43
7.3 Future Work 44
References 46

List of Figures

Figure 1: Diagram of the events which occur when a player completes a mission in a game. 14
Figure 2: The retention prediction model as displayed on the deltaDNA platform when attempting to optimise a sample of 20 or less players [10]....16
Figure 3: Gender splits across the game genres on deltaDNA platform [11] 18
Figure 4: Graph of the retention curves for 12 player segments. The colour of the curve outlines whether players in the segment have high or low game time on install day. 20
Figure 5: Graph of the retention curves for 12 player segments. The colour of the curves are determined by the value of player platform type. 21
Figure 6: Graph of the retention curves for 12 player segments. The value of the user country metric determines the colour of the curves. 21
Figure 7: The true and predicted player retention curves based on a cohort of players from a basket of 10 games. Predicted curve is fit using 7 days of retention rates. 23
Figure 8: The true and predicted player retention curves for each player segment across a basket of 10 games. Predicted curves are fit using retention values day 0 to 7. 24
Figure 9: The true and predicted player retention curves for a single model and average of segment models. Predicted curves are fit using 7 days of retention rates. 24
Figure 10: Player activity for 2 days following install. 27
Figure 11: Scenarios outlining two possible player lifetimes across an observation period. 28
Figure 12: Regression approach to identifying instances of churn. 30

List of Tables

Table 1: Output Values of a Binary Classification. 11
Table 2: Player metrics and the values they can take used to segment players. 19
Table 3: Decision tree results for a game agnostic model. 33
Table 4: Results of the accuracy of the Huber Loss regression method in labelling churn days for a game agnostic model based on various cut-off values. 33
Table 5: Results of accuracy measures for Huber Loss regression method of identifying churn days based on a cut-off value of 1.75. 34
Table 6:Accuracy measures for Huber Loss regression method of churn modelling using varied cut-off values. 34
Table 7: Decision tree to classify churn days - top 3 nodes. 35
Table 8: Performance measure results for the game specific churn models using Game A data. 40
Table 9: Performance measure results for the game agnostic churn models using Game A data. 41

1 Introduction

By 2020, the mobile games industry is expected to generate a revenue of $64.9 billion [1]. Ever since the introduction of smartphones, the number of games developers producing mobile games has increased significantly. As smartphones and tablets become more accessible, this industry is will see continued growth.

1.1 Background and Context

The results of a study in [2], estimate that 71% of UK adults currently own a smartphone with 47% of smartphone users using their phone to play mobile games [3]. The mobile games industry has seen substantial growth in the past 10 years and by 2020 it is expected to account for over half of the revenue produced by the gaming industry [1].

As the number of mobile games available increased, game developers had to discover methods to obtain the competitive edge and encourage players to choose their game over the number of games available. Hence, the free-to-play model was introduced.

Free-to-play games offer the game to be downloaded for free. Monetisation is then achieved by offering in-app purchases, for example virtual currency, and in-app advertising. The free-to-play model has proven successful for many games, one of the most known free-to-play mobile games is Candy Crush Saga. In 2014, Candy Crush Saga was reported to have a player base of 93 million players and generated an estimated £298 million from in-app purchases across three months [4].

It is estimated that 50% of free-to-play games have a 1% conversion rate, that is 1% of all players will convert to payers [5]. This means that the average revenue per daily active user of a game can be a very small amount. Estimates suggest that average revenue per daily active user of a free-to-play game fall between $0.005 and $1[6]. Therefore, to earn revenue from a game requires a large player base. Whilst it is possible to generate a large player base through marketing campaigns, it is important to retain those players play for a long period of time to generate revenue [7].

Player retention is a measure of the percentage of returning players on each day after install. Conversely, player churn is the measure of 1 – retention. Hence, a player is said to have churned when they are no longer actively playing a game. Data mining can be used to predict player retention and churn. Prediction of player retention enables forecasting of future values such as the lifetime value of a customer. Whereas, predicting when a player is likely to churn enables implementation of strategies to intervene when a player is about to churn which will encourage that player to play for longer.

This project was carried out as part of a placement with the games analytics company deltaDNA. deltaDNA are a service provider of a platform which enables game developers to carry out analytics on their player data. Clients who use the platform have access to several functions which produce analytical charts and forecasting using game data. Some of the functions available are analysis of a user's journey throughout a game, prediction of the amount players will spend in their lifetime and segmentation of players to send targeted offers based on player characteristics.

Whilst deltaDNA have clients from a variety of games platforms, a large majority of the games are free-to-play mobile games. For the purposes of this work, all games analysed in this paper are free-to-play mobile games.

1.2 Scope and Objectives

The main goal of this study was to consider how differing levels of contextual information about a player or a game can affect the accuracy of predictive behavioural modelling. Hence, the study considers three approaches to modelling player retention and churn whilst attempting to understand how player demographics and in-game activity affect prediction of future behaviour.

1.2.1 Retention

The first section of this study concerns predicting player retention. Currently, the retention prediction model implemented on the deltaDNA platform uses optimisation to predict retention based on a period of observed retention values. For a game which has newly launched or which has a small player base, there may be little to no historical data. In this case, it is very difficult to make an accurate prediction of player retention.

This research aims to discover whether it is possible to predict retention using a short period of playing data. By analysing player data for a basket of games from different genres, the objective is to identify whether the shape of the retention curve is individual to each game or if there is a relationship between player demographics and retention.

1.2.2 Churn

The later part of this study focuses on predictive churn modelling. Whilst deltaDNA is primarily a service provider of a platform, the insight department works with some clients to provide more detailed analysis of their games. A frequent request from clients is to produce a churn prediction model to identify those players about to stop playing the game.

Building a churn model for a specific game requires a deep understanding of the structure of the game and often involves time playing the game to understand features which can be used to understand player behaviour. This can be a time-consuming process. However, as with all data mining projects, there is no guarantee of a high accuracy model.

The objective of this part of this study is to produce a general churn prediction model which can be applied to any game. That is the features of the model are independent of the game itself and rely on player behaviour which can be observed in any game. A game agnostic model could act as a baseline to evaluate the accuracy of a game specific model. Also, a model of this type gives insight into which features are important drivers of churn at different stages of player lifetime.

The last objective of this study was to produce a churn model for a specific game as requested by the client. By producing this model, it is possible to compare the performance of the game specific model against the previously discussed game agnostic model. The aim of this comparison is to discover how contextual information about a game impacts the accuracy of predictive churn modelling.

1.3 Achievements

The main objective of this project was to research the performance of predictive behavioural modelling and how it can be influenced by knowledge of contextual information about a player or a game. To achieve this objective required processing and analysing large datasets of data measuring in-game events. Lack of exposure to datasets of this size resulted in a steep learning curve when deciding the most efficient methods of handling and processing data of this size.

To effectively carry out this project, an increased knowledge of multiple technologies was required. For data analysis, this included a deeper knowledge of the programming languages R and Python. Specifically, Python's Pandas and Scikit-learn modules were used for data handing and analysis and data mining respectively throughout the project. The greatest increase in knowledge was required for writing SQL queries to the Vertica database. This included learning about many functionalities available in SQL queries including window functions which were used to create complex queries.

The results from the game specific churn model were presented to the client. The churn model was documented throughout to ensure the client could follow the process and reproduce the model in the future. Further documentation was supplied to ensure the decisions surrounding the approach were clearly explained to enable deeper understanding. Analysis of the model enables the client to understand which features have greatest importance at different stages of the player lifetime. Hence, player management strategies can be implemented to increase player retention.

The project has met the objectives as outlined previously. The results from the research into player retention give the company insight into patterns in player behaviour and their impact on player retention. The results from the churn prediction models provide the company with a baseline churn model which can be used in the future to assess the accuracy of game specific models. Documentation clearly outlines how the models can be replicated to predict churn for any game in the deltaDNA database. Further, the game agnostic nature of the churn model means it can be extended to model game data from any free-to-play mobile game.

1.4 Overview of Dissertation

Chapter 2 - *State of the Art* summarises the background and current literature in predictive retention and churn modelling with a specific focus on literature using datasets from the games industry. A technical outline of the data mining methods which will be referenced throughout the remaining chapters of the study is provided. Finally, the accuracy measures referenced throughout this study are outlined.

Chapter 3 – *Data Acquisition and Access* provides background information about the acquisition, storage and access to the data discussed in this project.

Chapter 4 - *Player Retention Prediction* considers how knowledge of player demographics impacts the prediction of player retention. Visualisations of player retention are created to display the retention curves based on the value of player metrics. Finally, the influence of contextual knowledge about a player is evaluated for prediction of player retention.

Chapter 5 – *A Game Agnostic Approach to Churn Modelling* considers an approach to predicting player churn when there is no contextual information about a game. Model features which can be measured in across any free-to-play game are outlined. Two approaches to predicting player churn using linear regression and binary classification techniques are evaluated.

Chapter 6 – *A Game Specific Approach to Churn Model* outlines a churn model created for a deltaDNA client. Background information about the game is provided. The performance of this game specific model is compared against the game agnostic model outlined in Chapter 5.

Chapter 7 – *Conclusion* will provide a summary of the study. Finally, the work of the project will be evaluated followed by an outline of some possible approaches for future work.

2 State-of-The-Art

2.1 Literature Review

Data mining is the attempt to discover patterns in large datasets which can be used to give meaningful insight into some process. For large datasets, it can be difficult to define specific rules or relationships in the data, therefore, machine learning techniques can be used to unearth these unseen relationships. Predictive models, such as regression, can be used to predict the value of an object. Whereas, classification techniques attempt to predict the class or label of an object.

2.1.1 Behaviour Modelling

Understanding customer behaviour and the reason customers churn is important to businesses in any industry. The use of predictive behavioural models enables businesses to act when a customer is about to churn which can have great impact on the customer base and, in turn, the businesses revenue. The relationship between churn and retention means that results from one model can be used to estimate the other, therefore, much of the research in this area focuses on predicting customer churn.

Whist most research into player churn is relatively recent in the games industry, there is a long history of churn analysis in other applications. One of the earliest areas of research was the telecommunications industry [1] which has since produced a vast amount of research [14] - [19]. However, there is research across a variety of applications including retail banking [20][21][22], insurance [23], subscription services[24], online advertisements [25] and community based answering platforms [26].

Generally, churn modelling is approached using a variety of classification techniques including decision tress, logistic regression, neural networks, and support vector machines.

[1] is one of the earliest works in customer prediction in the wireless telecommunications industry which compares the performance of logistic regression, decision trees and a neural network approach on a dataset of customers from the USA. It reports that neural network performed the best in this study. This result is found in literature focused around subscription to cellular wireless services [19]. [14] studies wireless communication data from a Korean company and reports logistic regression as the superior method. [16] compares the performance of support vector machines against other classification techniques and results show that it performs equally as well as other techniques.

There are differing results across industries. [24] reports random forest as the superior method in a newspaper subscription service, whilst [27] outlines a decision tree as the best technique to determine churn using customer complaints data. The work in [12] reports logistic regression as the most accurate method on retail banking data from a Finnish bank and [27] reporting random forest as the most accurate method on community based question answering dataset.

Clearly, there is no superior method across industries or even within one research area. Hence, the best approach taken is determined by the dataset itself. The complexity of neural networks may provide advantageous performance in some areas, however, for simplicity of understanding and implementation, decision trees and regression may be the preferred methods.

