Predicting Daily Active Users for Match-3 Mobile Games

Neil Patrick Del Gallego
College of Computer Studies
De La Salle University - Manila
neil.dg@playlab.com

Arvin Corpuz
College of Computer Studies
De La Salle University - Manila
arvin_corpuz@dlsu.edu.ph

Oliver Bob Urag
College of Computer Studies
De La Salle University - Manila
oliver_urag@dlsu.edu.ph

ABSTRACT

We present in this paper a technique to attempt to predict the amount of daily active users for a match-3 mobile game by considering several factors that drives a game's social virality. The data was extracted from the performance of two match-3 mobile games by Playlab Inc., Jungle Cubes and Dragon Cubes. In this paper, we describe our experiments on these two games and the outcomes of the prediction model extracted.

Keywords

Data Mining, Classification, Regression, Daily Active Users, Mobile Games, Free-to-play, Retention.

1. INTRODUCTION

Free-to-play mobile games follow a common business model. It is a necessity that these games have a healthy community to drive the virality of the game. One of the key measures for determining the success of a mobile game is by using Daily Active Users (DAU) as a metric. This refers to the total amount of unique users who spent considerable time in the game given a certain date. This refers to the "stickiness" of the application. A high value for DAU indicates that there is much activity and demand for the mobile game. It is essential for game development companies that develop mobile games that follows a free-to-play model, to keep the DAU value as high as possible. This paper attempts to predict DAU value for two commercial match-3 mobile games, Jungle Cubes (JNC) and Dragon Cubes (DNC) [1], that are currently owned by Playlab Inc.

Companies that follow a free-to-play business model use various analytics tools to track user's behavior and events triggered in their applications. In this study, Playlab Inc., uses Flurry Analytics and Google Play Developer Console to collect user data. One of the attributes capable of being tracked is the DAU.

In this study, we present a hypothesis that the DAU value is driven by other attributes or events that causes a user to use the application (or in this case, play for a considerable time) extensively. Such attributes considered are discussed in the succeeding sections.

Business decisions for a mobile game are executed when there is sufficient user data collected. One of the major problems encountered by companies such as Playlab Inc., is that major releases for the game don't immediately provide significant user data. It would be appropriate to provide a forecasting technique to determine if there would be a significant user activity on succeeding days. In a practical scenario, this paper aims to answer the question; given a certain day with these collected attributes, how high will the DAU value be X days after? In our experiments, we specifically attempt to predict the Daily Active Users value at Day 7.

2. SIGNIFICANCE OF THE STUDY

Behavioral analytics has recently emerged as a practice for commercial game development. These also relates to technological advancements brought about by mobile devices that enable developers to gather more significant data through the use of commercial analytics.

There is a significant increase on the number of games that are shifting to the free-to-play (F2P) model and this study are one of the few instances wherein commercial game dataset are used for academic research. Findings will be beneficial for game companies that strictly complies with the F2P model. In specific, those who greatly rely on high user activity for their games will strongly find this paper beneficial for their business decisions.

We present a novel approach on how analysts or developers may attempt to predict the DAU value on a practical scenario. We also attempt to define features that are highly important based from our analysis of attributes on our dataset and as supported by other similar study as seen in [4].

In this manner, we present the following contributions in this paper: We formally define from related study, as well as our findings, the attributes that affects the DAU value. We attempt to create several prediction models for the DAU value using machine learning techniques and explain its sig-

[1]We shall use JNC and DNC respectively as name convention on this paper

nificance on a practical setting. We present a simple method on how one can predict the DAU value This paper are one of the few research study that uses dataset from commercial games which is otherwise, confidential and unavailable for academic research. Despite a small number of installs for both games, Jungle Cubes have very strongly correlated attributes against DAU, which make the dataset significant. We present this in detail in the succeeding sections.

We attempt to apply our best training models for both games into real world use, which is otherwise not really presented by other studies. We discuss how this is done in the methodology section of this paper.

Related work tends to use datasets from games that are commercially successful, which already provides good prediction results since patterns can already be observed. Here in our study, we consider two games that has the same genre, but one is commercially successful, and the other one is not. We therefore see noticeable differences between training models for both games. We present this comparison in the experiments section.

2.1 Related Work

Using game analytics for research purposes has recently been pursued around 2012. Analysis of user behavior in digital games has become a fundamental practice for game companies. It is also open for numerous research opportunities due to its complex nature of modelling users while also taking into consideration the elements of the game. Thus, datasets concerning player behavior can be exceptionally complex like seen in World of Warcraft, a famous MMORPG (Massively Multiplayer Online Role-Playing Game), which has lead a team of researchers attempt to cluster players based from behavioral telemetry [2]. They applied numerous unsupervised learning methods to discover clusters of "player groups" based from their playtime data and their levelling pace.

2.1.1 Determining How Players Lose Interest

A study on action and shooter games released on Playstation 3, that uses player behavior dataset, have been used to discover how players lose interest in playing a game [1]. The dataset presented in their research paper were extracted from two single-player games (Just Cause 2, Tomb Raider: Underworld) and three multi-player games (Battlefield Bad Company 2, Medal of Honor, Crysis 2). All datasets have been sampled using simple random sampling or extraction of data on a timeline where player activities are high or the game was newly released.

