

Predicting Player Disengagement in Online Games

Hanting Xie^{*}, Daniel Kudenko[†], Sam Devlin[‡], and Peter Cowling[§]

Department of Computer Science
University of York
YO10 5GH, York, UK

Abstract. Game engagement, as one of the most fundamental objectives for game designers to achieve, has become an attractive industrial and academic topic. An important direction in this area is to construct a model to predict how long a player could be engaged with a game. This paper introduces a pure data driven method to foresee whether a player will quit the game given their previous activity within the game, by constructing decision trees from historical gameplay data of previous players. The method will be assessed on two popular commercial online games: *I Am Playr* and *Lyroke*. The former is a football game while the latter is a music game. The results indicate that the decision tree built by our method is valuable to predict the players' disengagement and that its human-readable form allow us to search out further reasons about what in-game events made them quit.

Keywords: Game Data Mining, Player Modelling, Decision Trees

1 Introduction

In the global game industry, over 1500 commercial games are published annually [8]. Nevertheless, only a few of them gain popularity and become memorised by the history of games. For the continual growth of the games industry, more companies could succeed if they could fully understand their players. Therefore, how to shape an in-depth understanding of players has become a big issue in the area of games. Recently, game data mining has become popular in this area to help developers to understand more about their customers. Some research focuses on how players behave in the game [14] especially whether they play in a legal way [6]. Others concentrate on classifying players by their play styles [10], which would be advantageous for the developer to better satisfy their customers.

For many players, except for external factors, a predominant reason which may lead to their disengagement is design flaws in the game. Therefore, it would

^{*}✉:hx597@york.ac.uk

[†]✉:daniel.kudenko@york.ac.uk

[‡]✉:sam.devlin@york.ac.uk

[§]✉:peter.cowling@york.ac.uk

be helpful if we could discover some rule-sets based on the game that could predict whether a player will quit (or disengage). To be more precise, if one's actions match all the rules in a set, he/she would be highly likely to leave the game without hesitation.

In this work, we promote a pure data driven method to investigate the causes of disengagement in games. The data are collected from two commercial online games in different genres; 'I Am Playr' [1], a first-person football game, and 'Lyroke' [2], a music game, both developed by 'We R Interactive' [3].

2 Background

In this section we will cover the relevant background material and terms necessary for the comprehension of this work.

2.1 Game Telemetry

To understand players, collecting data from users is always the first step. A widely utilized technology in the game industry is called *game telemetry*. Telemetry refers to obtaining data through remote access, in the sense of games, that is to transmit data collected from a game sever to a collection server and formatted there to support further analysis [11]. The data collected usually includes different types which are related to the genre of the game. For example, in sport games, valuable game metrics could be *match type*, *team selection* and *country chosen* etc. In the game industry, game telemetry data could be used both externally and internally. As an example of external use, the *World of WarCraft Armory* [5] provides its players with statistics about its characters and guilds. As an internal tool, developers could take advantage of game telemetry data to detect bugs and adjust the settings of the game.[15]

2.2 Game Data Mining and Player Modelling

Game data mining or game analytics is the application of data mining in the area of games [11]. Its intention is to train models from game telemetry data using machine learning algorithms. Player Modelling is one example of this where the models trained are intended to represent a behaviour of the player. For example, how they play or experience the game.

The method introduced in this paper is based on *supervised learning*. The problem of supervised learning can be considered as classification(or regression). In this case, a set of attributes or *features* to be analysed is selected. A single instance in a dataset refers to a specific vector of corresponding values of those selected features. Each instance of the dataset used to train the model is given a *label* corresponding to its classification or the group it belongs to. The model generated by supervised learning intends to uncover the correlation between a group of selected features and the labels. So that after training, the model should be able to assign labels to new incoming instances automatically.

2.3 Decision Tree

The model learnt in this paper is a *decision tree*; a tree data structure which is generated by a divide-and-conquer strategy [7]. Decision trees are one of the most easily interpretable data structures in data mining and have been chosen, therefore, as this will be helpful to demystify what events are likely to result in players' disengage. Given the human readable output of this model the resultant decision tree can be used by game designers to inform future development of the game without the designer requiring indepth knowledge of data mining.

A decision tree includes a root with several nodes connected by paths. The root and each non-terminal (or leaf) node are features of the dataset whilst the leaf nodes are the labels that would be assigned to a new data instance provided they had all the features given by the nodes from the root to that leaf. An example of decision tree is shown later in Figure 1. While building, every node could also be taken as root to be linked to more nodes until it has reached a leaf node. The decision tree is in fact another form of rule sets. Because the process of heading to leaves is just the same with matching rules.

There are various algorithms for building up decision trees, the specific algorithm used in this paper is called *C4.5* [9] but any decision tree learner could be used with our methodology. This paper will follow the original algorithm without any specific modification. Our contribution is the methodology of applying decision trees to predict player disengagement not a refinement of the learning algorithm. We have chosen C4.5 because the algorithm is widely used in data mining and so that the method we introduce in this paper can be easily implemented by interested game developers.

