Aalto University
School of Business

USING MACHINE LEARNING TO PREDICT CUSTOMER LIFETIME VALUE IN A FREEMIUM MOBILE GAME

Effect of seasonal features

Master's Thesis
Tuomas Tapper
Aalto University School of Business
Spring 2022

Aalto-yliopisto
Kauppakorkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Maisterintutkinnon tutkielman tiivistelmä

Author Tuomas Tapper

Title of thesis Using machine learning to predict customer lifetime value of players in a freemium mobile game: Effect of seasonal features

Degree Master of Science in Economics and Business Administration

Degree programme International Design Business Management

Thesis advisor(s) Assistant Professor Taija Turunen

Year of approval 2022 **Number of pages** 42 **Language** English

Abstract

Freemium business model is currently largely used in the mobile gaming industry. The key idea of the model is that a game can be played for free, and revenue is generated through in-app purchases and advertising. However, the freemium model makes predicting the lifetime value of players, the amount of revenue they will generate, challenging as the revenue distribution is highly skewed and majority of revenue is generated by a relatively small group of spenders.

Predicting lifetime value of players (LTV) is one of the hottest topics in the freemium mobile games industry. Knowing how much revenue players brings games companies competitive advantage as it allows for better user acquisition optimization and financial planning, to name a few. Freemium games have several unique characteristics that set them apart from other similar fields such as online retail and traditional games such as high amount of behavioral data and high skewness of the data as only a very small share of players spend money.

This thesis has two objectives. First, different state-of-the-art machine learning models are compared to see which performs the best predicting lifetime values on a 360-day window. The models used haven been proven to be the most accurate by recent studies and include deep multilayer perceptron, random forest, gradient boosted trees as well as linear regression. The second goal of the thesis is to empirically test whether including seasonal features to the prediction dataset improves the model performance. Two different ways of using seasonal features is tested. The first approach is one-hot encoding and second applying sine and cosine transformations to make the seasonal features cyclical, representing better real-life situation. To the knowledge of the author, this is the first time these methods is used literature in freemium game setting.

Results show that deep multilayer perceptron performs the best, standing apart from the other models. This suggests that there are some complex relationships in the data that simpler models cannot capture. Against expectations, including seasonal features do not improve performance of most of the models.

Keywords customer lifetime value, LTV, machine learning, deep learning, freemium, mobile game, prediction

Tekijä Tuomas Tapper

Työn nimi Koneoppimisen käyttö pelaajien elinkaaren arvon ennustamiseen mobiilipelissä: Aikaa kuvaavien muuttujien vaikutus

Tutkinto Kauppatieteiden Maisteri

Koulutusohjelma Kansainvälisen palvelumuotoilun johtaminen

Työn ohjaaja(t) Apulaisprofessori Taija Turunen

Hyväksymisvuosi 2022 **Sivumäärä** 42 **Kieli** Englanti

Tiivistelmä

Tämän hetken yleisin ansaintamalli mobiilipelialalla freemium-malli, jossa tuote tarjotaan ilmaiseksi kaikille, ja tuottoa kerätään mainoksilla ja vapaaehtoisilla maksullisilla elementeillä joita voi käyttää esimerkiksi nopeampaan etenemiseen pelissä. Tämä tarkoittaa että suurin osa pelaajista ei koskaan käytä rahaa pelissä ja valtaosa tuotoista tulee suhteellisen pieneltä kuluttajaryhmältä, mikä johtaa epätasaiseen jakaumaan ja vaikeuttaa tuottojen ennustamista.

Käyttäjien elinkaaren arvon (LTV) ennustaminen on hyvin tärkeä aihe mobiilipelialalla, sillä se mahdollistaa muun muassa markkinoinnin tehostamisen sekä tarkemman taloudellisen suunnittelun, mikä taasen antaa kilpailuetua muihin yrityksiin verrattuna. Freemium-pelit eroavat merkittävästi esimerkiksi tavanomaisista peleistä ja nettikaupasta, koska niistä on mahdollista saada paljon dataa pelaajien käyttäytymisestä, ja data on hyvin epätasaisesti jakautunutta koska vain pieni osa kuluttaa rahaa pelissä.

Tässä tutkimuksessa on kaksi osiota. Ensimmäisessä vertaillaan aiemmassa akateemisessa tutkimuksessa menestyneitä koneoppimismalleja ja käyttää niitä tulojen ennustamiseen 360 päivän ikkunalla. Valitut mallit ovat syvä neuroverkko, random forest, gradient boosted decision trees sekä lineaarinen regressio. Toisessa osiossa testataan parantaako vuodenaikoja kuvaavat muuttujat mallien tarkkuutta. Tähän käytetään kahta lähestymistapaa: Ensimmäisessä aikaa kuvaavat muuttujat esitetään binäärisenä one-hot encoding- metodia käyttäen. Toinen lähestymistapa on transformoida muuttujat sini- ja kosini funktioita käyttäen yksikkökehälle, jolloin ne kuvaavat paremmin ajan syklistä kulkua. Tämä on ensimmäinen kerta, kun kyseistä asiaa on tutkittu akateemisesti.

Tulokset osoittavat että syvä neuroverkko on paras malli pelaajien elinajan arvon ennustamiseen, mikä viittaa siihen että datassa on epälineaarisia trendejä joita yksinkertaisemmat mallit eivät löydä. Ajan kulkua kuvaavat muuttujat eivät odotusten vastaisesti paranna tuloksia.