The interest of playing a game cannot be measured directly but can be inferred from observable data as mentioned by [1]. In reality, a player's urge to play a game is influenced by several factors that appears as unforeseeable events to the analysts involved. This ranges from variance in playing schedules, personal satisfaction, release of new game content, or new games that competes with the player's attention. Using these consideration as mentioned by [1], they modeled the player's interest in playing a game as a random process. That is, at any given time, a player's interest is a random quantity that may or may not depend on previous values and future values cannot be predicted exactly. Thus, they have restricted their mathematical models to random process models; the Gamma distribution, Weibull distribution, Inverse Gaussian distribution, and Log-normal distribution.

Of all five games analyzed, researchers deduced that the total playing times follows the Weibull distribution. Following the Weibull model, it gives a good benchmark for gaming companies to determine how a player's playtime even before the game has been released.

2.1.2 Predicting Churn Rate

There is an existing research work that is highly related to our methodology. Researchers attempted to predict the Churn Rate of commercial mobile games which also uses some attributes we use for this study [4]. In their study, they threat the churn value as a binary classification task, a player is labelled as **churned** or **returning**. Given a specified *cutoff date*, the player will be labelled as **churned** or **returning** based from two formal problems defined in their study.

They described problem **P1** to be more straightforward. Given a cutoff date, players who did not return after the cutoff date are immediately considered as churners. The green dots in 1 are considered as **churners** while the red dots are players who managed to return after the specified cutoff date. This formal definition of churn is harsh and not useful for real-world applications. Problem **P2** is more relaxed. Given a grace period after the specified cutoff date, if players return during this period, they are considered as "about to churn" wherein their engagement to the game is already low. These are the players who are likely to quit soon and knowing how much players are inside this grace period aids gaming companies on potentially rescuing these players to get back to the game. In 1, the first two green dots refer to players already churned while the third green dot inside the grace period is flagged as "about to churn." The red dot is a player who managed to return to the game after the grace period.

Figure 1 shows an illustration of formal problem **P1** and problem **P2**.

- Feature Selection
  Attributes were universal and game-content independent. The following attributes stood out based from their feature selection tests and obvious observations that affect churning behavior; **Number of Sessions**, **Number of Days**, **Current Absence Time**, **Playtime per Session**, and **Average Time Between Sessions** and **Predefined Spending Category**. Some of these attributes are observed in our study.

- Experiments and Results
  In their research work, the most accurate classifier is the decision tree, among other different classifiers; neural networks, logistic regression, and naive bayes. F1-score goes as high as 0.916 for the decision tree model. We will also be using decision trees for our prediction since it has achieved a high accuracy from this study.

3. DATASET AND INFORMATION ABOUT ATTRIBUTES

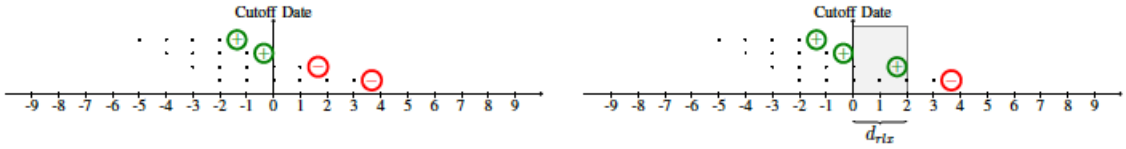


Figure 1: Formal problems defined as P1 and P2 described by [4]. P1 defines users as churners if they do not return after the specified cutoff date. In P2, players are given a grace period after the cutoff date. If players did not return after the specified grace period, they are considered as churners.

Table 1: Overview of Dataset Used

Dataset Filename	Game Title	Total Downloads	Overall Rating	Timeline of Dataset
DNC Dataset Android 0511-0911	 Dragon Cubes	50,000 - 100,000	4.2 out of 5	January 5, 2015 - September 11, 2015
JNC Dataset Android 0511-0911	 Jungle Cubes	100,000 - 500,000	4.3 out of 5	January 5, 2015 - September 11, 2015

This section contains an overview of the dataset used for this paper. The manner of extraction is also discussed in this section. Attributes are defined as well.

There are two datasets available for research and both are considered for the empirical study. The DNC Dataset and the JNC Dataset. Both datasets have been compiled and retrieved from Flurry Analytics, a commercial analytics tool used by Playlab Inc to track user's behavior on their commercial mobile games. Some attributes were retrieved from Google Play Developer Console such as user's ratings and daily crash reports. The DNC dataset refers to the Dragon Cubes game while the JNC dataset refers to the Jungle Cubes game. Both dataset are restricted to Android platforms only.

Table 1 shows the overview of the datasets used for the study.

Both datasets spans on a similar timeline, that is May 11 - September 11,2015. A span of four months have been deemed sufficient for analysis and increasing the timespan no longer yields better results.

4. ATTRIBUTE INFORMATION

This section discusses the attributes used for this study. These attributes have been gathered and selected from Flurry Analytics and Google Play Developer Console.

The initial selection of dataset has been manually performed by the researchers which they have deemed sufficient for analysis and have potential impact to the DAU value. On the succeeding section, we analyzed each variable and their correlation values to determine which variables have high relationship with the DAU value.