2.4 Feature Selection

In data mining, features are those factors related to outcomes. In terms of decision trees, they are the things that utilised to branch the tree. However, having too many redundant features can affect both the accuracy of the model built, the time to train the model and the memory needed. It is not rare that there are thousands of possible features to be analysed in a game. For example, one of the games to be analysed in this paper, I Am Playr, has 6408 key assorted events that could be useful features. Considering this, it is important to apply feature selection to prune those irrelevant ones.

There are many algorithms to perform feature selection. In this paper, we conduct feature selection (and model learning) using of WEKA [4]. The algorithm used by WEKA is called Correlation based Feature Selection (CFS). It is an algorithm based on an evaluation formula including both correlation measure and heuristic search strategy [12]. As with our motivation to use decision tree, the reason to apply CFS is that it is widely used, well developed and would be convenient for further investigations.

3 Games

To test our method, we applied it to two existing commercial games of distinct genres, both developed by *We R Interactive*. An introduction to both is included in this section and is intended to emphasise the significant differences in the games.

3.1 I Am Playr

I Am Playr is a *free to play*, first person, football game on the social network *Facebook*. Like other free to play games, I Am Playr offers different items such as boots, cars and other luxury items that players can purchase in the game.

Our method focussed on data regarding the occurrence of events in the game. In I Am Playr there are currently a total of 6048 events possibly experienced by players. Worldwide, most of those events happened nearly hundreds of times in only one minute. Those events in game originate from gameplay events and system events. The former is about the actions performed by players, for instance, ‘Wining Matches’, ‘Playing Videos’ and ‘Training’. Whilst the latter, system events, the game decides when the player will experiences, for example promotions such as a ‘Daily Bonus’ or ‘Free Coins’.

3.2 Lyroke

Lyroke is a commercial music game available on multiple platforms including Facebook, iOS and Android. The main mechanism of this game is that a song is played with most of the lyrics displayed, the players need to respond with the missing lyrics before the singer reaches them. The most common purchases in game are to unlock new songs.

Similar to I Am Playr, the events are from gameplay and system. However, the event types are very different due to the different genres or the games. In Lyroke, gameplay events include ‘Using Power Up Items’ and ‘Answering Lyric’ whereas system events include ‘System Gifts’ and ‘Achievements’.

4 Methodology

The objective of our method is trying to predict the disengagement of users as a result of the events they experienced. Specifically, we predict whether there will be a decrease in a players’ activity from one month to the next on the basis of their behaviour in the first month. In this section, we will show how our method is assessed in both games.

4.1 Data Collection

The data from these two games is all gathered by game telemetry and stored in a web data server. All the raw data we used for analysis are downloaded from

this sever. As mentioned before, the event data will be the only dataset covered in this paper, Table 1 shows a general description of its compiled format on the server.

As can be seen in Table 1, a complete event type includes 'st1', 'st2', 'st3' and the corresponding action of the event type, attribute 'n'. So, the format of events we use as features for training is 'st1-st2-st3-n'. Thus, as an example, the event (feature) appeared in Table 1 would be 'Item-Equip-Boots-IAMHelios'.

Another attribute should be noticed is the 'game week' ('i' in Table 1) which represents the in game week that the player has currently reached. Game week (or any similar measure of a player's progression through the game) could also be a relevant feature because the same event in game could generate contrasting meanings in different game weeks. Since it is not an event, and is partially game specific, we built models both with and without this attribute with the latter being for the purpose of generalisation. However, most games will have a similar metric that could be used in place of game week with similar results expected to the models generated using this feature.

With permission and help from We R Interactive, we are able to access up-to-date data. For I Am Playr, we analysed datasets from January to March 2014. Whereas in Lyroke, we had datasets from March to April 2014.

Attribute Name	Descriptions	Examples Instance
Timestamp	The Unix Time	'1388534450669'
Type	The type of this dataset	'event'
s	Anonymous User ID	'00008'
n	The specific action related to event	'IAMHelios'
v	Values related to action	'1'
i	The week in game that player is currently in	'3'
st1	Level 1 description of event	'Item'
st2	Level 2 description of event	'Equip'
st3	Level 3 description of event	'Boots'

Table 1: Event data formate with Example Instance from I Am Playr

4.2 Data Labelling

Before training, we must label players' change in engagement between months. This is necessary for training and evaluating the decision tree only. Afterwards, when deploying the model, players can be classified as one of the following types simply by observing the events they experienced in the previous month. For the purpose of future research on players' full retention trends, we decide to not only assign *Decreasing* labels but also consider labels of *Increasing* and *Stable*. Those three labels are distributed by the following steps described in Algorithm 1.