Avainsanat LTV, koneoppiminen, neuroverkot, mobiilipelit, freemium,

Table of Contents

List of Figures

List of tables

Abbreviations

LTV: Lifetime value
ML: Machine learning
UA: User acquisition
RF: Random Forest
LR: Linear regression
Deep-MLP: deep multilayer perceptron
CNN: Convolutional Neural Network
RNN: Recursive Neural Network
BTYD: Buy-till-you-die-model
RFM: recency, frequency, monetary value
GBM: Gradient boosted machine
GD: Gradient Descent
ARIMA: Autoregressive Integrated Moving Average
GAMM: Generalized Additive Mixed Models

1 INTRODUCTION

Predicting lifetime value of players (LTV) is one of the hottest topics in the freemium mobile games industry. Knowing how much revenue players brings games companies competitive advantage as it allows for better user acquisition optimization, player segmentation and financial planning. Freemium games have several unique characteristics that set them apart from other similar fields such as online retail and traditional games such as high amount of behavioral data and high skewness of the data as only a very small share of players spend money. Based on previous state-of-the-art academic research this paper will go through the models that have been used in LTV prediction in literature and empirically test how they perform on a freemium mobile game dataset in hand. Other goal is to investigate how including seasonal features affects the performance of the models.

LTV modeling for freemium games has been studied by several authors, the main method having been comparing the performance of different models on a dataset to find out which is the most suitable for the game in question (Sifa et al., 2015; Drachen et al., 2018; Kurki, 2020). However, as the topic has been studied relatively little in the freemium game industry, there is demand for more research to gain clearer understanding on the issue. The purpose of this study is to empirically test how the models that have been proven to work in the literature perform on the dataset in hand, to compare it and either confirm their findings or present new information.

Freemium mobile games are unique in the way they are designed and how players interact with them, so it is crucial to find out how different models perform on this data and how the performance differs from previous studies. This provides insights on whether certain models seem to be generally better suitable to LTV prediction or if it depends on the characteristics of a particular game. A non-academic goal of this study is to deliver the company that provided the dataset a better understanding of their game in terms of LTV.

The effect of seasonality in sales has been widely known in industries such as online retail and finance, as customers behave differently depending on the day of the week and season, for instance. It is also natural that sales of some products and categories such as sunscreen depend heavily on the season. However, there is little academic study on if freemium games

have seasonal variation and if this information could be used to improve LTV prediction accuracy. Such empirical study on the effect seasonal features, such as day of the week and week, for a freemium game is well justified considering that it is known fact, that there tends to be variation in how customers behave on different days of the week or month in online retail and finance, for instance (Bussiere, 2016; Zhou et al., 2016; Whang et al., 2020).

Chan et al (2019) discovered that using seasonal features when training machine learning models predicting sales of companies on an E-commerce platform remarkably improved the accuracy. This suggests that it could also improve freemium game LTV predictions, as the behavior of players likely is affected by things such as holidays and opportunity costs of playing. Seasonal features have been used previously in machine learning for example by Chamberlain et al. (2017) and Guitart et al. (2017), but to my knowledge there are no previous published studies comparing how they affect the results of machine learning LTV models in freemium games, which indicates that there is a clear knowledge gap that this thesis could fill.

Also considering that online retail has been used as a reference point by several authors, such as Kurki (2020) and Burelli (2019), it is justified to expect that a phenomenon that is so strong in online retail could also affect the LTV performance on a freemium game. According to Fader and Hardie (2005), online retail also belongs to the same category of noncontractual continuous business setting, making it suitable for comparison.

The research questions of this thesis are:

1. What models have performed the best on predicting the LTV in freemium games?
2. How do the findings of previous research compare with the game used in this study?
3. What is the effect of seasonal features on performance of LTV models?

The results show that deep-MLP stands apart from the other tested models, gradient boosted decision trees, random forest and LASSO regression. This implicates that there are complex nonlinear relationships in the data that simpler models cannot detect. Using the tested methods to include seasonal features does not improve the performance of most models.

2 LITERATURE REVIEW

2.1 Freemium games

Freemium approach in mobile games is currently the dominating business model in the mobile gaming industry. The idea is that the game is free to download, but players can make in-app purchases (IAP) to gain virtual currency or improve the experience by removing ads, for instance. Majority of the highest grossing mobile games such as “Candy Crush”, “Gardenscapes” and “Clash of the Clans” use the business model. The model has gained popularity also outside mobile gaming, “Counter Strike: Global Offence”, “PUBG” and “Call of Duty: Warzone” being some of the well-known titles on other platforms. There is a lot of variation within the freemium games in terms of monetization and what benefits players get with their purchase. In some games players can advance in the game by purchasing and using in-game currency, whereas in others self-expression and collectability of characters or in-game items are the main purpose of doing in-app purchases. Also the shares of IAP and ad revenues vary across games and genres. All these kinds of aspects affect the behavior of players and thus LTV predictions.

Freemium games differ quite remarkably from traditional video games due to larger variance in purchase regularity and amounts, a better comparison point being commodity stores (Burelli, 2019). Due to the game being free to download the data tends to be highly imbalanced, as a large majority of the players do not spend any money in the game. Also within the spending group the highest percentiles create the majority of revenues. Sifa et al. (2018) mention that in their analysis the top 20% and 10% of spenders generate 80% and 60% of revenues respectively, making the relatively small group crucial for the company. These factors and the nature of the freemium model also make predicting the LTV challenging as purchases can happen at any time and the amounts vary largely, which reflects to the business planning as a higher degree of uncertainty in estimates.