2.1.2 Games Industry

Research in the games industry is increasing as data becomes more accessible. There is research in a variety of areas including churn modelling[28][29][30] predicting player purchasing decisions [31], modelling player interest [32][33] and the impact of social influence on player behaviour [34].

However, one of the greatest causes of the lack of research in the games industry is lack of data available to researchers. Commercially, it is in a company's best interest to ensure their data is kept private. In terms of research, there are limited papers which use datasets consisting of a variety of games because there are few companies willing to share this kind of information or developers only have a handful of similar games. Therefore, majority of research focuses around the dataset of a single game [29][35][36][37]. There are exceptions to this case which involve research across multiple games. [38] explores how advertising affects player retention across 20 mobile games.

Much of the research involving datasets of multiple games focuses on discovering underlying distributions in metrics. [39] uses lifetime analysis techniques to fit distributions which determine how a player's interest decreases over time. Similarly, [32] studies the distribution of playtime in a selection of 3000 PC and console games from the Steam platform. Both [39] and [32] determine that playtime follows a Weibull distribution. Similarly, [40] and [41] found that session length and inter arrival time fit Weibull distributions for datasets across multiple games. Lastly, [33] determined that the total days played was best fit by a Weibull distribution for five mobile games.

There are two distinct types of games which have produced the greatest amount of research: free-to-play games and Massive Multiplayer Online Role-Playing Games (MMORPGs). There is a large amount of research in MMORPGs which may be due to the ability to infer properties of a system by probing-based techniques rather than accessing logs of player behaviour from the companies themselves. MMORPGS tend to have a large population of users across multiple servers, therefore, often the research focuses on network balancing [41].

2.1.3 Churn Modelling in Games

Churn modelling is applicable across all games; however, the definition of churn is different for each game. [36] identifies a customer who has not played for a window of time as churned, in this research they use a churn window of 4 weeks. Whereas, [42] which uses a dataset of subscribed players determines a player as churned when they cancel their subscription.

For free-to-play games there is a short player lifetime, therefore, churn measurements tend to be evaluated across shorter windows.[29] uses binary classification, where a player with no activity in the second week is labelled as churned. Similarly, [30] labels players with no activity for 14 days as churned. [28] approaches churn in two ways: hard churn if a player has no activity after a cut-off date and soft churn where the player may have a low number of sessions in a short window after the cut-off date.

As the measurement of churn is different across games, the classification method and features selected vary for each model. For the MMORPG Everquest II, [34] examines the effect of negative and positive social influence in churn prediction whilst [42] uses a player lifecycle based approach to which proposes three dimensions of player behaviour as engagement, enthusiasm, and persistence. [36] compares the performance of Hidden Markov models against other machine learning techniques for the MMORPG Destiny.

Previous research in churn prediction for free-to-play games is generally applicable in a single setting and cannot be transferred across games. Exceptions to this include [28] which is the most notable piece of research which approaches modelling in a game agnostic manner. This was the first approach to churn modelling which used game agnostic features in binary classification. The features include temporal features such as number of sessions, more complex features such as the parameters of a power law fitted to playtime history and features relating to virtual economy. This paper considers classifiers such as neural networks, logistic regression, naïve Bayes, and decision trees with decision trees claiming the best performance.

[29] outlines an approach to churn prediction for a single game, however, many game agnostic features were included in the model. Features including geographic location, average session duration and total play time are game agnostic, whereas, round specific features such as average moves and stars are specific to the game itself.

2.2 Methodology

2.2.1 Nonlinear Least Squares

Regression models are built to understand the relationship between input and output values. Nonlinear regression is a form of regression which fits a nonlinear model to a set of observations. Assuming there are many data points of the form (x_i, y_i) and the function of the nonlinear model takes the form $y = f(x, \beta)$, the model depends a vector of parameters β. Therefore, the aim of nonlinear regression is to find the values of the vector β where the model best fits the observed data.

Nonlinear least squares regression attempts to find parameters of a curve such that the curve is the best fit to the observed data, hence, the sum of squared errors between the observed values and the curve are minimised. Hence, the aim is to minimise

$$S = \sum_{i=0}^{n} (y_i - f(x_i, \beta))^2 . \tag{1}$$

The minimum of S occurs when the gradient is zero. To solve this requires solving the derivative of S with respect to each of the parameters:

$$\frac{\partial S}{\partial \beta_j} = 2\sum_{i=0}^{n} r_i \frac{\partial r_i}{\partial \beta_j} . \tag{2}$$

However, the derivatives $\frac{\partial r_i}{\partial \beta_j}$ are determined by the value of x_i and the parameters so there is no closed solution to this system of equations. Instead, a set of start values are defined for each of the parameters. Then small changes are made to the values of the parameters iteratively until the values converge at an approximate solution to the problem. The details of the process used to achieve an approximate solution are outside the scope of this project.

In this study, nonlinear least squares regression is carried out using the package NLS in R. The Port algorithm, as documented in the Port library[8], was chosen as it is the only algorithm which allows upper and lower bounds to the value of the parameters.

2.2.2 Huber Regressor

Linear regression models are built from the data to understand the linear relationship between input and output variables. In this paper, linear regression will be used to predict the number of subsequent days a player will play a game given the values of multiple variables.

Simple linear regression attempts to finds a linear function that predicts a dependent variable as a function of an independent variable. Given a dependent variable Y and an independent variable X, the equation which represents the relationship between these variables is represented by

$$Y_i = \alpha + \beta X_i + \varepsilon_i. \tag{3}$$

The aim is to find the equation of the straight line represented by

$$Y = \alpha + \beta X, \tag{4}$$

where α is the vertical offset and β is the slope of the regression line. Simple linear regression can be extended to multiple linear regression which predicts the value of a dependent variable as a function of multiple independent variables. The most common method of linear regression is Ordinary Least Squares which estimates the regression line by minimising the sum of squared differences between the observed values and those predicted by the function. However, least squares regression methods are highly sensitive to outliers in the data.

Outliers in game telemetry data can be caused by many issues. Outliers can be incorrect measurements caused by errors in data transmission, fraudulent behaviour, or flaws in the measurements themselves. However, outliers are also be caused by the natural deviations in player behaviour. Therefore, to reduce the impact of outliers on the calculation of the regression function, an approach to regression which is robust to outliers is implemented.

Huber loss is a loss function used in regression which is robust to outliers. A loss function quantifies the amount of error between predicted and observed values. Therefore, the aim of regression is to minimise the loss function. [9] defines the Huber loss function as:

$$L_\delta(y, f(x)) = \begin{cases} (y - f(x))^2 & \textit{for } |(y - f(x))| \le \delta, \\ 2\delta\,|(y - f(x))| - \delta^2, & \textit{otherwise}, \end{cases} \tag{5}$$

where f(x) is the predicted value and y is the observed value. Huber loss function treats instances of data as either inliers or outliers where the absolute error $|(y - f(x))|$ is greater than a threshold determined by δ. Huber loss applies a squared-error loss to inliers; however, absolute loss error is applied to outliers.

In this study, Python's Scikit Learn module is used to perform Huber loss regression. The parameter δ controls how many samples are classed as outliers and it is set at 1.35 to achieve 95% statistical accuracy [9]. A set of training data determines the coefficients of the model.

Overfitting of the model means the model cannot make accurate predictions on unseen data, therefore, a regularisation parameter can be altered to control the complexity of the model.

2.2.3 Decision Tree

A decision tree is a data mining technique which produces a tree-like graph with a set of branching decisions that end in a classification. Each node in a decision tree represents an input variable whilst the branches leading from that node represent the values the variable may take.

Classification of single object begins at the top node of the tree and follows the branch which corresponds to a given value of that variable. The process is repeated on each branch until all instances at a branch have the same classification value. When all instances on a branch have the same value, the leaf of the tree is reached.

The order in which variables appear in the tree is important to optimise the number of correct classifications. There are several algorithms which determine the structure of a decision tree. This paper implements the J48 decision tree in Weka which implements the ID3 algorithm.

The ID3 algorithm selects each node of the tree by calculating which variable provides the greatest information gain of the output classification. This measures how much the uncertainty of classification is reduced by splitting the data on that specific attribute.

To determine the information gain from splitting on a variable, the information of a value of that variable is calculated. The information provided by a single value of a variable, where p_e is the probability of the event occurring, is measured by

$$I(e) = -\log(p_e)\, I(e) = -\log(p_e). \tag{6}$$

Using, the weighted average information across all possible values of a variable is calculated. This is called Entropy which measures uncertainty. Entropy is calculated by summing the probability of each possible value multiplied by its information value from above as follows

$$H(x) = -\sum_{i=0}^{n} P(x_i)\log(P(x_i))\,.H(x) = -\sum P(x_i)\log(P(x_i)) \tag{7}$$

Finally, information gain is a measure how much uncertainty is reduced by splitting the data on a specific attribute. Therefore, it is calculated by the difference between the entropy of an outcome prior to splitting and the entropy given that the data is split on an input variable. Using the equation for entropy the information gain is calculated by

$$Information\ Gain = H(outcome) - H(outcome \mid input)\,. \tag{8}$$

ID3 calculates the information gain for each input variable and selects the one with maximum information gain to split on at that node. Branches are created for each value the variable can take. This process is repeated for the subset of data defined by each branch of a node until all objects at the current leaf are of the same classification value.

A hyperparameter which can be varied for this technique is the minimum number of objects at a leaf node. This number can be increased to avoid branching decisions which are very specific to one instance. In doing so this simplifies the tree and reduces the risk of overfitting to the training data.

Decision tree models were produced using the data mining software Weka as this it offers simple visualisation of the nodes of the tree and there was existing knowledge of this package.

2.2.4 Measurement of Model Accuracy

A classification model attempts to predict the class of a value or instance. Classification is often treated as a binary decision, which is a value that is classed as either 0 or 1. This can be used to classify values as 0 or 1 or, in the case of this paper, no churn and churn respectively. The output values of a binary classification model can take one of four values which are outlined in Table 1. There are many methods which use the counts of these classification outputs to measure the performance of a classification model.

Table 1: Output Values of a Binary Classification.

Result	**Actual Value**	**Classified Value**
True Positive (TP)	TRUE	TRUE
False Positive (FP)	FALSE	TRUE
True Negative (TN)	FALSE	FALSE
False Negative (FN)	TRUE	FALSE

Confusion Matrix

A confusion matrix is calculated to display the number of instances of each output result. The columns of the confusion matrix display the number of instances predicted in each class and the rows display the number of observed instances in each class. Therefore, the cell in the 'Negative' column and the 'Positive' row contains the number of observed instances which are positive but have been classified as negative. The following is a binary confusion matrix:

		Predicted Class	
		Negative	Positive
True Class	Negative	TN	FP
	Positive	FN	TP

Interpreting a confusion matrix gives insight into where a classification model performs well. The importance of the number of false positives and false negatives can vary depending on the use case of a model.

For example, a high number of false positives in a game churn model could result in many players being shown an offer to encourage them to continue playing. While this means an offer is shown unnecessarily, it is unlikely to have any major impact on a non-churners game play. However, if the churn model has a high rate of false negatives, then it fails to identify all players about to churn and the model does not do as intended.

Accuracy

The accuracy rate of a model can be determined from the values displayed in the confusion matrix. Accuracy is the number of instances which are correctly identified divided by the total number of instances, calculated by the following formula:

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP}. \tag{9}$$

The accuracy rate calculates the percentage of instances in the test set of data which are correctly classified. Hence, a churn model which classifies 80 players as either churn or no churn correctly out of a test set of 100 players has an accuracy rate of 80%.