- Install Date - Each instance in the dataset is organized by install date. This refers to the gregorian calendar date wherein an application is installed.
- Cohort Size - Refers to the total amount of users who have installed the application on the given install date.
- Day X - This represents the retention of the application given a certain date and cohort size. Installation date becomes day 0. Retention rate is the percentage of returning users on a specified install date. For example, day 1 has 40.75% retention and 1200 cohort size. Therefore, 40.75% of users have managed to return on day 1 (489 users in cohort size)
- CrashesANRDay1 - reality, crash reports come in a day after the specified install date. For example, May 11,2015 has 3 crash reports. This means that this value was only retrieved on May 12, 2015.This counts the total number of crashes and ANRs (application not responding) reports from the application. This has a negative impact for the user experience. In reality, crash reports come in a day after the specified install date. For example, May 11,2015 has 3 crash reports. This means that this value was only retrieved on May 12, 2015.
- DailyAverageRating - This refers to the average rating by users who choose to rate the application (1 to 5, 5 being the highest) on a given date. Rating an application is not mandatory. This is a primary determination for virality. Similar to CrashesANRDay1, the tally comes in a day after the specified install date.
- LevelPlayedEvents - Refers to the accumulated event tally that is triggered when a user plays a level on the application. This is triggered upon tap of the 'Play' button. This event is reported no matter the outcome of the level being played.
- LevelSuccessEvents - Refers to the accumulated events that are triggered if a user successfully completes a level. This is triggered when the 'Win' screen is shown to the user.
- LevelFailedEvents - Refers to the accumulated events that are triggered if a user fails a level. This is triggered when the 'Lose' screen is shown to the user.

- Session - Refers to the total amount of play sessions on a given install date. A high value for session count on a given install date means that there are a lot of playthrough activity
- MKTExpenses - This is the total amount of marketing expenses, in USD, spent to advertise the game. Given an install date, the marketing expense normally determines the cohort size. A high marketing expense means more advertising channels have been used to target more potential users to install the game.
- ActiveUsers - This refers to the total amount of unique users who spent considerable time in the game given a certain date. This refers to the "stickiness" of the application.This is one of the attributes essential for determining a game's success.
- ActiveUsersDay7 - This is similar to the ActiveUsers variable but offset 7 days after the install date. This is the variable to be predicted.

In reality, given a install date, and one would like to know how many daily active users would there be 7 days after, the following variables will be used: Cohort Size, Day 1, CrashesANRDay1, DailyAverageRating, LevelPlayedEvents, LevelSuccessEvents, LevelFailedEvents, Sessions, MKTExpenses, and ActiveUsers.

Note that some variables like DailyAverageRating and CrashesANRDay1, only becomes available a day after. In a practical scenario, one could make predictions by Day 2 since it is assumed that all variables are readily available.

5. EXTRACTION OF FEATURES

This section discusses how the dataset has been gathered. Flurry Analytics is a powerful commercial tool that enables developers to analyze user's behavior in mobile applications through data observations.

Jungle Cubes and Dragon Cubes have an API installed that reports events to Flurry as long as the user is connected to the internet. Playing these games on offline mode will store such events on the local file of the device which will be sent to Flurry upon connecting to the internet.

The manner of reporting events is pretty straightforward. Figure 2 shows the overview of the process of events reporting to Flurry. When an application starts, the timestamp and location (if possible to extract) is recorded and sends the data to Flurry. This means that the attribute, *Session*, is incremented. If this is a first-time launch, then the *Install Date* is reported to Flurry[2].

5.1 Custom Events

We define custom events significant for our analysis and adjust various game features as needed. These user interactions report different events which are triggered on specific

[2]Install Date information can be found in the application settings. If for example, user installed the app on date A, and decided to play it 5 days after, the value will always be date A (not A + 5)

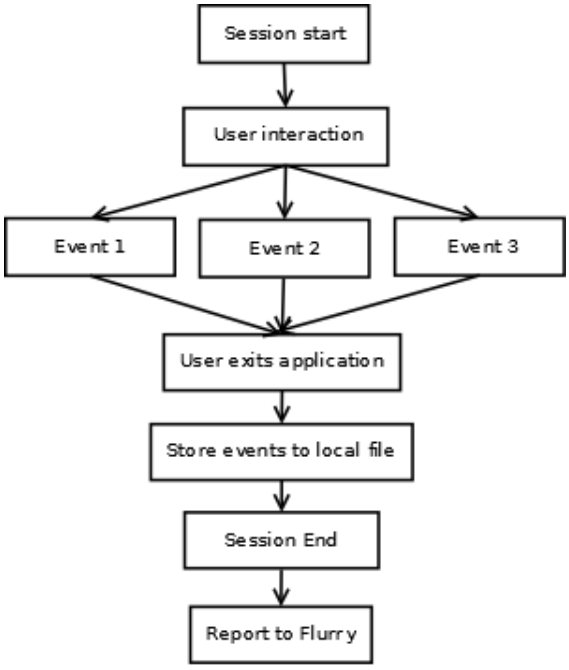


Figure 2: Overview of how events are recorded and reported to Flurry within an application's lifecycle

parts of the game. For this paper, we only considered 3 custom events, *LevelPlayedEvents, LevelSuccessEvents, LevelFailedEvents*. We will only consider these events and discuss points at which these events are triggered.

Refer to Figure 3 as reference.*LevelPlayedEvents* are triggered when the game results are shown to the user. This signifies that the game proper has concluded. In this figure, DNC and JNC will increment *LevelPlayedEvents*.

LevelSuccessEvents are triggered when the game results are shown and it has been concluded as a win result[3]. *LevelFailedEvents* are triggered when the game results are shown as a fail result to the user[4]. In Figure 3, the DNC shows a win result while JNC shows a fail result. DNC increments *LevelSuccessEvents* while JNC increments *LevelFailedEvents*.