2.2 Analysis setting and business context

Fader & Hardie (2009) present four different categories for customer relationships based on two variables: “Type of customer relationship” and “Opportunities for Transactions”. These can be further divided into two subcategories. Type of customer relationship can be either “contractual”, for example subscription, where the company knows when the relationship ends. Known examples of this are subscription services, for example Spotify and Netflix. In

"noncontractual" setting the company cannot know if the customer has churned or is still coming back. Example of this could be a supermarket, the shopkeeper does not know when and if the customer is coming back and whether not seeing them for a while is caused by them moving away permanently or just having been on a holiday.

The second variable is called opportunities for transactions and refers to when purchases can be made. In case it is "continuous", the purchases can happen at any time. An example could be the supermarket, where one can shop at any time and spend almost any amount. If purchases can only happen at a certain time, the case is referred to as "discrete", an example could be an event ticket that can only be purchased prior to the event.

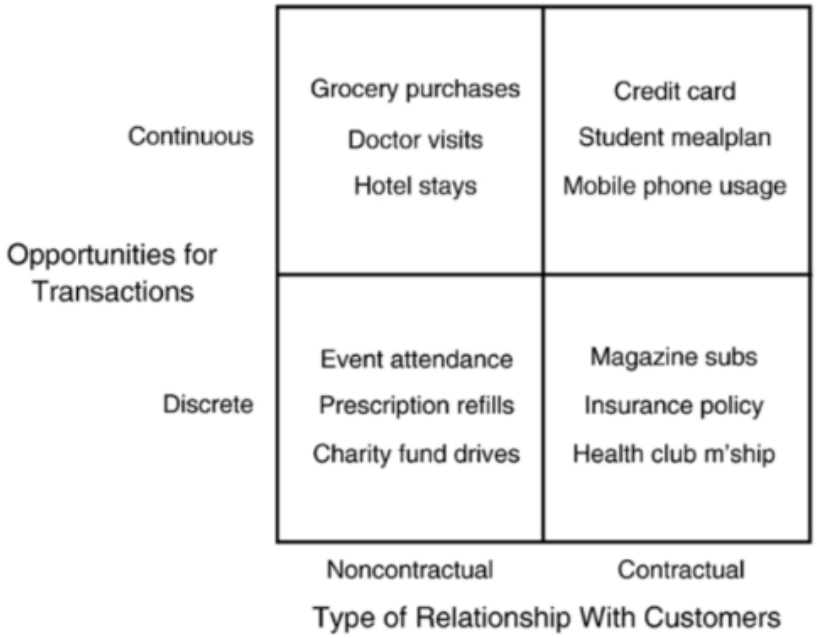

Figure 1: Categories for customer classification (Fader and Hardie, 2009)

According to the classification Fader & Hardie (2009), freemium games could be labeled as noncontractual and continuous, as the purchases can happen at any time and the amounts vary. Some games may however also include elements that could be classified as discrete and contractual, but usually the emphasis is on the former category. Fader & Hardie emphasize that a prediction model should be created for a single category and not mix them (2009). Of the four categories, the noncontractual and continuous is the hardest to predict, as their uncertainty whether the customer has churned and also the possible next purchase's time and amount are unknown.

2.2 LTV and use cases

2.2.1 Lifetime value definition

There are a few different ways the LTV is defined in the literature. Burelli (2019) defines LTV as "*Customer lifetime value (CLV or LTV) refers broadly to the revenue that a company can attribute to one or more customers over the length of their relationship with the company"*. Pfeiffer et al (2005) mention that LTV should be interpreted as the future cash flows of a player without including the acquisition costs. Berger and Nasr consider LTV from UA standpoint to be *"maximum profitable acquisition cost"*. Weinberg and Berger (2011) also include the revenue of people a player brings to the game, calling it "Connected customer lifetime value". Burelli (2019) mentions that it is often useful to use a fixed horizon, for example 1 year, for the prediction. In this thesis the lifetime value will be defined as the revenue that acquired players generate to the company over 360 days after downloading the game.

2.2.2 LTV use cases

Burelli (2019) states that marketing is usually the main use case for LTV models. As other uses he lists budget planning, following growth of games, user retention and customer segmentation. They continue that knowing the LTV allows also tailoring the games in order to improve monetization:

- UA: invest on campaigns with highest return on investment
- Customer service: The companies can prioritize high spending customers to keep them happier and more engaged
- In-product advertising: Not showing ads to high spenders can improve their engagement and spending, whereas more ads can be shown to zero spenders to monetize better
- Pricing and promo: Target promotions to low and zero spenders, whereas high spenders are excluded from these

User acquisition

User acquisition is defined by Burelli (2019) as "*all activities aimed at getting new customers to start using the service or buying the products offered*". One of the main motivations for predicting the LTV is that it allows for more efficient user acquisition (UA) operations. Knowing the value of players helps ensure the costs of UA are lower than the LTV, making it profitable. This is getting increasingly important as UA costs per install (CPI) have risen

significantly in recent years and being able to predict the LTV as early as possible brings great business value and is important for the success of a game (Sifa et al., 2015). The definition of LTV is crucial when analyzing the profitability of UA, as there are several approaches to it. The most exclusive option would be to only include the IAP revenues an acquired player generates, whereas Weinberg and Berger (2011) attribute all the revenue that a player brings, including the revenue of people that the player brought to the game. From the UA profitability point of view, the difference between these extremes can be drastic.

2.3 Methods used to predict LTV

LTV estimations have been usually done in three different ways: 1. Heuristics based on expert knowledge (Recency, Frequency, Monetary (RFM) most prominent), 2. Stochastic models and 3. Machine learning models (Valdivia, 2021). Burelli (2019) brings up that there is no industry standard practice for predicting LTV, and different methods have been used such as deep learning or statistical models.