It is important to note that a high accuracy rate does not necessarily signify a good model. For example, of the 100 players, 20 might be about to churn. The model could identify all 100 players as non-churners which produces the accuracy rate of 80%. Therefore, it is important to consider the confusion matrix and following measurements to evaluate the performance of a classification model.

Precision

Precision, also known as positive predictive value, is a calculation of the number of correctly identified instances out of the retrieved instances. For a churn model, this is considered as the number correctly identified as about to churn out of those predicted as churn.

A high precision rate is a sign of a low number of false positives whereas a low precision rate means a high number of false positives. Hence, precision measures the ability to predict positive instances correctly, or the ability to not predict a negative instance as positive. The formula to calculate precision is as follows:

$$\Pr ecision = \frac{TP}{TP + FP}. \tag{10}$$

Recall

Recall, also known as sensitivity, is a measure of the completeness of a model. That is the number of relevant instances identified. In terms of churn, this is a measure of the number of correctly identified churn days out of those who will churn.

A high recall rate is a sign of few false negatives whilst a low recall rate means there are a high number of false negatives. Hence, recall measures the ability to predict a positive instance correctly, or the ability to not predict a positive instance as negative. The formula to calculate recall is as follows:

$$\mathrm{Re}call = \frac{TP}{TP+FN}. \tag{11}$$

3 Data Acquisition and Access

3.1 Data Acquisition

When a user is playing a game, several events occur which make up their game play. The process of a player completing a level within a game can be broken down into events. A breakdown of the basic events which occur when a player completes a mission are shown in Figure 1: Diagram of the events which occur when a player completes a mission in a game. The complexity and structure of a game determine the number and variety of events which can occur within a game.

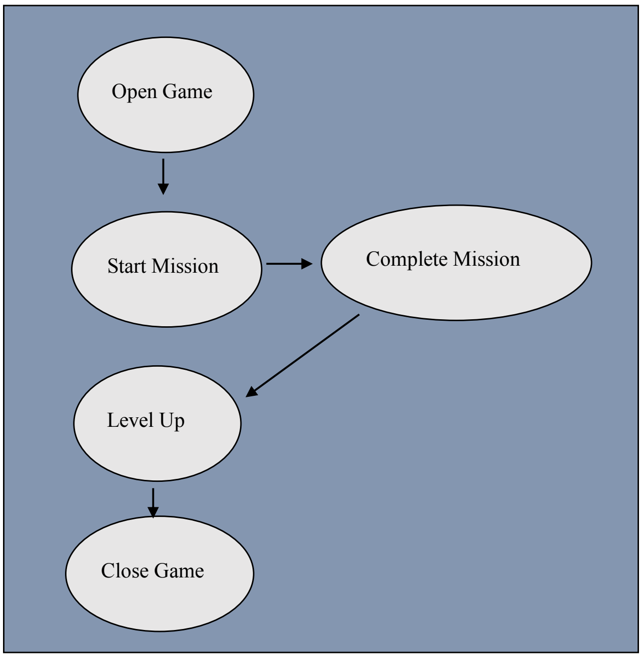

Figure 1: Diagram of the events which occur when a player completes a mission in a game.

This event level data is the data which is sent from games to the deltaDNA database. When a game is developed clients can include a software development kit (SDK) which enables a game to record in-game events and upload them to the database. Each event records information such as the user ID, game ID, the event timestamp, and the event name. The event names, for example 'mission started', are determined by the structure of the game and the events which occur within it.

There are several default events which are collected for all games such as new player or game started. Event level information is stored in the event store table. Similarly, player information such as age, gender, platform, user country and acquisition channel are collected where possible for all games. This information about players is stored in the player metric table of the database. Whilst there are tables of further information in the deltaDNA database, the data required for this project was taken from the event store and player metric tables.

When an event occurs, the SDK sends the event data in a JSON format to the deltaDNA database. A Vertica database is used which has a distributed architecture which offers scalability and fast completion of queries.

3.2 Accessing Data

Data was extracted from the Vertica database by using the Vertica Python module available in Python. This offers direct connection to the database and queries are written to the database in SQL script. Using the Vertica Python module means that data is retrieved and can be processed in python.

Much of the following analysis involves processing data from multiple games. To query multiple games at once, python code which runs a threaded set of queries was used to query the tables of up to 10 games at once.

Results of the database queries were stored in Python's Pandas to ensure easy analysis and handling of large sets of data. Also, the output from queries was written to a csv file using Python's CSV module to enable data to be analysed in both R and Weka.

4 Player Retention Prediction

One of the key measurements for game developers is player retention. The player retention rate measures the percentage of players from a cohort who are active at some day after install. Hence, 30-day retention measures the number of players who remain active out of a cohort of players who installed 30 days previously.

Measuring retention enables forecasting, for example, a simple measurement of the value a player will generate in their lifetime can be calculated by multiplying retention and the average spend per player. Modelling player retention gives further insight into player behaviour and allows developers to understand how to encourage players to play for longer.

The current retention model available on the deltaDNA platform uses a period of historical retention values and optimisation to estimate the future retention rates. To ensure an accurate prediction, the model requires playing data for a cohort of at least 100 players [10]. Otherwise, the model is based on default parameters and the retention will be estimated to fall between a wide interval of values. The graph in Figure 2 is displays the retention predictor available on the deltaDNA platform. It shows the estimated bounds of retention prediction where there is not sufficient data in the sample.

Figure 2: The retention prediction model as displayed on the deltaDNA platform when attempting to optimise a sample of 20 or less players [10].

The default retention estimation can fall between a wide range of values. Therefore, the aim of this section is to discover if segmenting players into bins can produce a more accurate prediction of future retention rates. In other words, the aim is to discover whether the shape of the retention curve is related to many player metrics and independent of the game or whether the shape of the retention curve is game specific.

For a game which is new to the market or which has a small player base, it is difficult to predict future retention rates. Ideally, a model would be able to predict retention based on retention values for a short period following player install. This would also be beneficial for a game which has seen significant changes to the structure as a short period of observations would be required to update the estimation of player retention.

Therefore, the approach to this research has several steps. Initially, player metrics are outlined and players are split into segments according to their values for each metric. The retention rates for each segment are visualised to display any patterns in the retention curve based on the values of these player metrics. Lastly, we compare the accuracy of predicting player retention for each segment against a single model for all players.

4.1 Data and Pre-processing

The aim is to predict retention up to day 180. For a period of this length, a large sample is required to ensure there are instances of players who are active for the full period of 180 days. There are many measurements about a game which give insight into the size of the player base. The average number of daily active users and average number of daily installs are sufficient measurements to determine the size of a game. Hence, games are restricted to those with at least 1000 daily installs and 10,000 daily active users.

The aim is to discover if there are any patterns in retention across games, therefore, analysis is carried out on a basket of 10 games of differing genres and types. The 10 chosen games are all free-to-play mobile games.

The sample of players are measured as those who installed in the month between dates 1/9/16 to 1/10/16. For each day since install, the number of players active on that day is counted. Therefore, on day 0 since install, the count will be the total number of players in the sample. Day 0 is the equivalent of the install day for all players. The retention rate is calculated measuring the number of active players on that day divided by the number of players in that sample. Therefore, retention on day 0 is 100%.

4.2 Player Metrics

The aim is to segment players into bins according to the value they take for a variety of metrics. Ideally, each player bin would have at least 1000 players in the sample to ensure accurate modelling. Therefore, variables which have are distributed across the possible values are necessary.

There are many player characteristics which could be considered including user age and gender. However, Figure 3 displays the uneven distributions of players by gender across the genres of games on the deltaDNA platform. Instead, to ensure there are sufficient players in each bin and a manageable number of bins, metrics which can take many values are collated into groups.

GENRE	MALE	FEMALE
Action	75%	25%
Strategy	68%	32%
Puzzle	18%	82%
Social Casino	39%	61%

deltaDNA

Figure 3: Gender splits across the game genres on deltaDNA platform [11]

The first player metric considered is the player country. Whilst the data from most countries is reliable, there are issues in some countries surrounding hacking games and fake devices. For this reason, data is sampled from countries which are known to have reliable data and a large player base. The following values are considered for user country: English Speaking countries (GB, IE, US, CA, AU, NZ), Western European countries (FR, ES, PT, IT, DE, NL, AT) and Latin American countries (BR, AR, GT, CL, EC, PE, UY, VE).

The second player metric considered is the platform device. Platform measures the brand and type of device a player is using. For example, Android Tablet and Android Phone are stored as two separate devices. To limit the number of bins, the platform measurement concerns the brand of device being used not the handset type. Also, the platform measurement is limited to the two brands with biggest player base to ensure each bin has a sufficient number of players. The possible values for player platform are Android and IOS.

Finally, the aim was to include features which capture length of play and frequency of play on the first day of play. The proposed metrics to measure this were play time and number of missions completed. However, these values are highly correlated, which means a player who has completed a low number of missions is unlikely to have a high play time.

To avoid the issues caused by correlated features, play time on install day was the only measure chosen to capture player behaviour. Since players are segmented based on their value for play time it is impossible to measure individual play time. Instead, players are split based on whether their play time is above the median value for that game (high play time) or below the median (low play time). For some games, play time is a measurement of how long the game is open, therefore, the value of play time can take extreme values. To remove outlier values, any measurement of play time per day greater than 4 hours is reduced 4 hours.

Efforts were made to ensure that all player bins had a sufficient sample of players, however, is it unavoidable that for some games there were either no players (game released on one platform) or the sample of players was very small. These player bins were not considered in the analysis.

An outline of the final features and the values they can take are outlined in Table 2: . The final product of this is 12 player segments, one for each combination of the feature values.

Table 2: Player metrics and the values they can take used to segment players.

Feature	Values
Platform	Android, IOS
Country	English Speaking, Western European, Latin America
Play time on Day 0	0.5 (50^{th} percentile), 1 (over 50^{th} percentile)

4.3 Visualising Features

To compare the impact of the value a player takes for each of the player metrics, the retention curves are visualised. For each of the 12 player bins, the number of players on each day is aggregated across all sampled games. The retention rates are then calculated on each day by dividing by the number of players in that segment on day 0.

Figure 4 displays the retention curves for each player segment from install to day 30. Clearly, there is a significant difference in the retention values dependent on the time spent playing on install day. Following day 0, there is a steep drop in the retention rate for players with low game time. Across the 30 days, it is apparent that the rate of retention is higher for those who play for a longer period than the median value. As the number of days since install increases, player retention tends towards 0.

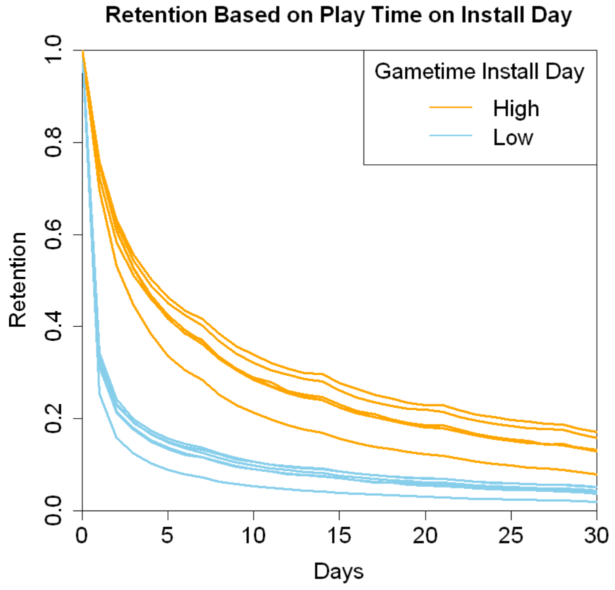

Figure 4: Graph of the retention curves for 12 player segments. The colour of the curve outlines whether players in the segment have high or low game time on install day.