6. METHODOLOGY AND CLASSIFICATION TECHNIQUES

This section contains the methodology and different classification techniques used to predict the value **DAU-Day7** from two datasets, JNC and DNC.

On the business point of view, Jungle Cubes is a fairly successful game released commercially by Playlab Inc. due to its constant revenue despite having a small amount of users playing the game. On the other hand, Dragon Cubes have

[3]A win for both games means that the objectives has been met by the user and can therefore proceed to the next level of the game. This is a positive event

[4]A fail for both games means that objectives were NOT met by the user. They have to repeat the level again. This is a negative event.

Figure 3: Custom events defined in this paper are triggered when these screens are shown. DNC (left) shows a win result while JNC (right) shows a lose result.

fairly considerable marketing expenses but did not reach the company's overall vision for the game. It also has seemingly random patterns as described by the marketing team. Here in this section, we attempt to uncover the plausible reason for this kind of outcome based from the dataset.

6.1 Methodology

Using the dataset we have extracted from JNC and DNC, we attempt to use this dataset as a whole and provide several models to attempt to predict **DAU-Day7**. Several machine learning techniques via WEKA were applied. We shall only discuss the outstanding set of models that perform best for our scenario.

6.1.1 Discretization of dataset for nominal use

We attempted to predict **DAU-Day7** in numeric and in nominal form. Our dataset by default uses numeric values for **DAU-Day7**. However, we wanted to observe if other classification techniques that uses nominal values for prediction may also provide significant results for prediction. Thus, we discretize our dataset to identify numeric ranges and group them into proper categories.

Therefore, we use two types of dataset for both games; the numeric type and the nominal type.

6.1.2 Test Procedure

We all use cross-validation procedure *k=10* on all tests. After finalization of the results and evaluating the best models, we attempt to use these models on a test set that is also retrieved from Flurry to resemble actual deployment and practical use.

6.1.3 Applying best models for real world use

Part of this study is to also apply the best prediction models to actual use in the industry. The analytics tool continuously tracks succeeding events from users. We identified a cutoff date for our dataset and succeeding records from both games will be used as test set[5].

6.2 Classification Techniques

Given a numeric and nominal type of datasets, we have a total of 4 datasets to test given that we have two games to consider. For numeric predictions of **DAU-Day7**, we choose to discuss the results of **M5Base, REPTree,** and **Multilayer Perceptron** due to their noticeable results. For the numeric type of datasets, we choose to use **decision-tree induction (J48)**. We performed feature selection and modified needed parameters for each training algorithm to increase the accuracy of the model. Specific details are discussed per each training technique in this paper.

7. EXPERIMENTS AND RESULTS

This section contains the experiments performed and the results of the models used. As mentioned in the previous section, we have the nominal type and the numeric type of dataset for both games. We proceed with results on using numeric values for **DAU-Day 7** followed by nominal values to represent ranges for **DAU-Day7**.

For the numeric type, we consider the following metrics

[5]Our cutoff date is based on when we have finalized the attributes for the dataset. That is on September 11, 2015. We started the experiment after this date and succeeding dates in Flurry will be used as test set, which we ended collecting data on November 9, 2015.

for evaluation; correlation coefficient, mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), and root relative squared error (RRSE). We discuss M5Base, REPTree and Multilayer Perceptron as separate subsections.

For the nominal type, we consider the kappa statistic and F-score for discussion on both games.

7.1 Correlation Analysis

We measured correlation of attributes with **DAU-Day7** to determine which attributes should we include for training different prediction models. This was our first basis for feature selection. However, we did not solely rely on correlation analysis and considered other feature selection techniques[6] to improve the accuracy of the models.

We primarily used correlation analysis to determine if there are strong relationships with **DAU-Day7** for both games. We observed that JNC has significantly strong relationships while DNC did not.

Table 2: Correlation of Attributes against DAU-Day7

	Jungle Cubes	Dragon Cubes
Cohort Size	0.6802	0.4655
Day1	-0.0054	-0.2016
Sessions	**0.8241**	0.4261
ActiveUsers	**0.9037**	0.4630
LevelPlayedEvents	0.6817	0.4226
LevelSuccessEvents	**0.7279**	0.4700
LevelFailedEvents	0.6217	0.3111
MKTExpenses	**0.7349**	0.3747
TotalPurchases	**0.8507**	0.1577
AverageRating	0.0803	0.1578
CrashesANRDay1	0.6194	0.2668
AvgSessionSeconds	-0.0595	-0.2073
MedianSessionSeconds	0.1644	-0.0938

JNC attributes yielded significantly higher positive correlation against **DAU-Day7**. There are four strong correlation values (in bold text)[7] while DNC do not have any. This indicates further proof that DNC do not yield any noticeable patterns due to its fair performance in the gaming market. JNC, on the other hand, already have a trend if we're going to use correlation as a measure.

7.2 Feature Selection

In order for our learning algorithms chosen to perform best, we filtered out the unneeded attributes primarily influenced by using established feature selection techniques. We use a wrapper scheme for feature selection for learning algorithms to perform best by removing unnecessary features. This technique is discussed in [5]. We used a best-first search method together with the selected attribute evaluator. For each learning algorithm used in our study, different set of features are selected.

[6]Various search and selection techniques are available in WEKA for feature selection.

[7]We use 0.7 as threshold for indicating strong relationship.

Basing from the initial features "proposed" to be selected by using this scheme, we manually selected, or removed some features deemed significant by this technique. Solely relying on the wrapper scheme still induced noise on the final outcome of the model. We based our manual selection method through correlation analysis and repeated observations on how it affects the overall accuracy of the model.