A major problem for predicting LTV for freemium games is that the data is highly unbalanced due to low share of spenders (Sifa et al., 2015). Valdivia (2021) also emphasizes the fact that even within spenders, the behavior of the sub-population of high spenders stands apart and recommends treating them separately from others. Sifa et al. (2018) have similar findings as they note that the top 20% of spenders generate 80% of revenues. The literature presents several ways to deal with the skewedness. Sifa et al. (2015) used SMOTE, *synthetic minority oversampling technique-Nominal Continuous* to improve the LTV prediction and decrease the imbalances in the data. SMOTE is also used by Kurki (2020) in predicting churn, which is closely related to LTV. Chen et al. (2012) used tweedie loss to deal with the zero inflated data, whereas Sifa et al. (2015) seek to exclude the zero spending cohorts by applying a classifier model on the data before doing the regression.

There is no industry standard for observation period, but seven days seems to be most used in freemium games (Sifa et al., 2018; Drachen et al., 2018; Kurki, 2020). Sifa et al (2018) point out that using one week's data is good in a sense that there is weekly periodicity in player behavior. Being able to predict the LTV as early as possible brings companies competitive advantage (Sifa et al., 2018). Burelli (2019) mentions that it is often useful to use a fixed horizon for the prediction, the usual choice being 1 year. The problem with this approach is that one must wait a year to find out how accurate the predictions are. However,

Kurki (2020) uses a remarkably shorter window, predicting the day 30 LTV. This has the advantage of having a shorter feedback loop, but the drawback is that revenues generated after that are excluded.

2.3.1 Heuristics based on expert knowledge

RFM (recency, frequency and monetary) models can be used to predict LTV. The letters stand for:

a. Recency, time since latest purchase
b. Frequency, number of purchases during a certain period
c. Monetary, the total amount spent during a period.

RFM models have been used for over 40 years and have been very common due to their easy usability and simplicity (Rathi, 2011). Fader et al. (2005) have proven that RFM can also be used as a base for LTV prediction by using iso-value curves (also based on pareto/NBD). They used the model to predict the LTV of customer groups in an online music store and were able to find non-linear relations that could not have been discovered by data observation.

There is plenty of research on LTV in the field of marketing. A widely used group of probabilistic models is called “buy till you die” (Fader and Hardie, 2009). The models are based on recency and frequency and thus related to the RFM-framework. Advantages of this approach are the ease of implementation and them only requiring transactional data for training the model. Fader & Hardie (2009) argue that one should start modeling a problem with the simplest approach and add complexity only if the model is too simple.

Schmittlein et al. (1987) introduced the most discussed model from this group, Pareto/NBD. Originally it was used to monitor customer base size and development but has since been modified to be used in LTV as well after introducing monetary features. Since it was published, numerous variations of the model have been presented to take into account aspects such as purchase regularity and day zero churn.

Chen et al (2018) concluded that parametric models Pareto/NBD, BG/NBD, MGB/CNBD-k had issues dealing with large data and they only considered spenders, leaving out majority of the data. They could also only use transactional data in the model (Chen et al., 2018). This is not optimal as Sifa et al. (2018) mention that non-transactional data contributes to

prediction accuracy, even though transactional data has the highest importance. The situation is similar with stochastic models, which are also based on purchases (Sifa et al 2018). Based on these findings, it seems like using above -mentioned models for other than benchmarking is not viable, as the behavioral data plays such an important role in the prediction accuracy. Also considering that the amount of data is expected to increase in the future, the ability of models to process large datasets is crucial.

De Carvalho Santos (2020) mentions that even though average and Pareto/NBD based models have been widely used, machine learning approaches have gained popularity lately. Machine learning models have certain advantages such as allowing a computationally viable solution considering the large datasets and including wider range of features more easily. ML models can also be laborious and expensive to develop and simple changes such as changing the prediction timeframe for different scenarios can require relatively large efforts (Valdivia, 2021).

2.3.2 Classification and regression

There are several approaches to predicting LTV in the literature. Sifa et al. (2015) state that building an LTV model has three steps: predicting 1. the number of spenders, 2. the purchase count and 3. the monetary value. In the first problems Sifa et al (2015) used two-step approach, considering it as classification and regression problem. In the former they used decision trees, random forest and support-vector machines, and poisson trees for the latter. In this study random forest had the best accuracy. Also Drachen et al (2018) used a similar classifier and regression-approach to first identify spenders and then use that as a starting point for regression. The benefit of this approach is that as Sifa et al. (2018) discovered, the false positive zero spenders explain most of the error and classification could help with this issue. However, there are no studies comparing if such a multi-step approach is more accurate than a single regressor.

2.3.3 Linear and LASSO regression

Regression analysis is used in modeling relationships between variables (Montgomery et al., 2021). Due to its simplicity and applicability in numerous problems and fields of study, it is often considered the most used statistical model. The idea of the model is fit a line that minimizes the L2 norm, which is the total squared distance between the prediction and actual values.

Lasso (Least Absolute Shrinkage and Selection Operator) is a linear regression model with L1 regularization. Adding the regularization term enables the model to set weights to different features and thus allows the model to generalize better compared to unregularized linear regression (Muthukrishnan and Rohini, 2016). Lasso allows coefficients of features to go to zero, which helps eliminate correlated features from the dataset. Hyperparameter alpha can be used to control lasso. High alpha values result in more features being given zero weight and vice versa. Setting alpha to 0 is like normal linear regression.

Linear regression is used to predict LTV by many authors (Sifa et al. 2018, Kurki, 2020; Runge, 2020) and the results are mixed. Sifa et al., (2018) finds LR, RF and deep-MLP to have comparable LTV prediction performance, with the latter being slightly better than the other two. Also Kurki (2020) proposed Linear regression as the best performing model with RF. However, Runge (2020) stated that linear regression is too simple of a model to fit to the complex game data. Based on the literature, it seems like linear regression is a promising option depending on the data used.