Figure 5 and Figure 6 display the retention curves for each segment where the platform and country are highlighted respectively. As previously discovered, the feature play time has great influence over the retention rate, therefore, the curves are split into two clusters: low and high play time. Within each cluster there is a pattern based on platform and user country.

For users of the same play time value, players on Android devices generally appear to have lower retention rates than those on IOS devices. There are similar patterns in the user country metric. The retention curves of players from Western European and English-Speaking countries decline at a more gradual rate, whereas, the retention rates for players from Latin American counties are generally lower for players of the same value for game time.

Visualising the retention curves for each player segment gives insight into the relationship between player metric values and the slope of the retention curve. There is some variation in the retention curves for players from different platforms and countries. However, the most revealing player metric is the amount of time played on the day of install. This suggests that players who play for less time on the day of install are likely to churn before those who play for longer on install day. This makes sense intuitively as a player who shows less interest on install day is likely to lose interest in the game at a faster rate.

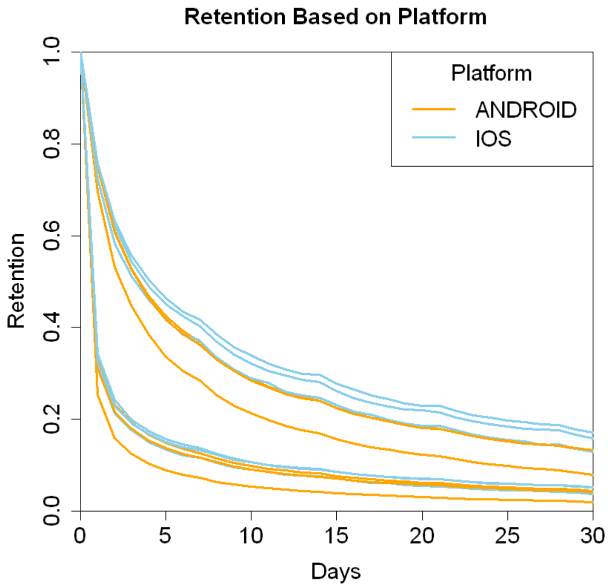

Figure 5: Graph of the retention curves for 12 player segments. The colour of the curves are determined by the value of player platform type.

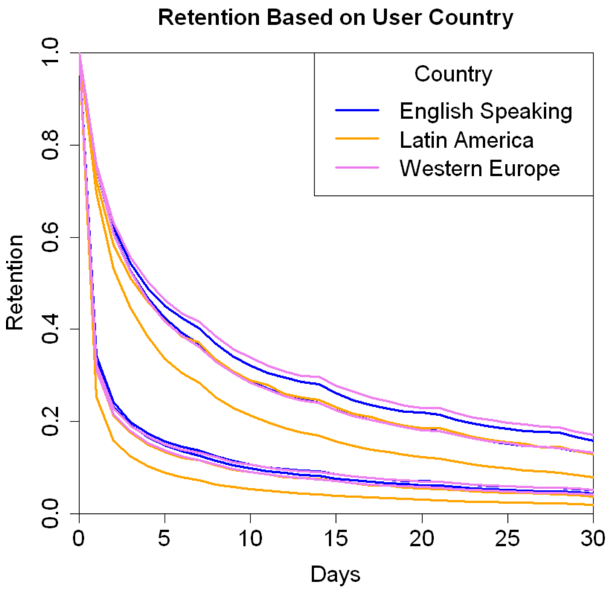

Figure 6: Graph of the retention curves for 12 player segments. The value of the user country metric determines the colour of the curves.

4.4 Modelling

The aim is to discover whether contextual information about a player affects the accuracy of prediction of retention. Models are fit using a retention values for a short window after install. Hence, it is evaluated whether modelling players as a single group or modelling individual player segments produces a more accurate prediction of retention.

4.4.1 Distribution of Retention Curve

Prediction of future retention involves fitting the retention curve to a set of observed values. To do so, the distribution of the retention curve must be identified. The literature review in Section 2.1.2 discusses several papers which find aspects of player behaviour follow a Weibull distribution. [32] determined that the number of days played follows a Weibull distribution. Therefore, the cumulative density function of a Weibull distribution can be used to determine the probability that a player plays less than a specified number of days. [12] defines the cumulative density function of a Weibull distribution is as follows:

$$F(x) = P(X \leq x) = 1 - \exp(-\beta\, x^{\alpha}), \qquad x \geq 0, \alpha > 0. \tag{12}$$

The complementary cumulative density functions of a distribution are called survival functions. A survival function measures the probability that a value is greater than x. Hence, the survival function of a Weibull distribution has the following formula:

$$S(x) = P(X > x) = \exp(-\beta\, x^{\alpha}), \qquad x \geq 0, \alpha > 0. \tag{13}$$

Assuming the total number of days a player plays follows a Weibull distribution; the survival function measures the probability that a player plays more than x days. In other words, the survival function measures the probability that a player is retained on day x which can calculated using the retention rate. The survival function of a Weibull distribution is often referred to as a Stretched Exponential function.

4.4.2 Predicting Retention

To predict the future retention rates for a cohort of players, the parameters of the Stretched Exponential function must be estimated. Using the nonlinear least squares method outlined in Section 2.2.1, the parameters of the distribution are estimated based on the retention values on days 0 to 7. The retention values can then be estimated up to day 180.

A single set of retention values is calculated for all players in the sample for days 0 to 180. The single model is fitted using the first 7 days of data and predicts the values of player retention for days 8 to 180.

To build the segmented model, retention values are calculated for each of the 12 player segments. For each segment, the parameters of the retention curve are estimated based on player retention across days 0 to 7. For each segment, the predicted retention curves for days 8 to 180 are generated. The average value across all segments is calculated to compare the accuracy of the model against the single model values.

4.5 Results

The graph in Figure 7 displays the actual player retention curve against the predicted player retention curve based on modelling all players in a single model. The predicted model appears to be a good fit in the short period following install. However, predicted retention curve underestimates the retention rate.

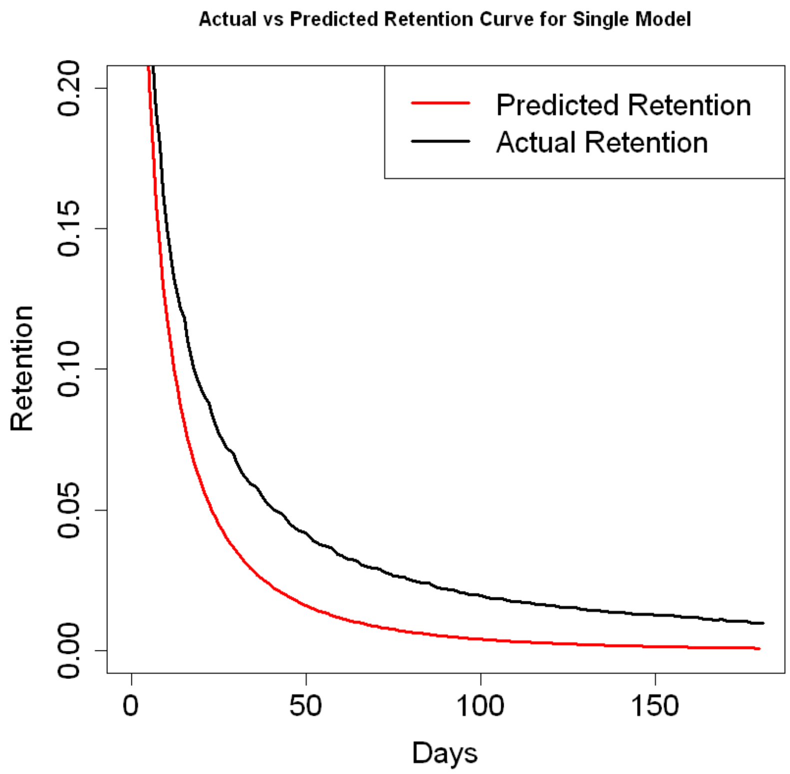

Figure 7: The true and predicted player retention curves based on a cohort of players from a basket of 10 games. Predicted curve is fit using 7 days of retention rates.

Similarly, the predicted retention curves for each player segment can be seen in Figure 8. Again, the models fail to estimate the values of retention correctly and the models appear to underestimate the number of number of players retained.

Figure 9 displays the actual retention curve against the predicted retention curves for both the single model and the average of the segment model. The average of the predicted retention curves of each segment appears to generate a more accurate prediction of retention up to approximately 50 days after install. After this point, both the single model and the average of the segment models fail to predict the true values of player retention.

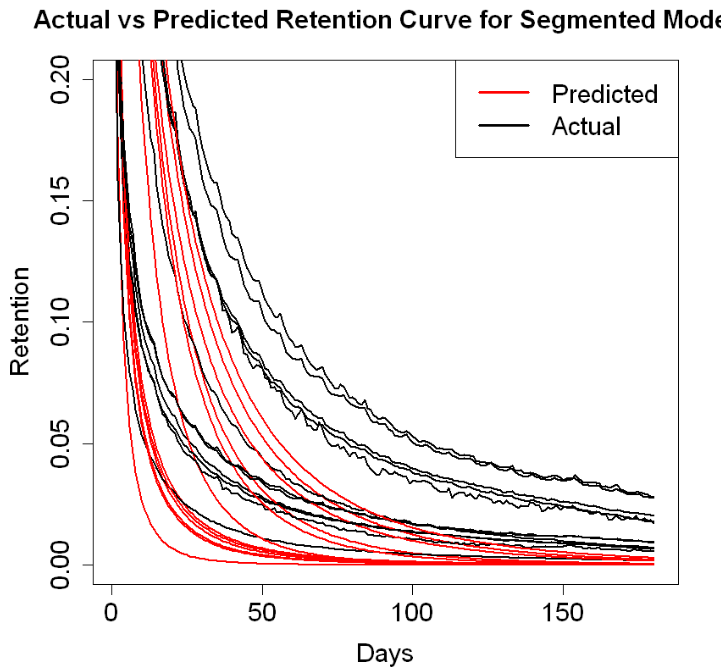

Figure 8: The true and predicted player retention curves for each player segment across a basket of 10 games. Predicted curves are fit using retention values day 0 to 7.