For the case of JNC and using decision-tree induction, we exclude this from our general feature selection scheme and used linear forward selection scheme as stated here in this paper [3]. These contributed into a slightly higher accuracy rate for JNC using decision-tree induction.

7.3 M5Base

We attempt to use M5Base provided by WEKA which is based from [6] due to its combined nature to induce decision trees and output linear models at the leaves. This suits well for our study due to the numeric form of **DAU-Day7** and the ability to output a formula which can easily be used for marketing and business decision purposes.

Table 4 shows the selected features used for creating the model using M5Base. Notice the differences of features selected for both games.

Applying the said features from Table 4, we attempt to use m5Base as predictor using the default configuration and resulted in the following measures seen in Table 3. JNC has significantly higher magnitude of error, referring to MAE and RMSE measures compared to DNC. However, the RAE and RRSE for JNC is significantly lower than DNC which means that the model of JNC seems to be more reliable than a simple predictor[8]. If we take a look into their corresponding test sets, notice the significant change of metrics for both games.

Figure 4 shows the correlation coefficent differences. Notice that JNC has high correlation. DNC has high correlation on its training set but using the test set, the correlation has dropped significantly which makes the model questionable for real world use.

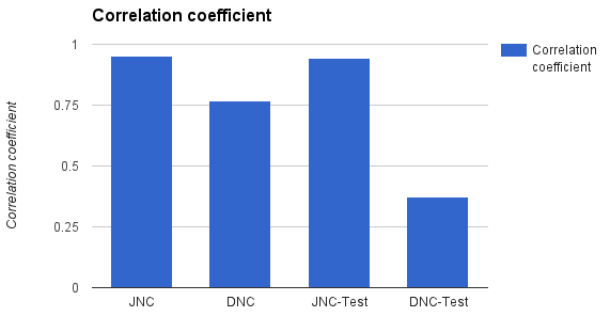


Figure 4: Correlation coefficents for m5Base

[8]Simple predictor is just using the mean as predicted value for DAU-Day7.

Table 3: Summary of results per learning algorithm

Metrics	JNC	DNC	JNC-Test	DNC-Test
Correlation coefficient	m5Base:0.9499 REPTree:0.934 MP:0.9315	m5Base:0.7673 REPTree:0.6565 MP:0.5197	m5Base:0.9442 REPTree:0.6198 MP:0.0383	m5Base:0.3731 REPTree: 0 MP:0.6587
Mean absolute error	m5Base:363.3713 REPTree:417.3018 MP:457.19	m5Base:226.1379 REPTree:262.0065 MP:277.4295	m5Base:380.6705 REPTree:943.719 MP:1530.9904	m5Base:276.2044 REPTree: 166.4048 MP:480.7897
Root mean squared error	m5Base:522.864 REPTree:602.4758 MP:608.1291	m5Base:390.7313 REPTree:456.0032 MP:516.1081	m5Base:410.3039 REPTree:1030.8327 MP:1677.0443	m5Base:285.4435 REPTree: 177.1683 MP:483.6563
Relative absolute error	m5Base:25.1493 REPTree:28.8819 MP:31.6426	m5Base:55.0067 REPTree:63.7316 MP:67.4831	m5Base:9.3048 REPTree:23.0675 MP:37.4223	m5Base:31.9779 REPTree: 19.2657 MP:55.664
Root relative squared error	m5Base:31.0991 REPTree:35.8343 MP:36.1706	m5Base:64.3702 REPTree:75.1233 MP:85.0251	m5Base:9.9679 REPTree:25.0429 MP:40.7419	m5Base:32.966 REPTree: 20.4612 MP:55.8576
Total Number of Instances	250	250	28	28

Table 4: Selected features for M5Base

	Jungle Cubes	Dragon Cubes
Features	Cohort Size Day 1 LevelFailedEvents LevelSuccessEvents Sessions MKTExpenses TotalPurchases MedianSessionSeconds	CrashesANRDay1 LevelFailedEvents LevelSuccessEvents Sessions MKTExpenses AvgSessionSeconds

Figure 5 shows the summary of error tendencies of both games using m5Base. We can see that the MAE and RMSE of JNC is somewhat higher than DNC, but looking into RAE and RRSE, we see that JNC outperforms DNC by a huge difference. The prediction model becomes more reliable than a simple predictor for JNC's case.

7.4 REPTree

We applied REPTree induction for JNC and DNC as it builds a regression tree that is also suitable for real world use. Examining the results, it did not perform on par with the outcome of M5Base. However, the outcome of this learning algorithm has comparable results than any other learning algorithms we choose. We therefore, mention the results of the model. Fewer features were deemed significant by our feature selection method in REPTree as seen in Table 5.

Table 5: Selected features for REPTree

	Jungle Cubes	Dragon Cubes
Features	Cohort Size DailyAverageRating LevelFailedEvents MKTExpenses MedianSessionSeconds MKTExpenses	CrashesANRDay1 LevelFailedEvents Sessions MKTExpenses

Table 3 shows the summary of metrics for JNC and DNC. Figure 6 and Figure 7 shows the graph of JNC and DNC's

Figure 5: Error tendencies for m5Base

correlation coefficients and error tendencies respectively. Notice that JNC and DNC's results on cross-validation procedure did not really differ from the results using m5Base. However, attempting to apply this to our provided test set, we observed that values vary greatly.