Firstly, in step 1, we take only users that exist both in January and February into consideration. Next, in steps 2 to 6, each user's (usr) event counts in both January ($EventC_{Jan}(usr)$) and February ($EventC_{Feb}(usr)$) will be calculated and recorded in the event count lists ($EventCList_{Jan}$ and $EventCList_{Feb}$). And

then, in step 7 and 8, the two lists are sorted. After that, in step 9 to 12, for each user, if his/her event counts belongs to the first quarter of sorted January event count list, then we say the rank of this user is 1 in January. Likewise, his/her rank would be 2, 3 or 4 if they were in second, third or fourth quarter in January. Repeating the same thing for February. Finally, for each user, if his/her rank in January ($C_{Jan}(usr)$) minus his/her rank in February ($C_{Feb}(usr)$) surpasses 2, then he/she would be allocated to the Decreasing Group. Similarly, if a user's rank in February minus his/her rank in January is bigger than 2, it means he belongs to the Increasing Group. In comparison, players with ranks lower than 2 will be categorised into the Stable Group.

Algorithm 1 Label Distribution Algorithm

```

1:  $U \leftarrow U_{Jan} \cap U_{Feb}$ 
2: for user in  $U$  do
3:    $EventC_{Jan}(usr), EventC_{Feb}(usr)$ 
4:    $EventCList_{Jan} \leftarrow EventCList_{Jan} \cup EventC_{Jan}(usr)$ 
5:    $EventCList_{Feb} \leftarrow EventCList_{Feb} \cup EventC_{Feb}(usr)$ 
6: end for
7: Sort  $EventCList_{Jan}$ 
8: Sort  $EventCList_{Feb}$ 
9: for user in  $U$  do
10:   $C_{Jan}(usr) \leftarrow 1, 2, 3$  or 4 if user in 1st, 2nd, 3rd or 4th quarter of  $EventCList_{Jan}$ 
11:   $C_{Feb}(usr) \leftarrow 1, 2, 3$  or 4 if user in 1st, 2nd, 3rd or 4th quarter of  $EventCList_{Feb}$ 
12: end for
13: for user in  $U$  do
14:  if  $C_{Jan}(usr) - C_{Feb}(usr) \geq 2$  then
15:     $Label(usr) \leftarrow$  Decreasing
16:  else if  $|C_{Jan}(usr) - C_{Feb}(usr)| < 2$  then
17:     $Label(usr) \leftarrow$  Stable
18:  else if  $C_{Jan}(usr) - C_{Feb}(usr) \leq -2$  then
19:     $Label(usr) \leftarrow$  Increasing
20:  end if
21: end for

```

4.3 Model

As discussed in the preceding section, to train a supervised model, we need a set of instances with labels. In our case, an instance is a complete event history of a player with his/her label; decreasing, increasing or stable.

I Am Playr In I Am Playr, since we use players' actions in January to predict the trend of their retention between January and February, the feature set contains all events the player experienced in January. Subsequently, for each instance, we use the number of occurrences of an event (feature) as the value of this feature. At the same time, the labels are decided by use of the Algorithm 1.

After the data is prepared, we make use of the C4.5 (called J48 in WEKA) decision tree learning algorithm in WEKA to build the models. Modelling with and

without feature selection are both tried and their performances are compared. At the same time, as discussed in Section 4.1, we will also consider models both with and without the attribute ‘game week’. Finally, for completeness, we will consider models with both feature selection and ‘game week’.

In this paper, we conduct two experiments to assess the performance of our method. In the first we split players from the January-February dataset into 10 training and testing sets, and then perform 10-fold cross validations on them to observe the average performance. In this experiment, we randomly pick 3000 instances from each group (decreasing, increasing and stable), thus there are 9000 instances in total in this test. Those 9000 instances would be separated into training set and testing set automatically. The reason to use 3000 instances is that the smallest group merely contain around 3000 instances, so the use of this number to limit instances from other groups could ensure that the training data has a same size for each group.

The other one is to validate the decision tree trained by the trend between January and February to predict the trend between February and March. In this experiment, we use exactly the same model trained from the preceding experiment to test on 3000 (1000 instances for each group) randomly picked instances from February-March data.

Lyroke The method of training decision tree of Lyroke remains the same with I Am Playr. However, the dissimilarity is that since we only hold the data about March and April, only the 10 fold cross validation test could be examined. In this experiment, we pick 900 instances randomly from each group(decreasing, increasing and stable), thus there are totally 2700 instances in this test. In 10-fold cross, those 2700 instance would be divided into training set and testing set automatically. We choose 900 instances since this is the size of our smallest set.

4.4 Evaluation Metrics

In order to evaluate our method, ‘Recall’, ‘Precision’, ‘F-Measure’ and ‘Accuracy’ will be used as metrics of performance. Among them, ‘Recall’ refers to the true positive rate or sensitivity in the context of classification while ‘Precision’ represents the positive predictive value. ‘F-Measure’ is a harmonic mean of those two. As a commonly used metric, ‘Accuracy’ cares both true positives and true negatives. In formulas below, ‘true positives’, ‘true negatives’, ‘false positive’ and ‘false negative’ are represented by ‘tp’, ‘tn’, ‘fp’ and ‘fn’ respectively.

$$Recall = \frac{tp}{tp + fn} \quad (1)$$

$$Precision = \frac{tp}{tp + fp} \quad (2)$$

$$FMeasure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3)$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (4)$$

5 Result and Discussion

In this section we will discuss the relative performance of the models in predicting player engagement in both games. For comparison, we compare our method to a naive method of randomly guessing player engagement based on the known distribution of example instances.