2.3.4 Random Forest and Gradient Boosted Decision Trees

Random Forest

Random forest is an ensemble learning method, consisting of separate decision trees.
It uses “bootstrap aggregating” or simply “bagging”- method to generate multiple training sets. In bagging, one draws samples from the dataset randomly, creating a subset that is used to train a decision tree. This allows getting different decision trees while still using the whole dataset, which can improve the performance of the ensemble on new data. The final output of the model is the average of the different trees, which balances out outlier predictions. As the trees are different, the model is also less prone to overfitting.

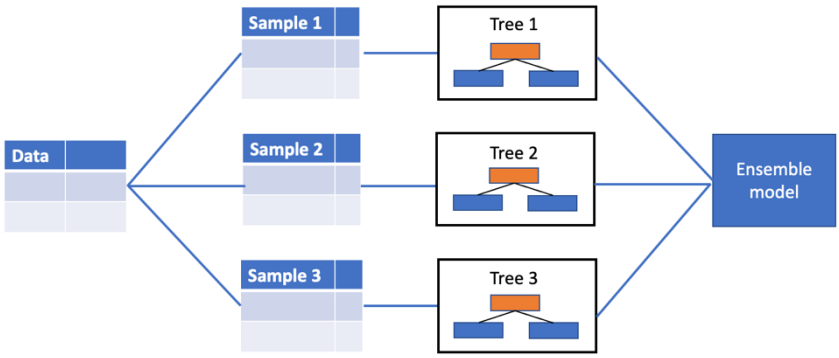

Figure 2. Bootstrap aggregation for random forest

However, an issue with decision trees is that they tend to overfit easily when the trees get deeper. This can be mitigated by combining them, so the errors of individual trees balance out. The basic idea is to put together the results of multiple decision trees, which decreases the generalization error as their combined output is used in the result. If random forest is used in regression, the mean of the outputs of the forest is used.

The most popular version of Random Forest was introduced by Breiman (2001).His novel addition was also using random features to train the decision trees. This makes the ensemble more diverse as the weights of single features are decreased. The left-out data is then used in measuring the generalization error by providing out of bag- estimates. Breiman (2001) argues that using the out of the bag estimates, allows one to train the model on the whole dataset, instead of using a separate test set. This is useful especially when dealing with small datasets.

Gradient boosting and XGBoost

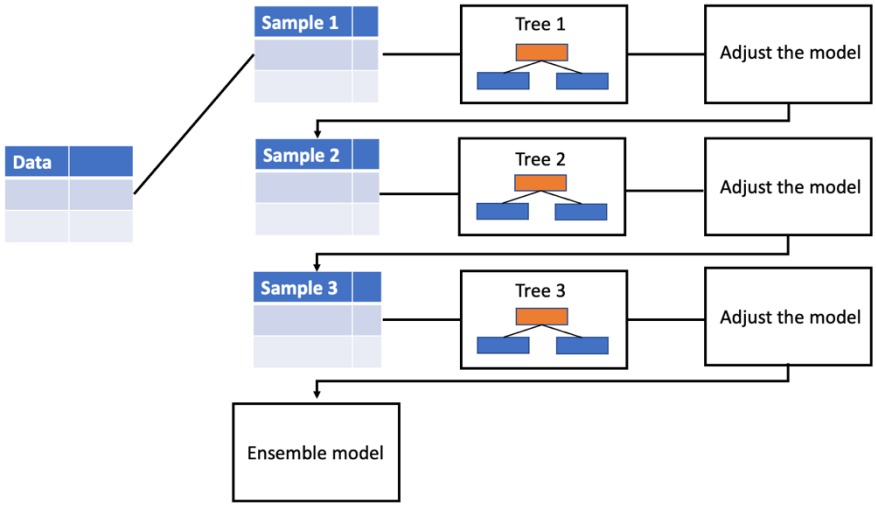

Figure 3. Sequential boosting approach

Gradient boosting is an ensemble method based on boosting, which was first introduced by Friedman (1999). The model uses an ensemble of decision trees, but unlike in random forest, the trees are built on each other instead of being independent.

The name gradient boosting comes from the fact that gradient descent is used to minimize the error. The model iterates through all training examples in order to find the place where the curve is the steepest and hence gives the maximal improvement with regard to error. The goal of the model is to find the zero derivative, which is the optimum for the model.

The step size of how much the model will update each iteration can be adjusted with the learning rate. However, a major issue for gradient descent is that it does not have a way to distinguish between the local and global minima, which may lead to converging without reaching the optimal loss.

This paper uses an implementation of gradient boosting called Extreme Gradient Boosting (XGBoost) that was introduced 2014 by Chen and Guestrin (2016). The main difference to preceding gradient boosting models is regularization, which helps avoid overfitting and thus improves the general performance. Another change is that XGBoost processes the trees in parallel to improve computational efficiency (Chen and Guestrin, 2016). The model and is

considered among the best performing models in the field of data science, having been proven successful in Kaggle machine learning competitions since its release.
XGBoost starts with a baseline prediction and builds a tree based on the residuals, which is the distance between true and predicted value. The residuals of the initial tree are then used to make predictions and again, the residuals are used to create the following tree. This iteration process is continued and allows the model to improve.

XGBoost uses similarity scores for the leaves to split the trees in an optimal way. Different trees are evaluated by calculating gains each possible split would bring and then choosing the one that has the highest gain. Splitting is continued further if the gain is above the set threshold, gamma, that is used to regularize the model and avoid overfitting. In case the gain is below gamma, the leaf will be pruned. XGBoost has also other regularization hyperparameters, such as lambda, which is used to decrease the sensitivity of model to single observations.