Notice that on *JNC-Test* tab, the error measures has significantly increased compared to m5Base. On the other hand, result in **DNC-Test** tab indicates that this learning algorithm faired better than m5Base when applied to

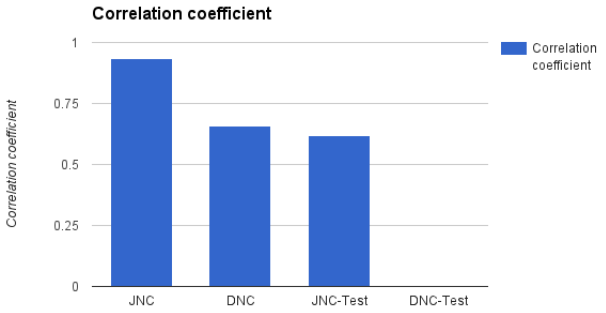


Figure 6: Correlation coefficents for REPTree

practical use. However, we cannot fully recommend using REPTree for practical use since the correlation coefficient is zero. There is no real pattern between variables that affect **DAU-Day7** and may not fully predict **DAU-Day7** on various events.

7.5 Multilayer Perceptron

We attempted to use artificial neural network using Multilayer Perceptron as learning algorithm for predicting **DAU-Day7**; on the impression that this learning algorithm is flexible and has various parameters involved that we can continuously adjust to improve the accuracy of the model. In this subsection, we discuss briefly how we adjust the settings to generate the optimal network.

Table 4 shows the selected features for multilayer perceptron using our method specified earlier in this paper.

Table 6: Selected features for Multilayer Perceptron

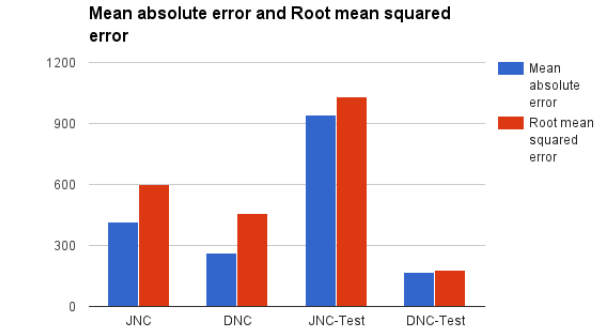
	Jungle Cubes	Dragon Cubes
Features	Cohort Size MKTExpenses TotalPurchases MedianSessionSeconds AvgSessionSeconds	Cohort Size MKTExpenses

We found that the number of hidden layers appropriate for this experiment is the total number of attributes and classes of our dataset[9]. We relied on acceptable training time to modify the learning rate and the total number of epochs. Using the default values provided did not yield acceptable results. Modifying these said parameters[10] noticeably increased the accuracy of the model that is comparable for this study.

Using cross-validation procedure, it can be observed that multilayer perceptron performed acceptably for JNC while DNC did not. The error measures for DNC for multilayer perceptron is the highest among the three numeric prediction models presented.

[9]Actual count from best results has 6 hidden layers. Tally is retrieved after selecting features

[10]Learning rate is set to 0.001, number of epochs set to 5000.



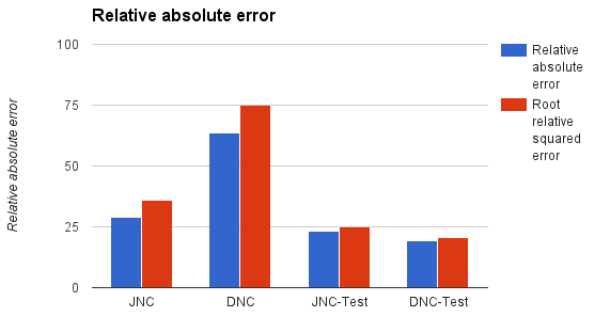


Figure 7: Error tendencies for REPTree

Applying multilayer perceptron on our test set, observe that JNC have a very low correlation coefficient this time, which is very different from the outcome of REPTree and m5Base. Notice that the MAE and RMSE have significantly increased which makes multilayer perceptron less than ideal for practical use in JNC.

Using multilayer perceptron for DNC, we can observe that the results from the test set closely resemble results from the cross-validation tests in m5Base. The RAE and RRSE is more than 50% which may indicate that using multilayer perceptron is only halfway better than a simple predictor.

Table 3 shows the summary of the metrics using multilayer perceptron. Figure 8 and Figure 9 shows the graph for the correlation coefficient and error metrics respectively.

7.6 Decision-tree induction

Using our nominal datasets for JNC and DNC, we explored how decision-tree induction will perform[11].

For the case of JNC, we concluded that 5 bins performed best across different test cases for discretization. Using 5 bins, we performed linear forward selection, which is exclusive for JNC and J48 experiment. Using this exclusive feature selection scheme, we managed to slightly have a better accuracy for this test. Our results for JNC on this type of learning algorithm indicates that it does not perform better than using learning algorithms for numeric predictions.