5.1 I Am Playr

There are in total four different models constructed for I Am Playr. The size of the original model is 1705 nodes with 853 leaves and was learnt from 6408 features. After applying feature selection, the number of features is reduced to 79, which results in 1451 nodes with 725 leaves in the tree. Another tree built with the ‘game week’ feature contains 117 nodes with 59 leaves and was learnt from 6409 feature. Compared with the original model, it is clear including the ‘game week’ prunes the tree. Finally, the last model with both feature selection and the ‘game week’ feature included is 69 nodes 35 leaves and was learnt from only 35 features. The first four layers of this model are illustrated in Figure 1.

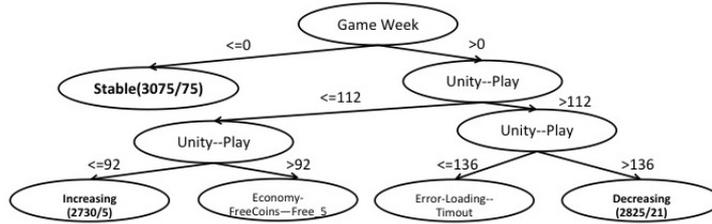


Fig. 1: Top of Decision Tree Learnt for I Am Playr with Feature Selection and the ‘Game Week’ Feature Included

As demonstrated by Figure 1, the decision tree is clear and easy to be interpreted. To classify the expected engagement of a new player, start from the root feature, ‘game week’, if its value is 0, the instance will be labelled as stable. Otherwise, we then consider the feature ‘Unity-Play’ (right child), this process continues until one of the 3 labels (decreasing, increasing or stable) is reached. One thing should be noticed is that many instances are classified as stable by only considering ‘game week’. This could be a sign of overfitting and may mislead the prediction, perhaps explaining the performance in Figure 7.

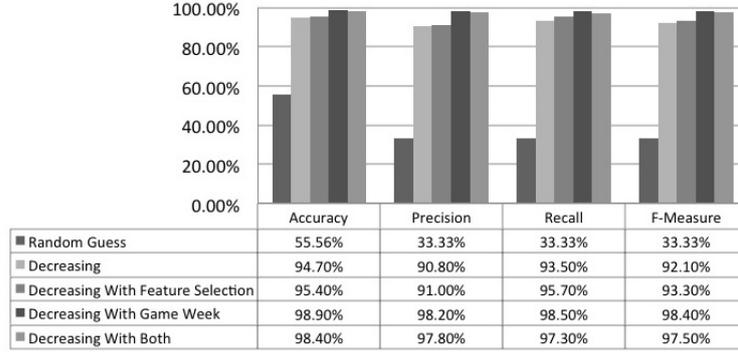


Fig. 2: 10 Fold CV Performance on Decreasing label in I Am Playr

10 fold cross validation Figure 2 indicates the performance of predicting decreasing engagement. As shown in the chart, the accuracy, precision, recall and F-Measure are high for all models. This shows our method can reliably predict player disengagement. At the same time, we could see that the feature selection process and ‘game week’ attribute further improve the accuracy. Nevertheless, if we use both of them, the accuracy is not as high as when we use ‘game week’ without feature selection. This might be because when we use ‘game week’, some other important features are filtered by the feature selection.

Since the four methods (except for random guess) are showing close performance, it is needed to check the RMSE (Root mean squared error) on them to identify which one is the best. The accuracies with RMSE are $94.70\% \pm 0.4362$, $95.40\% \pm 0.4197$, $98.90\% \pm 0.1063$ and $98.40\% \pm 0.1207$ respectively. This further supports that ‘Decreasing With Game Week’ is the best model, but it also shows that all models perform statistically significantly better than random guessing.

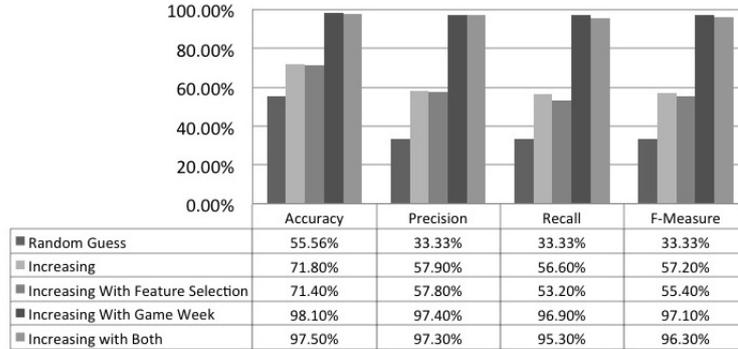


Fig. 3: 10 Fold CV Performance on Increasing label in I Am Playr

Figure 3 illustrates the 10 fold cross validation performance of predicting the increasing labelled group. Apart from predicting disengagement, we also try to find out whether we could predict which players would get more addicted to this game. The graph here shows that we achieve a significantly higher accuracy than random guess.