It seems that tree-based models such as random forest and extreme boosted decision trees (XGBoost) and deep learning models are the promising candidates for LTV prediction at the moment. Sifa et al. (2015) bring up that decision trees are preferred as they allow for better interpretability of results compared to neural networks. Also, Kurki (2020) favors supervised methods such as random forest over deep learning models for similar reasons.

Miettinen (2019) predicted the next purchase amount and time for a freemium mobile game and found that random forest had the highest accuracy compared to Artificial Neural Networks. He considers that the success of nonlinear and complex models indicates that purchase behavior is nonlinear by its nature. However, Kurki (2020) had conflicting findings as linear regression performed almost as well as random forest. In the study by Drachen et al. (2018), random forest performed better than XGBoost in LTV prediction in terms of NRMSE and R2. In the study, the R2 values ranged from 7% to 40% depending on the segment used.

2.3.5 Deep Learning

Deep Multilayer perceptron

Deep Multilayer perceptron (deep-MLP) deep learning model, which takes its inspiration from the working mechanism of a human brain, where a complex net of neurons are used to

process information. The perceptron on which neural networks are based on was first introduced by Rosenblatt (1958) and even though deep learning models have existed for a long time, the recent increases in data volume and easier access to computational power explains why it has become popular recently. Neural networks tend to be computationally heavy due to their complexity.

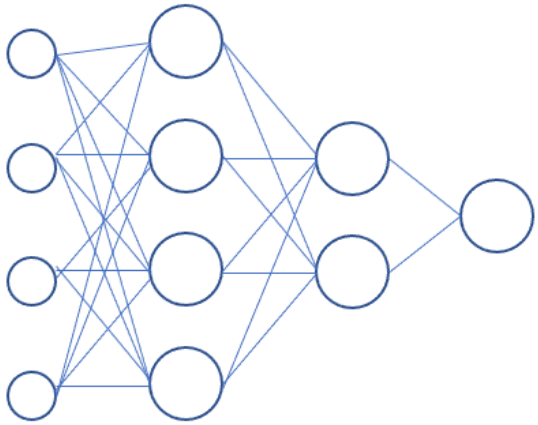

Figure 4. Neural network structure with fully connected layers

A deep-MLP consists of input and output layers, activation functions as well as hidden layers, which are why it is called 'deep'. MLP consists of fully connected layers, meaning that each neuron in a layer has connection to every neuron in the following layer as one can see in the figure 4. Due to the structure of the model, it can fit to nonlinear data. Neural networks are often used in complex tasks such as image and speech processing and can be used in both regression and classification.

MLP is based on forward- and backpropagation. First the weights are initialized, whereafter the actual "learning" happens through backpropagation, where gradients are used to update the weights. Iterating this loop and adjusting the weights allows the model to learn complex relationships that simpler models cannot detect. Flexibility is one of the key advantages of this model family, as the high number of hyperparameters allow for customization. On the other hand, this makes fine-tuning the model laborious and requires more in-depth knowledge on the model compared to simpler models.

A shortcoming of the model family is that they are considered to be “black box”- models referring to the fact that due to the model structure, it is hard to interpret why the model gave certain results. Neural networks are also the most difficult model class to fine tune and implement as choosing the architecture such as number of layers must be done manually (Runge, 2020). He also notes that overfitting is a major issue when dealing with neural networks due to their flexibility.

Convolutional neural networks (CNN) are deep-learning models that use different architecture compared to MLP. The key difference is that CNN uses convolutional layers, which allow for processing unstructured data in an efficient manner and is the reason why the models are used for computer vision.

Chen et al. (2018) found that deep learning models performed better on predicting the LTV compared to parametric models. The ability of deep learning models to handle behavioral data and identify high spenders set it apart from the Pareto-based models. Further, they found that CNN performed slightly better than multilayer-perceptron. CNN also had other positive characteristics such as using raw data and faster computation compared to MLP. Use of raw data makes modeling simpler as there is no need for feature engineering, which is time consuming. CNN performance is also expected to enhance with more data, making it promising considering that the amount of data is likely to increase in the future Chen et al (2018). They also noted that CNN is good for detecting high spenders, who were neglected by the parametric models.

Sifa et al. (2018) used deep multilayer perceptron combined with SMOTE (synthetic minority oversampling) to predict LTV and high spenders and found that SMOTE improved the performance, as the data is very unbalanced due to small percentage of spenders. It outperformed other tested models, random forest, linear regression, and decision trees. Higher spenders had the highest prediction accuracy, whereas false negative zero spenders contributed the most to the error. Sifa et al. (2018) also noted that the amount of money spent was the most important feature in predicting the LTV. They note that the top 20% and 10% of spenders generate 80% and 60% of revenues respectively, making the relatively small group crucial for games companies (Sifa et al., 2018).

Similarly to Sifa et al. (2018), Runge (2020) compared different models when predicting high spenders. Also in his study, deep-MLP combined with SMOTE outperformed Random Forest and Linear Regression allowing one to identify 83 % of the top 30% of spenders with 7 days observed data. He also used three different observation periods: app download as well as with 1- and 3-days data. Runge concluded that clickstream data from mobile apps allows separating the spenders from non-spenders only poorly.

Miettinen (2019) and Runge (2020) had similar findings in their studies, both concluding that the capability of neural networks to learn non-linear rules flexibly is the key reason for their relatively high performance. They also noted that in-game behavior does not correlate with purchase behavior as many non-spenders behave very similarly to spenders. This is a big problem from LTV modeling point of view as if there is no way to tell the two groups apart, it will lead to high errors.