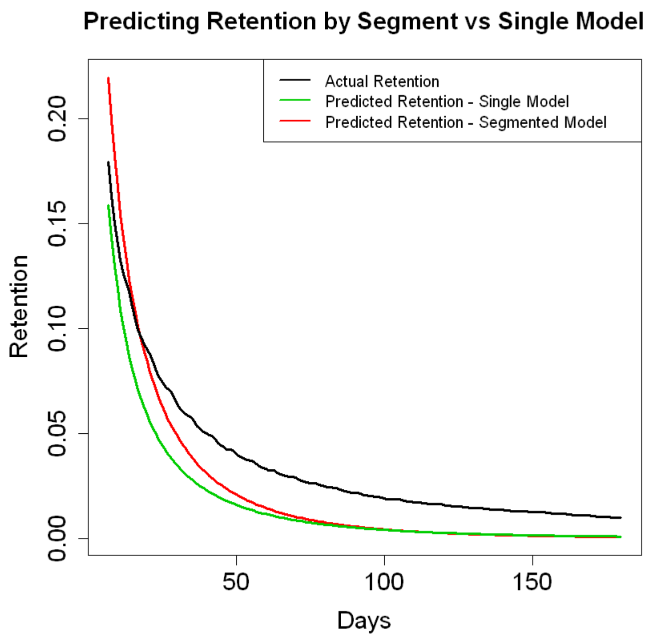

Figure 9: The true and predicted player retention curves for a single model and average of segment models. Predicted curves are fit using 7 days of retention rates.

4.6 Evaluation & Next Steps

The results for both a single model and the average of the segment models fail to estimate player retention correctly. However, of the two models, fitting the models for each segment then taking the average of the predicted values appears to produce the best result.

In both cases, the models under predict the values of retention. This could be due to the short period of observations modelled, hence, models fit to retention data across a longer period would likely produce better results. On day 0, 100% of players are retained; however, there is a significant decrease in the number of active players by day 1. Therefore, fitting the model based on the retention values from day 1 to 7 is likely to produce a more accurate result.

Of the considered player metrics, the amount of time spent playing was clearly the most important feature in determining player retention. Ideally, more granular information about the time spent playing and other behavioural measures would be considered to generate a more accurate prediction. However, introducing more metrics and a range of values for each metric would result in a larger number of segments. Segmenting players into a greater number of bins reduces the likelihood of a sufficient sample of players in each segment. Instead, to model player behaviour at a more detailed level, the remaining chapters of this study focus on the task of modelling when a player will churn based on player behaviour measures.

5 A Game Agnostic Approach to Churn Modelling

The remaining chapters of this study will focus on modelling player churn. A player is said to have churned when they have stopped playing a game, hence, they are no longer retained. The nature of free-to-play games means players are likely to churn after a short period following install. Analysis of the data from Section 4 determined that the average player retention on day 10 across the sample games was 17%. Therefore, over 80% of players who installed in the sample period were no longer actively playing 10 days after install. Hence, in many cases the player lifetime is a short period of time.

Modelling player churn has two main benefits. Firstly, modelling player behaviour in the period prior to churning means measures can be put in place to attempt to reduce the number of churners. As outlined in Section 1.1 there are many approaches such as offering deals to purchase virtual currency and rewards for completing missions which can be employed to encourage players to continue playing a game. However, for these methods to be a success, it is important to catch players before they churn rather than after the event. Secondly, churn models can give an insight into the most important features in driving churn. These results can be used to adapt elements of a game to encourage longer player lifetimes.

5.1 Game Agnostic Approach

As discussed in Section 1.2.2 building a churn model for a specific game can be time consuming with no guarantee of a good result. Other issues arise for games which are new to the market if they have little historical data. In this case, it is difficult to train a churn model if there are few instances of players churning. Similarly, if there are updates or changes to a game then a new model would need to be trained on updated data.

The objective is to produce a predictive churn model which can be applicable to any game. However, there are many genres of games which are all structured differently. For example, an action game might feature progression through levels and register an event when a player completes a mission. Whereas, a puzzle game might record events such as each time a player requires a hint.

Game developers using the deltaDNA platform can decide which events to store and the name they are given. Therefore, games which have identical events may store these under different names. The lack of uniformity in the naming of events means that modelling specific events is difficult.

To overcome these issues, a game agnostic churn model is proposed. This is defined as a model which can predict player churn without having underlying knowledge about the specific game. In this case, the model does not depend on contextual information about the game. Instead, features are chosen which measure elements of player behaviour that can be observed in any game.

5.2 Method

Creating a player churn model requires defining the meaning of churn and selecting a modelling technique. Section 2.1.3 contains a discussion of the approaches taken to churn modelling in the games industry and other industries.

Of the literature reviewed, there is many classification techniques considered such as decision trees, logistic regression, and support vector machines to model player churn. These methods use binary classification to label a player as churn or no churn. Similarly, there is no definitive measurement of churn in the literature. [36] classes a player as churned if they have a period of 14 days of no activity. Whereas, [42] consider a player to have churned if there is no activity in the second week after install.

Player behavioural features are measured for each active player day. Therefore, the aim is to predict if a day will be the last day a player is active. Hence, the models attempt to predict which days are churn days (when the number of subsequent days left is zero). Churn is defined as the last day a player is active.

For each player, the total number of days played is calculated, that is the number of active days between install and the last observed day. Therefore, for each day a player is active, the number of subsequent days left to play is calculated by the total number of days minus the number of active days to date. An active player day is a day in which an event occurs, whereas, a real player day is measured as a calendar day with no guarantee the user will be active. A visual of how the number of the lifetime of a player based on active days is outlined in Figure 10.

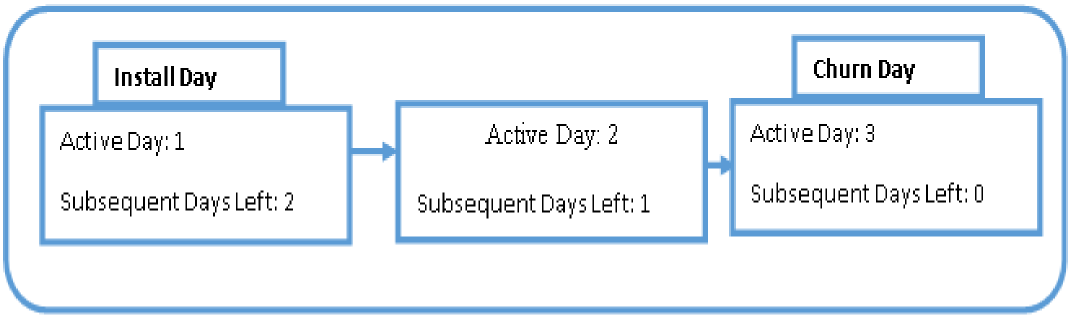

Figure 10: Player activity for 2 days following install.

Player behaviour is observed for 30 days following install, however, some players in the sample play for more than 30 days. However, by the definition of a churn day outlined previously, the final day of observed activity (30 days after install) would be assumed to be a churn day. Figure 11 displays two possible scenarios of player lifetime. In scenario 1 the players last active day is within the observed period; therefore, their churn day is identified correctly. Scenario 2 shows a player who plays for longer than the observation period. In this case, their assumed churn day would be the last observed day in the window (the purple circle).

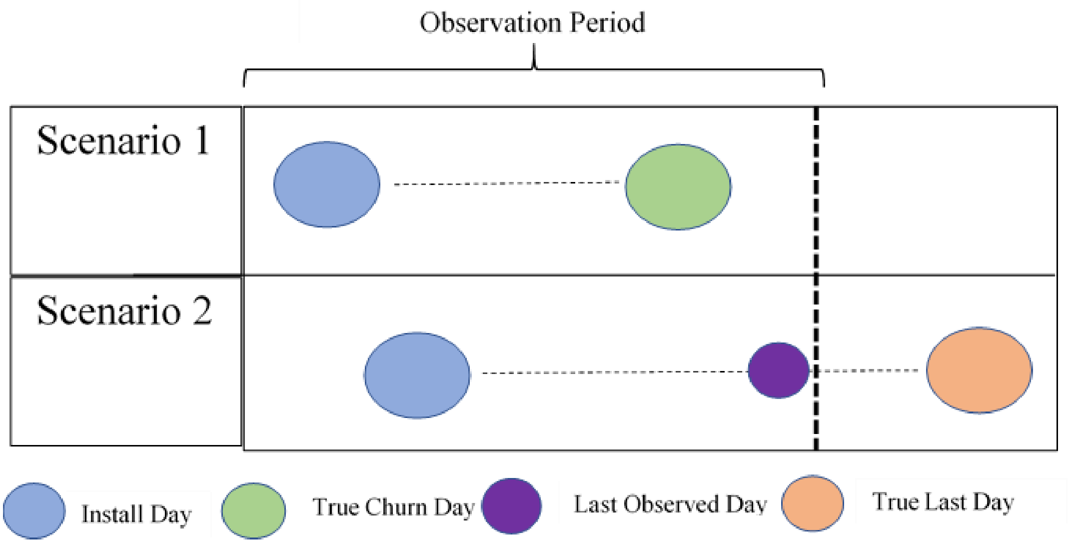

Figure 11: Scenarios outlining two possible player lifetimes across an observation period.

To avoid falsely identifying a player as about to churn, player activity is observed for a maximum of 30 days, however, only the first 20 days after install are modelled. This reduces the risk of falsely labelling a player day as a churn day when they are active for more than 30 days. Also, since the average player lifetime is short in free-to-play games, the most important days to model are those which follow closely after a player installs.

For the considered techniques, the dataset consists of behavioural data measured on each active player day. Instead of generating a single model across the entire playing lifetime, data is split on the number of real days since install. The aim of this is to give insight into which features are important in driving churn at different stages of the player lifetime. Therefore, the dataset is split into 6 sets: install day, 1 to 2 days since install, 3 to 5 days since install, 6 to 10 days since install, 11 to 15 days since install and 16 to 20 days since install. Each of the datasets is split into 60% which is used to train the model and 40% which is set aside to test the performance of the final model.

In all papers researched, churn modelling was approached with binary classification techniques. Two approaches are considered to identify which player days are a churn day. These are a Decision Tree and a Huber Loss Regression method.

The first approach considers a classic classification technique of a Decision Tree which creates branching decisions that result classification of a churn day or no churn day. The second method considered attempts to predict the number of days left to play using Huber Loss Regression and if the predicted number of subsequent days is below a cut-off value, that day is classified as a churn day.

5.2.1 Decision Tree

The Decision Tree approach is a standard binary classification which creates branching decisions that result in a classification. The chosen Decision Tree and algorithm is outlined in Section 2.2.3. The data mining software Weka was chosen to build decision tree models. The data was collected from the database and split based on the number of days since install. Data was then stored in csv files which can be read into Weka.

This method aims to classify which player days are a churn day. Each active player day is labelled as churn, if the number of subsequent days left to play is 0. Otherwise, the player day is labelled as no churn. The model is then trained to identify instances which are churn days.

Each of the 6 models are trained using 60% of their relevant dataset. The model is tested using the remaining 40% of the data which is unseen up to this point. The models were restricted to a minimum number of objects per branch of 100 instances to restrict the complexity of the model. Performance and accuracy of the models is evaluated using the accuracy measurement methods outlined in Section 2.2.4.

5.2.2 Huber Loss Regression

Churn modelling is typically carried out using binary classification techniques. Rather than attempting to classify a player day as a churn or no churn day, his method attempts to predict the number of days left to play then assigns a class based on the result of the predicted value.

Two possible measurements for the time left to play were considered: the number of calendar days left to play and the number of active days left. Whilst the number of calendar days left to play might have more relevance for player intervention, it is harder to predict since a low activity player might only place once more many days in the future. Therefore, the measurement used to evaluate time left to play is the number of active days left to play.