In this example, feature selection does not improve these metrics, but it does speed up the process of training. Adding feature selection reduces the time cost from 539.9s to 3.3s which is because it shrinks the number of features from 6048 to only 79. Therefore, it also helps to save memory for storage. An important thing to be noticed is that when we apply ‘game week’ attribute on this, the performance is improved dramatically, as ‘game week’ is an important feature to show whether a player is experienced or not. Thus it helped to improve the accuracy of model by dividing and conquering. However, since the two methods with ‘game week’ are showing close accuracies, an RMSE check of them is necessary. The accuracies with RMSE are $98.10\% \pm 0.1063$ and $97.50\% \pm 0.1207$ respectively. Therefore, the method with only ‘game week’ is the best again, but the significant reduction in training time that occurs when adding feature selection may be favoured over the small improvement in these metrics.

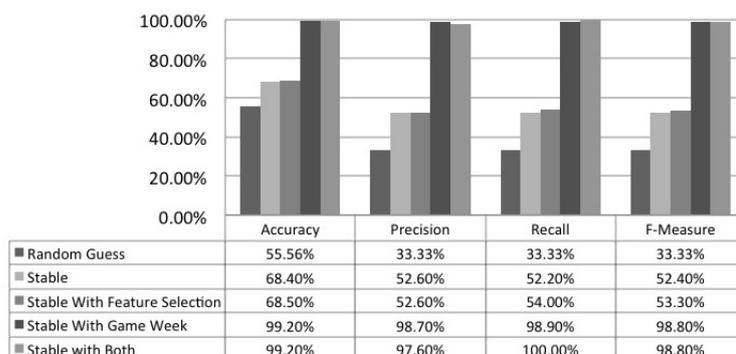


Fig. 4: 10 Fold CV Performance on Stable label in I Am Player

Figure 4 shows the 10 fold cross validation performance of predicting the stable labelled group. To predict the stable group is the most demanding task in the research as players could keep stable in entirely different situations. For example, those players who only play once a week belong to stable group whereas some frequent players are also in the Stable Group as a result of certain play patterns shaped by their timetable. This result shows that the original model performs significantly better on all metrics than random guessing. The application of feature selection is shown to be of no substantial use once again on performance, however as discussed above, feature selection is essential to reduce the time cost. Similar to what happened to increasing group, including the ‘game week’ feature significantly improves all metrics. Likewise, the RMSE check is needed to distinguish the two accuracy bars related to ‘game week’. The accuracies with RMSE are $99.20\% \pm 0.1063$ and $99.20\% \pm 0.1207$. So there is no longer a significant difference between using ‘game week’ with and without feature selection and, therefore, if predicting stable players was a priority we would recommend using feature selection for it’s benefits in reducing the memory and time requirements when training the model.

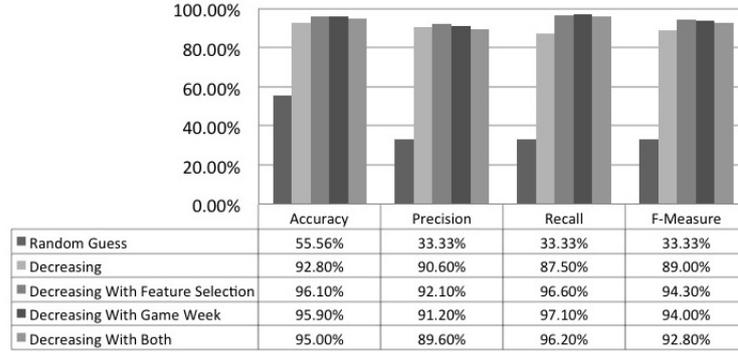


Fig. 5: Verified on February - March Performance on Decreasing label in I Am Playr

Verified on February - March Data Figure 5 reveals the result of testing on the February-March data. It shows that the performance on the decreasing group (disengagement) still remains great. So it proves that our model has a desirable function to predict disengagement in the case of generalisation. In addition, the performance could be further improved by applying feature selection or ‘game week’ attribute. Similar to what we did in 10 fold cross validation, RMSE would be performed for all accuracy bars (except random guess) to figure out which method is the best. The accuracies with RMSE are $92.80\% \pm 0.4535$, $96.10\% \pm 0.4175$, $95.90\% \pm 0.4827$ and $95.00\% \pm 0.4855$ respectively, and it indicates that the model with only feature selection performs the best.