Bauer and Jannach (2021) propose a Recurring Neural Network (RNN)- based model for LTV prediction in e-commerce, which allows for using the past data as input for coming layers (Encoder-decoder sequence-to-sequence RNN). RNN: s are commonly used for processing sequential and time-series data as the output of neuron depends also on the proceeding values. Another mentioned advantage is that this model does not require manual feature engineering to learn seasonality and trends. To improve performance, they also use GBM to predict the total sum of all purchases. Bauer and Jannach (2021) state that in addition to ML, LTV predictions can be seen as a time series forecasting problem. They mention that instead of directly modeling the total revenue as in case of ML, first predicting the purchases with time series and then aggregating the results to get the LTV allows to capture periodic effect like seasonality more easily.

2.4 Seasonality

There are clear indications that seasonality and factors such as day of the week, season, paydays, and holidays affect the behavior of customers in different industries. Kooti et al. (2016) note that there are weekly and daily consumption patterns in customers' online shopping behavior. Furthermore, the effect is also verified to exist by Zhou et al (2016) and Hwang et al. (2020), who found that people spend more on weekends compared to weekdays. The day of the week effect has been also proven to exist for example in the stock markets (Runge, 2020).

There are few studies taking the seasonality into account in predicting purchases in mobile games and especially in the case of LTV predictions. Guitart et al. (2017) included day of the week and month as features when predicting purchases and playtime using time series, autoregressive moving average (ARIMA) and generalized additive (GAMM) models. However, even though it was found to work in short window predictions (30 days), they mention that it gets less accurate when the window is extended. Also, Sifa et al. (2018) mention that there is weekly periodicity in the data, but do not go deeper into the topic.

Whang et al. (2020) used several time-based features when studying purchasing behavior in retail; day of week to capture the variation in spending on different weekdays, week to include seasonality and year to represent long-time changes. Also Chamberlain (2017) uses date as a feature and stresses out the importance of using 12-month training data to ensure the seasonality is represented in their game data.

Chen et al (2020) predicted the sales of different retailers on an E-commerce site Tmall.com. They designed two mechanisms for forecasting sales: seasonality extraction and distribution transformation. They found that including these as features in commonly used regressors, like neural networks, gradient boosting decision trees, improved the prediction performance significantly. After clustering different retailers, they apply Fourier basis functions to extract the seasonality from the different groups. They discovered that the aggregate sales of all retailers follow Tweedie distribution, a combination of Poisson and gamma distributions, after a logarithmic transformation. This is caused by the fact that retailer status; whether they are open, follows Poisson distribution, whereas sales obey gamma distribution. Using Tweedie loss in regression improved the prediction results. Tweedie loss is also commonly used in the insurance industry (Chen et al., 2020). They state that the seasonality extraction and label distribution can be applied to most forecasting models, neural networks and gradient boosting decision trees included.

The most intuitive way to include seasonal features would be adding the day of the week data encoded as numbers from 1-7 for each weekday, for instance. The number of the week and month could be added in a similar manner as well. The problem of this approach is that it creates a hierarchy that does not exist in real life and the model processes like there was a trend. This problem is often solved with one hot encoding, where a column is created for

each category and the value is binary. In case of weekdays, this could mean that column "weekday1" is set to 1 if even occurred on Monday and 0 otherwise.

As this study is done using cohorts instead of single players, using more granular seasonal data such as hour of transaction is not used but in case of player level predictions, including them would be justified. However, using such simple encoding might not be optimal as time is cyclical by its nature. For example, December the 31st and January the 1st are following days in real life, whereas for the model they seem to be an year apart. Chakraborty and Elzarka (2019), solve this problem by doing sine and cosine transformations for the features. This allows the datapoint to be mapped in a realistic manner, which follows the real-world setting. As one can see from the figure 5, only doing sine transformation correctly presents time as cyclical, but this presents a new issue where two different units (00:00, 12:00) output the same value. Thus, it is necessary to add cosine transformation, that can be used to create each timepoint an unique combination of abovementioned features Chakraborty and Elzarka (2019). These transformations are applied to all the seasonal features used in this paper.

Using numerical representation of the seasonal features creates the risk of model using them "incorrectly" as in real life the numerical value of for example day of the year does not mean anything, but the model might use the ordinal values and fit to noise. On the other hand, using numerical features allows for getting insight on how close some dates, for instance are. If used as fully categorical metric, there would be no way to tell whether days follow each other or if they are far apart. Other shortcoming of this study is that the difference between predicting purchases using seasonality and adding seasonal features to ML models predicting aggregate 360-day revenue with 7 days observations is quite big.

In figure 5, the top-left corner represents the data before the transformation from the plot one can see that a time just before and after midnight are very far from each other, which does not reflect the reality. On the next two plots (b,c) one can see the sine and cosine transformations respectively. Both are representing time in cyclical manner but only for a 12-hour window. Finally, in the last one (d) cosine and sine values are plotted and one can see that the data now has cyclical form, and each point has unique coordinates.