A Huber loss regression model, as described in Section 2.2.2, is used to predict the number of subsequent days left to play for each player day in the dataset. If the predicted number of subsequent days is less than a cut-off value, that instance is classified as churn. Otherwise, if the predicted number of subsequent days is greater than the cut-off value the instance is labelled no churn. The process which is followed to label a churn day based on the predicted number of days left to play is outlined in Figure 12.

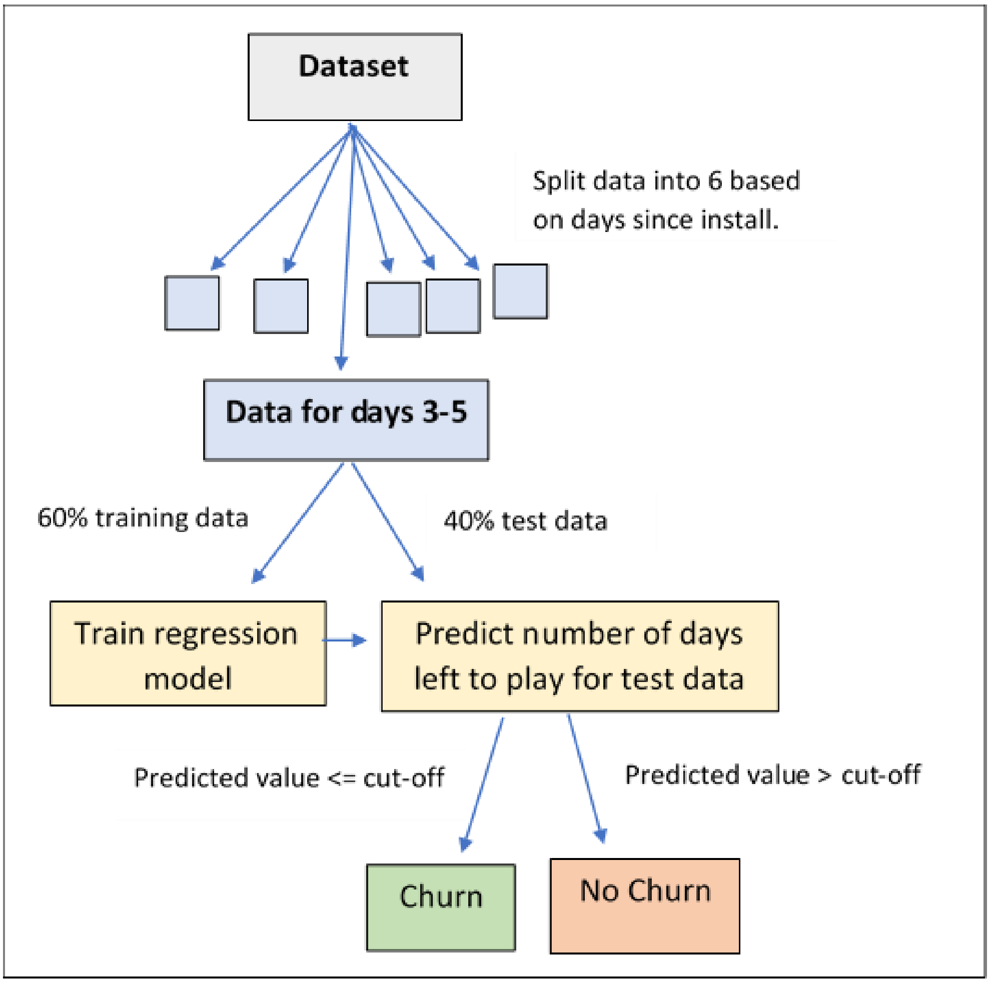

Figure 12: Regression approach to identifying instances of churn.

Since player days are given a classification value, the model performance measurements outlined in Section 2.2.4 can be used to evaluate the performance and accuracy of the model. The models were trained and tested using the Scikit-learn library available in Python.

5.3 Features

For this model, the aim is to select features which measure player behaviour that can be captured in all games stored on the deltaDNA database. There are many approaches to measuring player behaviour, for example; social activity, spending activity and progression throughout a game. The features chosen for this model fall into four categories: frequency, length of play, variability, and intensity of play. These categories capture a variety of information about player behaviour, however, they measure values which can be captured in any games.

The study carried out in [28] produced a model which consisted of game agnostic features, however, many of those features were calculated at session level. Measuring features at session level gives a more granular insight into player behaviour, however, some of the games which send data on the deltaDNA database do not register gameplay in sessions. Therefore, the number of features which measure at session level have been limited. For the few features which do measures at session level, each day is treated as one session for games which do not register sessions. All remaining features are aggregated at day level.

Features of the game agnostic model can be split into four categories: frequency, length, variability, and intensity of play.

To measure the frequency of play the chosen features are activity, average number of sessions per day and the number of events. Activity measures the number of active days to date out of all possible calendar days to date. The number of events measures the number of events which have occurred to date. Lastly, the average number of sessions per day measures the number of sessions to date divided by the number of active days to date.

Length of play measures the time that a player has spent playing, hence, the average daily playtime. This is calculated by the total time played to date divided by the number of active days to date.

Variability measures how many different types of events have occurred while playing. The proportion of event types is calculated by the number of distinct events which have occurred for that player to date out of the total distinct events to date across all players.

Intensity of gameplay is measured by the frequency that events occur. This is measured as the average number of second between events per session per day. Hence, the average time between events is calculated for each session then averaged across the sessions in that day.

This study considers behaviour across several games; therefore, the measurements of the features must be normalised by the values of all players of that game. Similarly, many of the features measure values aggregated to date, e.g. the number of events to date. These values are normalised by all players on the same day since install. Therefore, an example measurement for a player of Game X on Day 5 would be normalised by all players of Game X on 5 days since install. The method of normalisation used is:

$$Normalised\ X = \frac{X - MEAN(X_i)}{STDDEV(X_i)}, \tag{14}$$

which ensures that values are distributed with a mean of 0 and standard deviation of 1. Many of the features are logarithmically scaled to reduce the impact of extreme values on the models.

Finally, the number of days since install and the number of subsequent days are measured. The number of days since install is used to split data into the 5 datasets required for each model. For the prediction model, the model attempts to predict the exponential logarithm of this value to reduce the effect of instances where the number of subsequent days is very large.

5.4 Data

Most instances of player churn occur in the first few days following install. For this reason, player behaviour is observed for 30 days following install. Data was extracted from the deltaDNA database for players who installed between the two-week period between 26/6/17 to 10/7/17 and observed player activity for 30 days after install.

Many of the features are measured at day level, however, these involve processing event level data. For processing purposes, it is not possible to use large games such as those considered in Section 4.1. Instead, games with a smaller player base are required. Therefore, 10 games were selected which have at least 100 daily installs and 1000 daily active users, hence, these games are approximately a tenth of the size of player base as the games studied in Section 4.1. The names of these games will not be provided to ensure anonymity.

The data was cleaned to remove any days which gave no insight into player behaviour or incorrect values. When an application is opened this sends events to the database, therefore, if an application is open then closed immediately this registers a player day. For this reason, any day which has less than 1 second of playtime was removed from the dataset. Similarly, events which occurred on dates outside the sample period were removed. For some applications, play time is measured as the amount of time an application is open. Therefore, daily play time measurements which were greater than 4 hours were altered to 4 hours to remove any extreme values caused by an application running in the background of a phone.

5.5 Results

5.5.1 Decision Tree

A decision tree technique was used to classify which players days are churn days. Table 3: Decision tree results for a game agnostic model. displays the results of the performance measures for each of the 6 models tested using their respective test datasets. The measure of accuracy is high for all models; however, it is important to consider the values for precision and recall to obtain a better understanding of the results.

The precision value for all models is greater than 50%, therefore, the number of false positives is always less than the number of true positives. The measure of recall is much more varied across the models as recall varies between 14% and 99% across the models. When the recall is low, the model fails to identify all churn days. The models which perform best are based on install day and 16-20 days after install.

Table 3: Decision tree results for a game agnostic model.

Model	*Day 0*	*Day 1-2*	*Day 3-5*	*Day 6-10*	*Day 11-15*	*Day 16-20*
Accuracy	0.681	0.742	0.845	0.877	0.906	0.919
Precision	0.672	0.516	0.603	0.894	0.585	0.926
Recall	0.798	0.282	0.203	0.568	0.142	0.992

5.5.2 Huber Loss Regression

The method of predicting the number of days left to play and assigning a class based on whether the number of days is less or greater than a cut-off value enables the flexibility to alter this value.

Several cut-off values were evaluated to determine which has the value has greatest accuracy in identifying player churn days. Table 4: Results of the accuracy of the Huber Loss regression method in labelling churn days for a game agnostic model based on various cut-off values. displays the results of the accuracy measures across all models for different cut-off values. As the cut-off value is increased, more instances are classified as churn, hence, the value of recall increases.

Table 4: Results of the accuracy of the Huber Loss regression method in labelling churn days for a game agnostic model based on various cut-off values.

Cut-off value of predicted subsequent days	*1.5*	*1.75*	*2*
Accuracy	0.778	0.768	0.759
Precision	0.551	0.532	0.518
Recall	0.697	0.725	0.745

The results in Table 5 display the values of the performance measures for the 6 models tested using their respective test datasets. Several cut-off points were considered; however, the selected cut-off value of 1.75 predicted subsequent days had the best performance across all models.

The results show that precision remains steadily around 50% in all models. However, recall declines as the models are trained using increasing number of days since install. Hence, other than the model based on install day data, the models fail to identify all churn days.

Table 5: Results of accuracy measures for Huber Loss regression method of identifying churn days based on a cut-off value of 1.75.

Model	*Day 0*	*Day 1-2*	*Day 3-5*	*Day 6-10*	*Day 11-15*	*Day 16-20*
Accuracy	0.551	0.742	0.831	0.874	0.898	0.918
Precision	0.542	0.486	0.477	0.526	0.491	0.615
Recall	0.994	0.467	0.324	0.383	0.239	0.06

One of the most important aspects of this method is the ability to alter the cut-off value. A cut-off value of 1.75 predicted subsequent days fails to capture all churn days as the number of days since install increases. Hence, the flexibility of the cut-off value means that the value can be varied for different models. In doing so, the aim is to classify a larger number of days as churn days which will ensure greater completeness in identifying true churn days. Recall declines as the number of days since install which are modelled increases, therefore, the cut-off value can be increased gradually to increase the recall value.

Table 6 displays the performance measures for each of the 6 models using different cut-off values for the predicted number of subsequent days left. As expected, increasing the cut-off value results in higher recall values in most models. In this case, recall of approximately 0.5 is achieved in 5 out of 6 models. This means that around 50% of true churn days are identified correctly. Almost all models tested with a higher cut-off value experience a decrease in precision which means there are an increased number of falsely identified churn days.

Table 6:Accuracy measures for Huber Loss regression method of churn modelling using varied cut-off values.

Model	*Day 0*	*Day 1-2*	*Day 3-5*	*Day 6-10*	*Day 11-15*	*Day 16-20*
Cut-off Value	1	2	2.5	3	4	5
Accuracy	0.666	0.727	0.815	0.838	0.859	0.821

Precision	0.638	0.502	0.441	0.426	0.339	0.234
Recall	0.882	0.488	0.488	0.568	0.563	0.534

5.6 Analysis

5.6.1 Feature Importance

One of the most important results which can be gathered by modelling churn based on the number of days since install is which features have greatest importance to each model. Therefore, measures which drive churn can be identified and strategies can be put in place to encourage players to play for longer.