[11]Using J48 in WEKA

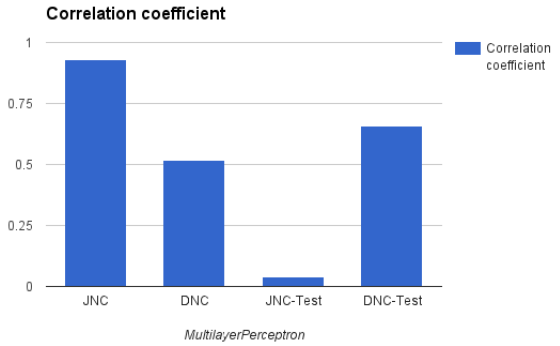


Figure 8: Correlation coefficents for Multilayer Perceptron

For the case of DNC, we notice that using this scheme has resulted into more weight being put on the 1st bin (around 92%), which made the discretized dataset severely biased. To counter this outcome, we used equal-frequency binning such that using 5 bins, the distribution of weight is even. Table 7 shows the features selected for both games using decision-tree induction.

Table 7: Selected features for J48

	Jungle Cubes	Dragon Cubes
Features	Cohort Size	Day 1
	LevelPlayedEvents	LevelFailedEvents
	Sessions	LevelSuccessEvents
	MKTExpenses	Sessions
	MedianSessionSeconds	MKTExpenses
		AvgSessionSeconds

Referring to Table 8, we can observe that for both JNC and DNC, the overall accuracy of the model does not really differ. While the discretization is different for both games, the overall accuracy are almost similar. Applying the model to our test set, around 7% of the data in JNC were misclassified while DNC's test set achieved a 100.0% correctly classification.

For the case of JNC, observing the kappa statistic is only 0.6608 using cross-validation and only 0.4717 on its test set. F-measure is slightly higher on JNC's test set. With this outcome, we deduce that our test set might not be enough to further recommend using J48 model for actual use. We noticed that our test set do not cover all ranges for **DAU-Day7**.

This goes the same for DNC but more extreme. The kappa statistic and F-measure are actually **misleading**. Since there is not much user activity and support for DNC[12], it follows that there will be a low amount of daily active users. On this given circumstance, our model predicted 100% of such instances in the test set. We can only prove that the model successfully captures this kind of scenario but not on

[12]Major development for DNC has stopped according to Playlab Inc. and no further acquisition campaigns are implemented when this study was conducted.

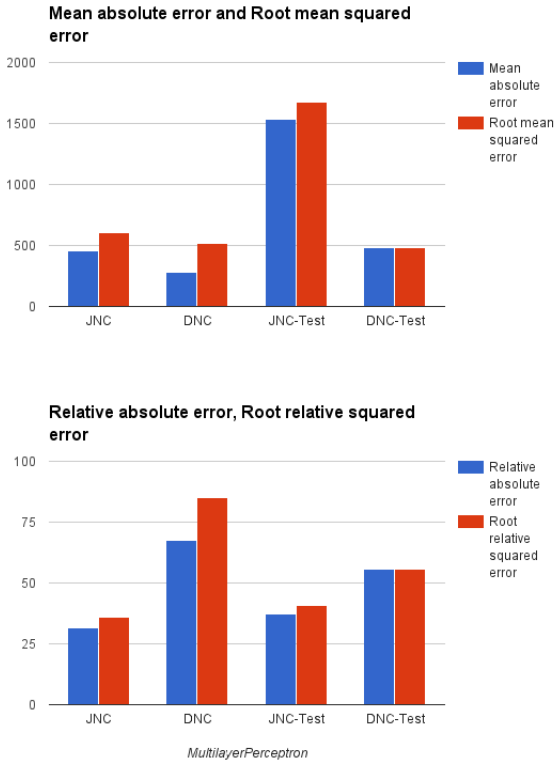


Figure 9: Error tendencies for Multilayer Perceptron

cases where the **DAU-Day7** is higher.

7.7 Observations and Recommendations

We observed that according to our results, accuracies and features selected are different from each learning algorithm. Out of the three learning algorithms proposed, it seems that using m5Base as the prediction model yields the best possible outcome for both games. It favors more for JNC as the dataset being presented has very high correlation coefficient for **DAU-Day7** unlike DNC.

While REPTree yields the lowest error rate when applied to DNC's test set, we do not recommend using REPTree for long-term use. The current situation of DNC in the gaming market has low acquisition rate, meaning that only a small amount of users get to download the game on a daily basis [13]. Furthermore, marketing expenses for DNC has stopped which may explain the low acquisition rate. This kind of scenario best fit the REPTree prediction model but somehow may not properly predict major changes in user acquisition. We therefore see m5Base as more fitting to consider the scenario where marketing expenses will increase, sudden change in session length or increase of user activity via triggering our custom events.

While multilayer perceptron appear flexible, we do not recommend using multilayer perceptron as prediction model

[13]We based this observation from our gathered test set and also observed its trend on Google Play Store.

Table 8: Summary of results using decision-tree induction

Metrics	JNC	DNC	JNC-Test	DNC-Test
Correctly Classified Instances	74.8201%	72.8%	92.8571%	100.0%
Incorrectly Classified Instances	25.1799%	27.2%	7.1429%	0.0%
Kappa statistic	0.6608	0.63	0.4717	1.000
F-measure	0.730	0.723	0.945	1.000
Total Number of Instances	250	250	28	28

for both games. First, the dataset of both games are small-scale. We therefore need more data to see any clear observations that multilayer perceptron performs acceptably. Training time of the model also consumes more time and CPU performance than the other learning algorithms we've presented, which m5Base and REPTree performed better than multilayer perceptron. Notice that multilayer perceptron performed acceptably on cross-validation method but performed worse when applied to practical use (using test set).