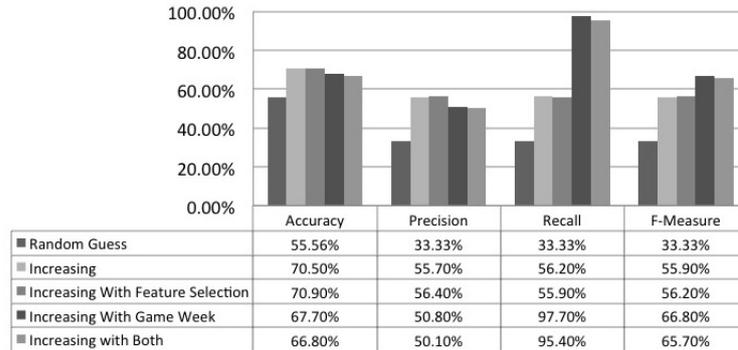


Fig. 6: Verified on February - March Performance on Increasing label in I Am Playr

Figure 6 shows less improvement in predictions using our model. However, the performance of all decision trees is still higher quality than that of random guess. So it means the patterns of increasing playtime players tend to change considerably during different months. However, one thing should be noticed is that with the help of ‘game week’ attribute, the recall of the increasing group could be boosted noticeably. According to the formula of recall, it is to say that the number of true positives has been increased. As the accuracies of original

model and the model optimised by feature selection performs similar with each other, RMSE should be applied to show the details. The accuracies with RMSE for those two methods are $70.50\% \pm 0.4535$ and $70.90\% \pm 0.4175$. According to this, the best remains the model with just feature selection applied.

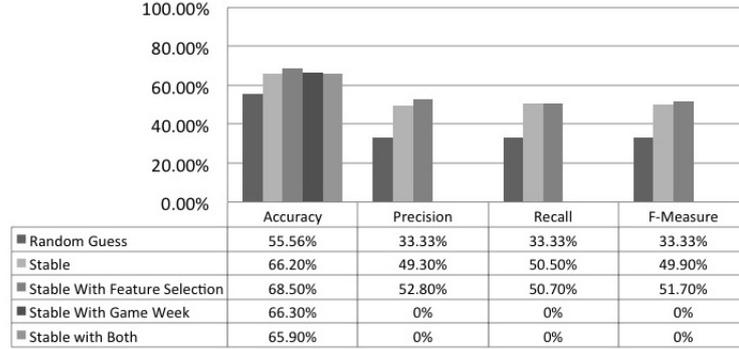


Fig. 7: Verified on February - March Performance on Stable label in I Am Playr

Figure 7 displays a similar result as the increasing group. Furthermore, for this group, if the ‘game week’ is included, the precision, recall, and F-Measure drop to 0. According to the formulas, this occurs because these models give no true positives. Which is also to say that no instances are correctly assigned to the stable group. This is possible as in the original model of January-February, the ‘game week’ attribute is trained to be the only important feature to predict whether a player is in the stable group or not. This fact could be clearly seen in Figure 1. However according to the February-March data, the patterns of stable users generate another type of tendency. Since the accuracies of all methods are close to each other, the RMSE is used. The accuracies with RMSE for all methods are $66.20\% \pm 0.4535$, $68.50\% \pm 0.4175$, $66.30\% \pm 0.4827$ and $65.90\% \pm 0.4855$ individually. So as can be seen, the conclusion remains again that the model with only feature selection performs the best.

5.2 Lyroke

As with I Am Playr, there are also four models for Lyroke. The size of the original model is 589 nodes with 295 leaves and was learnt from 7382 features. After applying feature selection, only 33 feature are left. This resulted in a tree with 413 nodes and 207 leaves. The tree with the ‘game week’ feature included contains 149 nodes with 75 leaves learnt from 7383 features. Including the ‘game week’ has again reduced the tree size. Finally, the last model with both feature selection and the ‘game week’ feature is 39 nodes with 20 leaves and was learnt from only 23 features.

10 fold cross validation Figure 8 suggests that we attain a great performance in the decreasing group once more, which could be further enhanced by feature

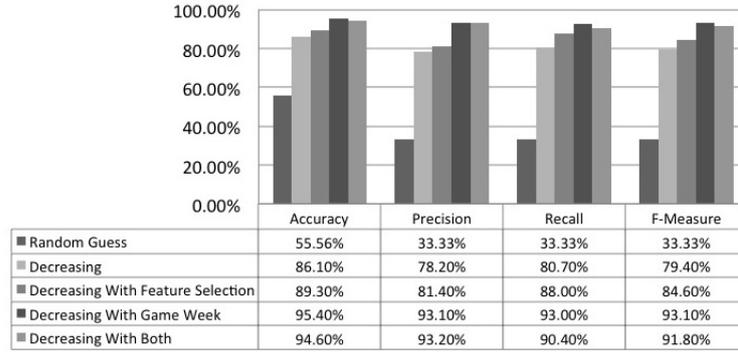


Fig. 8: 10 Fold CV Performance on Decreasing label in Lyroke

selection and the ‘game week’ attribute. The accuracies with RMSE are $95.40\% \pm 0.2120$ and $94.60\% \pm 0.2166$ for the last two bars with ‘game week’ respectively. Therefore, the model with only ‘game week’ performs best.