The decision tree model selects nodes based on which has the greatest information gain, therefore, nodes at the top of the tree are of the greatest importance in determining the class of an instance. For each of the 6 models split by days since install, the up to the three top nodes are outlined in Table 7.

Table 7: Decision tree to classify churn days - top 3 nodes.

Model	Top Three Nodes
Day 0	Playtime
Day 1- 2	Average number sessions, Playtime, Proportion of event types
Day 3-5	Number of events, Activity, Time between events
Day 6-10	Activity, Number of events, Playtime
Day 11-15	Number of events, Activity
Day 16-20	Number of events, Average number sessions, Proportion of event types

The results suggest that on install day the most definitive feature to determine player churn is the amount of time spent playing. This can be linked to the results from Section 4.3 which determined that the amount of play time on install day had a significant impact on player retention. The significance of this metric suggests that exploration of a game is more important than actual play on install day. Game developers could use this to motivate players to fully explore the game using signposting and rewards for completing walk throughs of different tasks.

Across the remaining models, frequency of play is regularly an important feature in determining player churn. Therefore, encouraging players to play regularly is likely to extend their player lifetime. Player management strategies such as implementing daily missions which offer increased rewards will encourage players to develop long term behaviour of playing regularly and discourage churn.

The proportion of event types which occurred is an important feature in the models based on days 1 to 2 and days 16 to 20 since install. This suggests that players who explore the different elements of the game are less likely to churn. To encourage this behaviour, players could be guided through the various aspects of a game in the first few days following install. Similarly, players active on day 16 to 20 after install could be prompted to explore any elements of the game they have not discovered.

5.6.2 Comparison of Models

The results of both the decision tree method and the Huber Loss regression method to identify churn days determine that both models consistently achieve high accuracy in the number of correctly classified instances. However, the values of precision and recall identify areas where the models do not perform well.

Player lifetimes are short for free-to-play games as many players churn shortly after install. Therefore, the datasets used to train models on install day and days 1 to 2 are likely to have more instances of player churn than models later in the player lifetime. This may explain the reason for good performance of models based on install day data.

The decision tree classifier approach, resulted in high precision rates and varied recall values. This means that the models had low number of falsely labelled churn days and classified a high number of churn cases as no churn.

The trade-off between the number of false churn cases and false no churn cases depends on the individual game. For free-to-play games, the risk of sending a prompt or an offer to a player who is not about to churn is less important than identifying those players who are about to churn. Therefore, high recall is favourable above high precision.

One of the main benefits of the Huber Loss regression method is the flexibility to control the cut-off value of the predicted number of days left to play. Controlling the cut-off value affects the number of players classified as churn. Therefore, a company which is concerned with reducing the number of player falsely identified as churn could set the cut-off value to a low value. This is would reduce the number of true churner identified, however, the number of false positives could be lowered. However, for a company which is more concerned with identifying players who are about to churn than falsely identifying players as about to churn could implement a higher cut-off point.

The results of the Huber Loss regression method enable the cut-off value to be different for each model. The results in Section 5.5.2 determine higher recall values can be achieved by increasing the cut-off value as the days since install being modelled is increased. Again, the ability to alter the cut-off value offers flexibility to control the performance of a model.

Implementing a churn model using a decision tree offers simple visualisation of the rules of the model. This enables understanding of which features have greatest importance to each model. To compute feature importance for the Huber Loss regression method, the absolute normalised coefficients can be calculated for each model to ensure all features have equal scales.

A Huber Loss regression method to modelling player churn is more complex than a standard decision tree and would require decisions to be made about the best cut-off value. However, the flexibility introduced by using a cut-off value enables control over the precision and recall of a model.

6 A Game Specific Approach to Churn Modelling

A client of deltaDNA requested a churn model to model to identify which of their players are about to churn. This offered an opportunity to create a game specific churn model. The results of the game specific model can be compared against the game agnostic model outlined in Chapter 5 for this game.

Comparing a game specific model against the game agnostic model offers insight into how contextual information about a game affects the accuracy of prediction. These results can be used to evaluate the whether a game specific model is necessary to identify those players about to churn.

6.1 Details of the Game

For anonymity, the name of the client and game will not be revealed. Instead, the game will be referred to as Game A.

Game A is a free-to-play mobile game. It is a crossword game which displays an empty crossword that players fill by finding words in a set of letters. Game A features many levels which increase in difficulty. Players are offered two help functions: shuffles and hints. A shuffle changes the order in which the set of letters are displayed to the user. Hints can be used to reveal one letter in an uncompleted word in the crossword.

The in-game currency used in Game A is coins. Coins are awarded when the user completes a level, however, they can also be purchased using real currency. A user can also choose to watch a video advertisement to earn coins at an increased rate. Coins can be used to purchase the previously mentioned hints.

6.2 Method

Following the discussion in Section 5.6, the selected technique to build a churn model for the client was the Huber Loss Regression method. This offers the client flexibility to decide the optimal cut-off value for their model. The regression model produces a set of model coefficients which can be easily implemented in a system to identify players about to churn, therefore, this model offers a simple implementation. Further, the model is easy to retrain with a new dataset if there are significant changes to the game.

Like the game agnostic model, churn days are identified as those when a player has zero subsequent days of play.

The objective of this project is to measure how contextual information affects the accuracy of prediction. Therefore, the performance of the game agnostic model can be compared against the game specific model to measure how contextual information about a game affects performance of the models. For Game A, a game specific model will be built using features which are specific to the game. This will be compared against the game agnostic approach as outlined in Section 5.3.

Analysis of the sample data collected for Game A determined that over 80% of players churned within 10 days from install. Therefore, the period from install to day 10 has the greatest number of observations of player churn. For this reason, the data is split into 6 datasets: each real day 0 to 4 since install and days 5 to 10 since install. Models are trained using 60% of these datasets then tested using the remaining 40%.

6.3 Features

Four categories were identified to measure player behaviour within Game A: activity, competency, spending and progression.

Activity measures how often the player plays the game. The frequency in the number of sessions played and the length of sessions are a good measure of player activity. Average daily number of sessions tracks the frequency of plays whilst average daily play time tracks the length of play.

Competency measures how good the player is at the game. Player competency is measured by the number of shuffles used, the number of coins awarded and the number of invalid words entered.

Spending measures aim to capture how aware players are of in-game currency and the impact this has on their gameplay. Hints are purchased using the in-game currency of coins, therefore, measuring the number of hints used is a simple way to measure awareness of the items available for purchase. Similarly, as watching video advertisements enable users to earn coins at an increased rate, measuring the number of videos watched gives insight into how the desire to purchase hints impacts gameplay.

Lastly, progression measures how quickly the player progresses through the game. The main path of progression in Game A is completion of levels (often referred to as missions).

Like the game agnostic model, many of the features are measured as cumulative values (e.g. the total number of hints to date). For this reason, measures which are aggregated to date are normalised by all players on the same day since install. All remaining measures are normalised by all players. Features are normalised using the method in Equation 14 to ensure that the distribution of each metric is normal, that is they have mean of 0 and standard deviation of 1. In doing so, all features of the model have an equal scale. Therefore, the magnitude of the coefficient of each feature gives insight into that features importance in determining the number of subsequent days left to play.

6.4 Data

Data was extracted from the deltaDNA database for players who installed between the period 14/6/17 to 21/6/17 and observed player activity for 28 days after install. The dataset was limited to all users who installed and remained on the most recent update of the game. This removes any influence of the build of the game on the results. Lastly, as noted in Section 5.4, measurements of play time can be influenced by an application remaining open while not being played. For this reason, total play time was limited to a maximum of 4 hours per day.

6.5 Results

This section displays the results of the Huber Loss regression approach to modelling player churn for Game A. The results for the game specific model and the game agnostic model will be discussed.

6.5.1 Game Specific Model

Many cut-off values were tested, however, the best results across all models were achieved when a cut-off of 1.75 was used. That is, any player predicted to have less than 1.75 subsequent active days left of play was classified as churn. The results of the performance measures for the game specific model using a cut-off of 1.75 are displayed in Table 8.

Model accuracy varies between 56% and 70% across the models. The values of precision are always less than 0.5. This means that the number of falsely identified churn days is always greater than the number of correctly identified churn days. However, the values for recall are much greater with values between 0.5 and 0.7. Therefore, the models may falsely identify many churners but they also manage to identify more than half of those who will churn.

Table 8: Performance measure results for the game specific churn models using Game A data.

Model	*Day 0*	*Day 1*	*Day 2*	*Day 3*	*Day 4*	*Day 5-10*
Accuracy	0.7023	0.6672	0.6092	0.5758	0.5558	0.6438
Precision	0.4659	0.3551	0.3479	0.3581	0.3605	0.4089
Recall	0.6905	0.5970	0.6168	0.6594	0.7034	0.4983

6.5.2 Game Agnostic Model

Since the aim is compare the performance of the game agnostic model and the game specific model, the cut-off value is kept at 1.75 predict days left to play. This means the results of both models can be compared. The results of the performance measures for the game agnostic model using a cut-off of 1.75 are displayed in Table 9Table 8.

Model accuracy varies between 59% and 68% across the models. Comparison of the results against the game specific model shows that the accuracy is higher using the game agnostic approach for all models except on day 0.

Like the game specific approach, the values of precision are always lower than 0.5. Again, the precision values for the game agnostic models are slightly better than the game specific models except on day 0. The game agnostic model produces similar recall values to the game specific model. The day 0 game agnostic model produces the highest recall value of 84.96%. However, as the number of days since install increases the number churners the models fail to identify increases.

Table 9: Performance measure results for the game agnostic churn models using Game A data.

Model	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5-10
Accuracy	0.6022	0.6847	0.6387	0.5947	0.5941	0.6528
Precision	0.4445	0.3604	0.363	0.37	0.3848	0.4373
Recall	0.8496	0.6479	0.6336	0.6544	0.6739	0.4982

6.6 Analysis

Comparison of the results of the game agnostic model against the game specific model for Game A produce some interesting results. The performance of both models is very similar. In both cases, the models have low precision rates and high recall using a 1.75 cut-off value.

The performance measurements show that the game agnostic approach produces better results across the models on days 1 to 10 since install for accuracy, precision, and recall. However, the game specific approach produces the better model on day 0. In both cases, the performance of the models is underwhelming. However, insight into which features are most important on each day can be gathered from the regression model coefficients. The client can then use these results to implement player management strategies to encourage users to play for longer.

Comparing the results of the models suggests that the game agnostic approach to churn modelling can produce equal, if not better, results than a game specific model. To confirm these results, further testing would be required to compare game specific models for a variety of games against the game agnostic approach.

7 Conclusion

The final chapter of this study will discuss how the objectives of this project were approached and how this could have been carried out more effectively given the knowledge gained throughout this study. This section will act as an evaluation of the project overall with some direction as to the how this study could be developed further.

7.1 Summary

This project considers behavioural predictive modelling from two perspectives: player retention and player churn. Plotting the curves of actual player retention offered insight into which player metrics are the most influential in determining player retention. Furthermore, using a truncated period of data to estimate player retention confirmed that the distribution of player retention can be modelled by a Stretched Exponential function. The comparison of a single model against the average of segmented player models determined that contextual information about a player had little impact on prediction of long term player retention.