For the case of decision trees, we aim to provide this as a supplement alongside with the preferred prediction model. This works best with using M5Base as both of them are structured as trees. One could use M5Base to determine the regression formula and use the decision tree model to approximate **DAU-Day7** as additional clarification. Such cases wherein the prediction of M5Base agrees with the result of the decision tree model will make business decisions more reliable. However, DNC's case do not fully capture most of the cases using the decision tree model.

Out of all the learning algorithms used, we see that JNC yields acceptable results. It has low error rates and high correlation coefficient for most tests we've performed. This may justify the success of JNC in the commercial gaming market as we therefore see a trend in the data by just observing the correlation values. However, we do not justify that such trends contribute to the overall increase in daily active users. m5Base yields the highest correlation coefficient across our JNC tests and also produced low error rates. Using the model for practical use, we see that while there is little difference in MAE and RMSE, the RAE and RRSE of JNC yields only 9%. This makes it suitable for real world use.

8. CONCLUSION AND FUTURE WORK

We now attempt to verify if it is possible to predict **DAU-Day7**. Using Flurry Analytics and Google Play Developer Console to extract data and user behavior, we analyze if certain features affect **DAU-Day7** and if it is possible for such models be used to predict **DAU-Day7**.

We consider how features are frequently selected by using our proposed feature selection scheme. We see that on all our learning algorithms, those that relate to session activity (total number of sessions and session length) are selected across all our test cases[14]. This supports the findings of [4] wherein session count and length has an effect on application virality.

[14]Total of 6 test cases; 3 learning algorithms applied for JNC and DNC

8.1 Marketing expenses, session count and session length

We observed that marketing expenses has been selected on all cases. Marketing expenses for JNC has high correlation with **DAU-Day7** (0.73). While *MKTExpenses* are also being selected as a contributing feature for DNC, it yields a low correlation value when we analyzed the dataset. This supports the speculation of the marketing team that DNC does not have enough traction or "stickiness" wherein users who get to install the game leave before Day 7. Based from our study, we propose a finding that *MKTExpenses* gets high correlation with **DAU-Day7** once the game has enough content to keep users engaged. To make engagement factors high, business decisions should improve the outcome of session length and number of sessions (*Sessions* and *MedianSessionSeconds* should have strong correlation with **DAU-Day7**). Once those are set, increasing marketing expenses(to increase social reach or promote the game), will also increase the potential of gaining additional daily active users. We propose to the readers to further verify this claim as future work.

8.2 Factors in game design

We observe that relying on gathered data and game events, JNC performed better and has a better accuracy on most of the training models we proposed. Using the same features for DNC, we see that it performed poorly in general. We therefore see that the success of a game also contributes to better data and discover more patterns than if a game did not really do well in the market. We deduce that DNC has underlying problems in how the game was designed. None of the features yield high correlation value with **DAU-Day7**. Events and patterns from the data are not observable.

We propose as future work that such games like DNC should also have a concrete model to quantify the engagement factor of the game. Inferring the fun factor of a game may also be modelled by gathering user sentiment or initial feedback. Relying on events that only consider session length, triggered events and marketing expenses may not fully capture such cases like this.

8.3 Considering quality of the game

We see that we yield a somewhat high correlation value between *CrashesANRDay1* and **DAU-Day7**. However, this is misleading; more crashes means more DAU-Day7 which should not be the case. We only see this as a probabilistic event wherein such events occuring is somewhat proportionate to how many users are active. We should see such negative events that drive the **DAU-Day7** down. We therefore propose as future work that software quality be measured for games as it plays an important role in properly predicting user virality. Consider quality-of-life features in the game

that makes it easy for the users to understand the mechanics. Some games tend to have sharp learning curves which makes them quit the game early.

To conclude our study, **DAU-Day7** can be predicted using our proposed method for successful games. JNC yields the lowest error rate which makes it ideal for our learning algorithms proposed in this paper. On the case of DNC, we conclude that it does not perform as expected on JNC and may not be practical to use such model. We propose that games wherein cases are similar to DNC, one should consider taking into consideration factors in game design and quality of the game. Out of all the learning algorithms tackled, we recommend using m5Base as the model to be used for practical application due to how well it covered different scenario in the data.

9. ACKNOWLEDGMENTS

We thank you Jakob Lykkegaard Pedersen and Thomas Andreasen for allowing us to use the dataset for Jungle Cubes and Dragon Cubes. We would also like to give thanks to Suhana Chooli, the marketing manager of Playlab Inc., which provided the details about the marketing expenses.

10. REFERENCES

[1] C. Bauckhage, K. Kersting, R. Sifa, C. Thurau, A. Drachen, and A. Canossa. How players lose interest in playing a game: An empirical study based on distributions of total playing times. In *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012, Granada, Spain, September 11-14, 2012*, pages 139–146, 2012.

[2] A. Drachen, C. Thurau, R. Sifa, and C. Bauckhage. A comparison of methods for player clustering via behavioral telemetry. In *International Conference on the Foundations of Digital Games, Chania, Crete, Greece, May 14-17, 2013.*, pages 245–252, 2013.

[3] M. Gütlein, E. Frank, M. Hall, and A. Karwath. Large-scale attribute selection using wrappers. In *Proc IEEE Symposium on Computational Intelligence and Data Mining*, pages 332–339. IEEE, 2009.

[4] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting, and C. Bauckhage. Predicting player churn in the wild. In *2014 IEEE Conference on Computational Intelligence and Games, CIG 2014, Dortmund, Germany, August 26-29, 2014*, pages 1–8, 2014.

[5] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. Special issue on relevance.

[6] Y. Wang and I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.