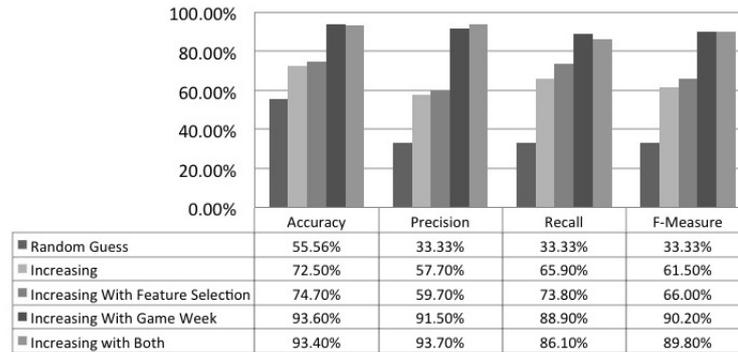


Fig. 9: 10 Fold CV Performance on Increasing label in Lyroke

Figure 9 shows again that all our models acquire a significantly higher accuracy than random guess. The accuracy without the ‘game week’ feature exceeds 70% and could be used for problems at a reasonable error standard. At the same time, feature selection also boosted both the performance and speed. Especially the speed, as adding feature selection reduced the time cost of training the model from 61.88s to 0.28s. Also the number of features to be analysed decrease from 7382 to only 23. Like I Am Playr, the accuracy could be improved to higher than 90% by applying ‘game week’ attribute too. Similar to decreasing group, the accuracy bars related to ‘game week’ should be checked with RMSE. The results are $93.60\% \pm 0.2120$ and $93.40\% \pm 0.2166$ correspondingly. In this case, there is no significant difference in performance. Therefore, the speed and memory advantages of using feature selection may be favourable if predicting this class of players is a priority.

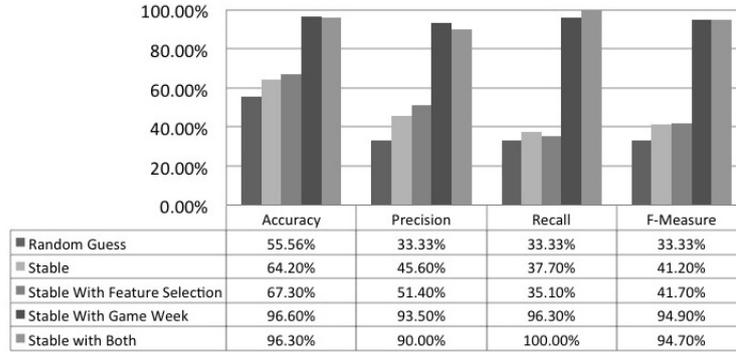


Fig. 10: 10 Fold CV Performance on Stable label in Lyroke

Figure 10 shows a similar result as the increasing group. As discussed above, the task of distinguishing stable players is a much more complicated process. Despite this, we still observe a significant improvement compared with random guess. With 'feature selection' applied, the performance could be improved once again. Also, as we discussed before, the time cost of it has been reduced a lot. Similar to increasing group, the performance could be improved to higher than 90% by including the 'game week' attribute. Likewise, the accuracies of those bars related to 'game week' should be investigated by RMSE. The corresponding accuracies with RMSE are $96.60\% \pm 0.2120$ and $96.30\% \pm 0.2166$. This indicates again that there is no significant difference and, therefore, the model with both feature selection and the 'game week' feature may be favourable.

5.3 Summary

From the results above, we conclude our method is accurate and stable to predict disengagement cases in varying conditions. The findings can be applied to other genres of games, which means it is possible to track players' certain behavioural patterns for detecting their disengagements.

For the increasing and stable groups, the original model could offer an accuracy around 60%-70% which is significantly higher than that of random guess. Furthermore, including the 'game week' feature could significantly improve the performance. Although it is not a general attribute in all games, most should have similar attributes used to record the progress of players. However, when the model is used to perform classifications in a later month slot (February-March), the performance turned out to be of inferior quality. This means that the action patterns of players who belong to increasing/stable group are changeable between month slots.

Finally, using feature selection typically does not have a significant effect on the accuracy of the model but does significantly reduce the time taken and memory required when training the model.

6 Related Work

Relevant existing research in this domain tend to be based on two concepts: *player modelling* and *game design by data mining*. The main directions of player modelling are: *player analysis*, *behaviour detection* and *character simulation*

In the direction of *player analysis*, the purpose of work by Mahlmann et al. [13] resembles that of this paper. In that paper, the authors introduced how to utilize the system records (e.g. Playing time, Total number of deaths, Causes of deaths etc.) of players to predict when players are going to leave the game *Tomb Raider: Underworld*. This work differs from ours because we focus on the events experienced by players within a period of time in different game genres. By focusing on events the resultant model includes only features developers can manipulate the occurrence of directly.