The approach to game agnostic churn modelling outlined player behaviours which can be measured across any free-to-play game. Secondly, two approaches to churn modelling are considered and compared. The results of performance measures outlined that there is flexibility to improve the accuracy of a Huber Loss regression approach to churn modelling. Lastly, producing game specific churn model for a client provided a case study to evaluate the performance of the game agnostic churn model. Results show that both models have comparable performance which suggests that contextual information about a game does not necessarily produce better performance in churn modelling.

7.2 Evaluation

The project was successful in achieving the objectives. At the beginning of the project, a significant amount of research was required to understand the current approaches to predictive behavioural modelling and the areas of research which are beneficial to deltaDNA. Similarly, a considerable amount of time was spent learning new technologies such as Python's Pandas and Scikit-learn and deltaDNA database structure.

There were two main aims to this research. Firstly, to understand how contextual information about a player and their behaviour influences the accuracy of estimation of player retention. Secondly, to measure how contextual information about a game impacts the performance of predictive churn modelling. This study involved several objectives such as identifying patterns between player metrics and retention values and feature creation for game agnostic and game specific churn models. Similarly, the project proposed challenges in determining efficient methods of accessing, handling, and analysing extremely large datasets of event level game data.

The project met the outlined objectives by carrying out research into the previous studies of predictive behaviour modelling in the games industry, approaching behavioural modelling for both player retention and churn, and combining classification and regression approaches to build a churn model which could classify churners in any free-to-play game.

The approach to estimating retention for cohorts of players based on their player metrics was selected at the beginning of the project. The limitation of this approach was that including more detailed information and other metrics could create a very large number of player segments, hence, it was difficult to extend the research in this area. On reflection, a different approach such as survival analysis would have been considered.

The method of nonlinear least squares chosen to estimate player retention was simple to implement and produced reasonable results. Due to the time constraints of the project, further methods of estimation of the parameters were not considered. However, given more time, a method such as maximum likelihood estimation could be considered.

As discussed throughout this project, research in the game industry typically considers the data of a single game. Hence, this study adds to the small number of papers which model player behaviour across many games. This paper is one of few which approach modelling player churn in a game agnostic manner. Lastly, to the best of our knowledge, this is the first piece of research which compares the performance of a game agnostic churn model against a game specific churn model.

7.3 Future Work

Although the research was successful in meeting the objectives, there are many areas in which further research and development would be necessary. The following paragraphs outline areas in which future research could be carried out.

Future research might consider using survival analysis to predict the parameters of the retention curve. Survival analysis involves censoring data much like assigning a class and estimating the optimal parameters of the function using maximum likelihood estimators. Approaching retention prediction in this way would enable inclusion of more player metrics and granular detail of the metrics.

The game agnostic features of the churn model were selected based on those used in other literature and the available data in the deltaDNA database. Ideally, many features would be created then feature selection could be used to identify those which provide are of greatest information. Hence, future work in this area might consider feature creation on a wider scale.

Churn modelling enables the identification of which features are of most importance in driving churn. Player campaign strategies can be developed to encourage extension of player lifetime. To test the whether these player campaign strategies are successful, A/B testing can be used. In games, A/B testing involves splitting a cohort of players into two, one split of players will be shown a campaign and the other is not. This method can be used to evaluate the performance of a campaign in extending player lifetime. Whilst this is outside the scope of this study, it would provide insight into how the results of predictive churn modelling can be implemented.

Finally, as the number of mobile game players increases the availability to understand player behaviour becomes increasingly important. The amount of behavioural player data is fast expanding, however, the lack of data sharing in the games industry means research is restricted to modelling single games. In the future, I hope that data sharing becomes increasingly popular in the games industry in doing a greater understanding of player behaviour is reached.

References

[1] E. McDonald, "The Global Games Market 2017 | Per Region & Segment | Newzoo", Newzoo, 2017. [Online]. Available: https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/. [Accessed: 22- Sep- 2017].

[2]]"The Communications Market Report 2016", *Ofcom*, 2017. [Online]. Available: https://www.ofcom.org.uk/research-and-data/multi-sector-research/cmr/cmr16. [Accessed: 22- Sep- 2017].

[3] "Deloitte Mobile Consumer Survey 2017: The UK Cut", *Deloitte Mobile Consumer Survey 2017*, 2017. [Online]. Available: https://www.deloitte.co.uk/mobileuk/. [Accessed: 22- Sep- 2017].

[4] "Form F-1 Registration Statement", *Sec.gov*, 2017. [Online]. Available: https://www.sec.gov/Archives/edgar/data/1580732/000119312514056089/d564433df1.htm. [Accessed: 22- Sep- 2017].

[5] [1]"Nick Parker on Twitter", *Twitter*, 2017. [Online]. Available: https://twitter.com/NickParker4U/status/512192889091534848. [Accessed: 22- Sep- 2017].

[6] Y. Nizan and Y. Nizan, "33 Mobile Game Benchmarks and Rules of Thumb", *SOOMLA*, 2017. [Online]. Available: http://blog.soom.la/2016/08/33-benchmarks-in-mobile-games-business.html. [Accessed: 22- Sep- 2017].

[7] "What Analysing 400+ Games Has Taught Us - GameAnalytics", *GameAnalytics*, 2017. [Online]. Available: http://blog.gameanalytics.com/blog/what-analysing-400-games-taught-us.html. [Accessed: 22- Sep- 2017].

[8] "Port", *Netlib.org*, 2017. [Online]. Available: http://www.netlib.org/port/. [Accessed: 22- Sep- 2017].

[9] Generalized Linear Models", Scikit-*learn.org*, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/linear_model.html#huber-regression. [Accessed: 22- Sep- 2017].

[10] "Retention Prediction - deltadna documentation", *deltadna documentation*, 2017. [Online]. Available: http://docs.deltadna.com/reference/analyze/retention-prediction/. [Accessed: 22- Sep- 2017].

[11] I. Roseboom, "Gender split in F2P games: Who's playing what - deltadna.com", *deltadna.com*, 2017. [Online]. Available: https://deltadna.com/blog/gender-split-in-f2p-games/. [Accessed: 22- Sep- 2017].

[12] "1.3.6.6.8. Weibull Distribution", *Itl.nist.gov*, 2017. [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda3668.htm. [Accessed: 22- Sep- 2017].

[13] M. Mozer, R. Wolniewicz, D. Grimes, E. Johnson and H. Kaushansky, "Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry", *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 690-696, 2000.

[14] H. Hwang, T. Jung and E. Suh, "An LTV model and customer segmentation based on customer value: a case study on the wireless telecommunication industry", *Expert Systems with Applications*, vol. 26, no. 2, pp. 181-188, 2004.

[15] V. Umayaparvathi and K. Iyakutti, "Applications of Data Mining Techniques in Telecom Churn Prediction", *International Journal of Computer Applications*, vol. 42, no. 20, pp. 5-9, 2012.

[16] Y. Zhao, B. Li, X. Li, W. Liu and S. Ren, "Customer Churn Prediction Using Improved One-Class Support Vector Machine", *Advanced Data Mining and Applications*, pp. 300-306, 2005.

[17] Y. Richter, E. Yom-Tov and N. Slonim, "Predicting customer churn in mobile networks through analysis of social groups", in *Proceedings of the 2010 SIAM international conference on data mining.*, 2010, pp. 732-741.

[18] B. Huang, M. Kechadi and B. Buckley, "Customer Churn Prediction for Broadband Internet Services", *Data Warehousing and Knowledge Discovery*, pp. 229-243, 2009.

[19] A. Sharma and P. Kumar Panigrahi, "A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services", *International Journal of Computer Applications*, vol. 27, no. 11, pp. 26-31, 2011.

[20] G. Nie, G. Wang, P. Zhang, Y. Tian and Y. Shi, "Finding the Hidden Pattern of Credit Card Holder's Churn: A Case of China", *Lecture Notes in Computer Science*, pp. 561-569, 2009.

[21] T. Mutanen, S. Nousiainen and J. Ahola, "Customer churn prediction - a case study in retail banking", in *Proceedings of the 2010 Conference on Data Mining for Business Applications.*, 2010, pp. 77-83.

[22] Y. Xie, X. Li, E. Ngai and W. Ying, "Customer churn prediction using improved balanced random forests", *Expert Systems with Applications*, vol. 36, no. 3, pp. 5445-5449, 2009.

[23] K. Morik and H. Köpcke, "Analysing Customer Churn in Insurance Data – A Case Study", *Lecture Notes in Computer Science*, pp. 325-336, 2004.

[24] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques", *Expert Systems with Applications*, vol. 34, no. 1, pp. 313-327, 2008.

[25] S. Yoon, J. Koehler and A. Ghobarah, "Prediction of Advertiser Churn for Google AdWords", in *JSM Proceedings*, 2010.

[26] G. Dror, D. Pelleg, O. Rokhlenko and I. Szpektor, "Churn prediction in new users of Yahoo! answers", *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*, 2012.

[27] J. Hadden, A. Tiwari, R. Roy and D. Ruta, "Churn prediction using complaints data", in *Proceedings of world academy of science, engineering and technology*, 2006.

[28] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting and C. Bauckhage, "Predicting player churn in the wild", *2014 IEEE Conference on Computational Intelligence and Games*, 2014.

[29] A. Drachen, E. Lundquist, Y. Kung, P. Rao, R. Sifa, J. Runge and D. Klabjan, "Rapid Prediction of Player Retention in Free-to-Play Mobile Games", 2016.

[30] J. Runge, P. Gao, F. Garcin and B. Faltings, "Churn prediction for high-value players in casual social games", *2014 IEEE Conference on Computational Intelligence and Games*, 2014.

[31] A. Drachan, R. Sifa, F. Hadiji and J. Runge, "Predicting Purchase Decisions in Mobile Free-To-Play Games", 2015.

[32] R. Sifa, C. Bauckhage and A. Drachen, "The Playtime Principle: Large-scale cross-games interest modeling", *2014 IEEE Conference on Computational Intelligence and Games*, 2014.

[33] M. Viljanen, A. Airola, T. Pahikkala and J. Heikkonen, "Modelling user retention in mobile games", *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 2016.

[34] J. Kawale, A. Pal and J. Srivastava, "Churn Prediction in MMORPGs: A Social Influence Based Approach", *2009 International Conference on Computational Science and Engineering*, 2009.

[35] B. Weber, M. John, M. Mateas and A. Jhala, "Modeling Player Retention in Madden NFL 11", in *Innovative Applications of Artificial Intelligence*, 2011.

[36] Z. Borbora, J. Srivastava, K. Hsu and D. Williams, "Churn Prediction in MMORPGs Using Player Motivation Theories and an Ensemble Approach", *2011 IEEE Third Int'l Conference on Privacy, Security, Risk* and *Trust and 2011 IEEE Third Int'l Conference on Social Computing*, 2011.

[37] 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", *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 2016.

[38] Z. Burns, I. Rosebooom and N. Ross, "The Sensitivity of Retention to In-Game Advertisements: An Exploratory Analysis", in *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[39] 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", *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 2012.

[40] C. Chambers, W. 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*, 2005.

[41] D. Pittman and C. GauthierDickey, "Characterizing Virtual Populations in Massively Multiplayer Online Role-Playing Games", *Lecture Notes in Computer Science*, pp. 87-97, 2010.

[42] Z. Borbora and J. Srivastava, "User Behavior Modelling Approach for Churn Prediction in Online Games", 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, 2012.