Another paper focusing on players' interests also shows similar intention. Except instead of investigating when players will leave, Bauckhage et al. [8] focus was more about the abstract interest of player. The authors built a model which could smoothly match the change of interests and predict the likelihood of players' exit with distinct modelling methods. They focused more on fitting existed random process models while we try to build new models based on actual activities from players.

In terms of the utilization of players' behaviour, another related paper is written by Ahmad et al. [6] aiming at detecting gold-farmer player in a massively-multiplayer online role-playing game (*MMORPG*) game called *EverQuest II*. In this paper, the authors used the pre-processed behavioural data of players to perform a binary classification on whether players are gold farmers (who acquire in-game currency and sell for real-world money) or not. So its purpose is dissimilar with ours but the methods could be used interchangeably, with decision trees on events used to predict gold farmers. The advantage of an interpretable model could also be useful for understanding critical behaviours of gold farmers.

For tuning design, the work by Weber et al. [15] is an example. In this paper, the authors put forward a method to build up a model consists of game features and player's retention so that the most valuable game features could be found according to the sorted features list. The purpose of this paper is also different as it used games features rather than events to predict, which means it also covered factors such as control settings. Weber gave priority to adjusting game features (game settings) rather than the disengagement of players.

7 Conclusion

We introduced a method which could be used to predict the disengagement of players as a result of their recent activities. To validate our method, we tested it on two online games of different genres and achieved desirable outcomes.

Moreover, we endeavoured to use the same method in testing participants in the increasing and stable groups. The performance was better than that of random guess and could possibly be used under the condition of an accepted error standard.

The method has potential practical significance for two reasons. First, data analysts could better understand players and indicate what type of designs are not welcomed or accepted by users. Second, it would potentially become a very high-level tool for developers to adjust their design in games. The developers, however, do not have to understand the complicated mechanism and relationships in their event space to find out what will lead customers to quit as the resultant model is human readable and easily interpreted. One thing to be mentioned is that, our method assumes that events in game are frequently generated which could reflect the activities of players. Nevertheless, games which contain only sparse events might still be analysed as login information of players is frequent events that happened in every online game.

Future work will aim to improve the prediction of the increasing and stable groups. The next attempt could be using more users' information to classify players' behaviour. Also, as discussed earlier, the stable group contains too many players facing different conditions. Therefore perhaps if the players in this group could be clustered into more groups/labels, the performance of the model could possibly be improved. Finally, experimenting with targeting the players predicted to disengage with adverts and/or special offers could be useful to help maintain their engagement and prevent or delay them quitting.

References

1. I am playr. <http://www.iamplyr.com/>, accessed May 23, 2014
2. Lyroke. <https://game.lyroke.com/>, accessed May 23, 2014
3. We r interactive. <http://www.werinteractive.com/>, accessed May 23, 2014
4. Weka. <http://www.cs.waikato.ac.nz/ml/weka/>, accessed May 23, 2014
5. World of warcraft armony. <http://www.wowarmory.us/>, accessed May 23, 2014
6. Ahmad, M.A., Keegan, B., Srivastava, J., Williams, D., Contractor, N.: Mining for gold farmers: Automatic detection of deviant players in mmogs. In: Computational Science and Engineering, 2009. CSE'09. International Conference on. vol. 4, pp. 340–345. IEEE (2009)
7. Alpaydin, E.: Introduction to machine learning. Adaptive computation and machine learning, MIT Press (2010)
8. Bauckhage, C., Kersting, K., Sifa, R., Thureau, C., Drachen, A., Canossa, A.: How players lose interest in playing a game: An empirical study based on distributions of total playing times. In: Computational Intelligence and Games (CIG), 2012 IEEE Conference on. pp. 139–146. IEEE (2012)
9. Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., Scuse, D.: Weka manual for version 3-7-10 (2013)
10. Drachen, A., Sifa, R., Bauckhage, C., Thureau, C.: Guns, swords and data: Clustering of player behavior in computer games in the wild. In: Computational Intelligence and Games (CIG), 2012 IEEE Conference on. pp. 163–170. IEEE (2012)
11. El-Nasr, M., Drachen, A., Canossa, A.: Game Analytics: Maximizing the Value of Player Data. Springer (2013)
12. Hall, M.A.: Correlation-based feature selection for machine learning. Tech. rep. (1998)
13. Mahlmann, T., Drachen, A., Togelius, J., Canossa, A., Yannakakis, G.N.: Predicting player behavior in tomb raider: Underworld. In: Computational Intelligence and Games (CIG), 2010 IEEE Symposium on. pp. 178–185. IEEE (2010)
14. Thureau, C., Bauckhage, C.: Analyzing the evolution of social groups in world of warcraft®. In: Computational Intelligence and Games (CIG), 2010 IEEE Symposium on. pp. 170–177. IEEE (2010)
15. Weber, B.G., John, M., Mateas, M., Jhala, A.: Modeling player retention in madden nfl 11. In: IAAI (2011)