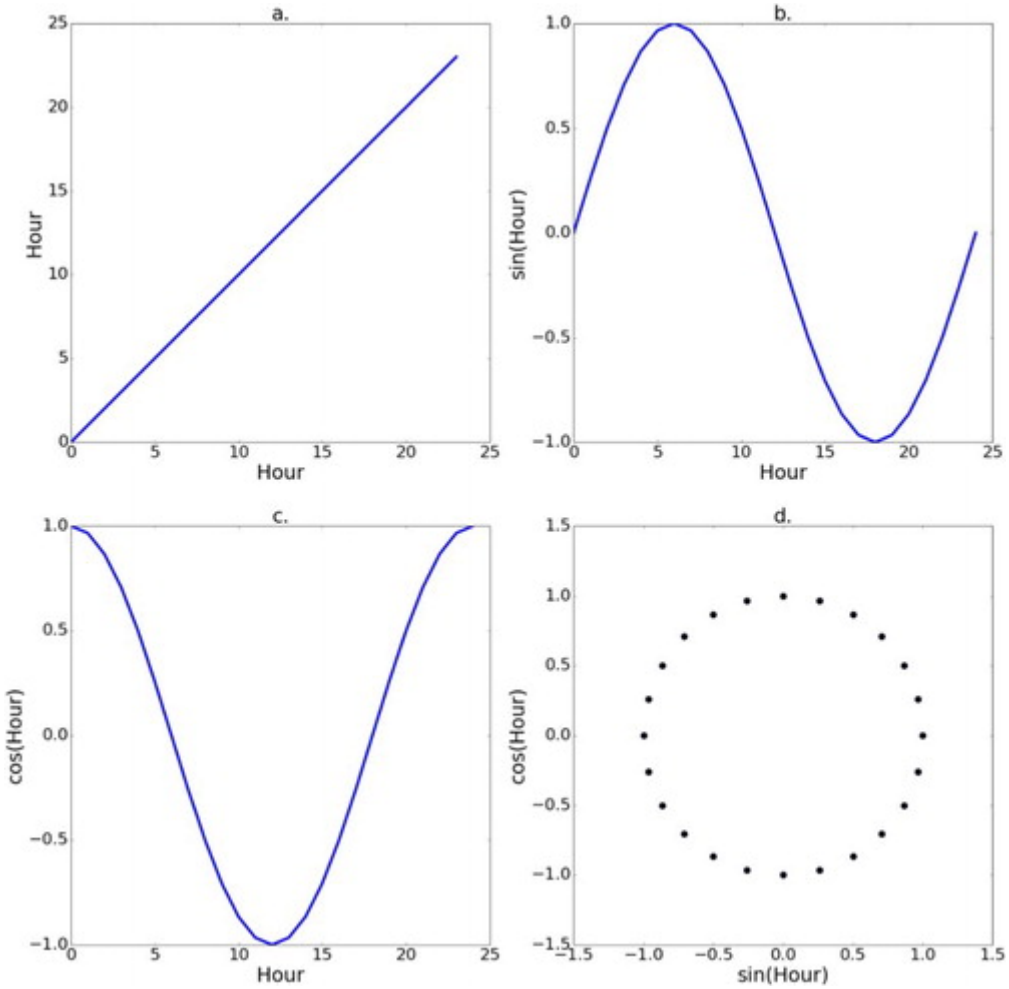

Figure 5. Sine and cosine transformations for time feature (Chakraborty and Elzarka, 2019)

Table 1. LTV modeling for freemium games in literature

Authors	Models	Observation period	Target day	Recommended model
Drachen et al. (2018)	RF, XGBoost	7	Not stated	RF
Chan et al. (2018)	DNN, CNN, Pareto/NBD, BG/CNBD, MGB/CNBD, BG/CNBD-k	Months, not defined exactly	Not stated	CNN, DNN
Sifa et al (2015)	Clf: RF, REG: Poisson tree	1,3,7	Not stated	RF classifier + Poisson tree regressor
Sifa et al (2018)	RF, LR, DT, deep-MLP + SMOTE	7	360	Deep-MLP
Guitart et al. (2017)	ARIMA, XGBoost, GAMM	0	<90 days	GAMM, ARIMA
Runge (2020)	LR, RF, deep-MLP	7	360	Deep-MLP
Miettinen (2019)	RF, SVM, deep-MLP	Not stated	Next purchase	Deep-MLP
Kurki (2020)	LR, RF, GB, MLP, Pareto/GGG, CBG/NBD	1,7,14	30	LR, RF

3 METHODOLOGY

The empirical part consists of two subtasks. First, different models are used to predict the LTV, whereafter the focus is on comparing these results with and without the seasonal features as well as feature selection. I will first present the data and case, which is followed by explaining the choice of methods and models, as well as giving a short introduction on how they work.

This paper will use 12 month transactional and behavioral data from a free-to-play mobile game, that is delivered by a Finnish mobile games company. The data includes discrete and continuous features that are normalized to improve performance. The prediction window is from day 7 to day 360 to balance between getting actual data to base the predictions on and getting the results early.

The empirical part will be implemented using Python 3-programming language and its libraries; Pandas, Numpy, XGBoost, Sklearn and Tensorflow. Pandas and Numpy are used to process the data whereas the rest are used to train, tune, and test the models. All the above tools used are open-source and can be freely accessed by anyone. They are validated through peer review and commonly used in the industry and academia.

The machine learning models compared are linear regression, random forest, gradient boosted decision trees and multilayer perceptron, The respective models were chosen as they have been commonly used in the literature to predict LTV and are expected to bring the best results. As Fader & Hardie (2009) argue, one should approach a prediction problem first with the simplest model and move to more complex ones if needed. This far machine learning and probabilistic models have been compared in LTV prediction by Chen et al. (2018) and Kurki (2020) and due to the low expected performance in these studies, this paper will concentrate on machine learning models.

3.1 Data preparation

3.1.1 Dataset

This study uses a 12-month dataset from a freemium mobile game. The data is limited to players from the United States who came from advertising campaigns. It contains 13715 datapoints, in this case cohorts, each having 1-4426 players. A cohort is defined by players that download the game on a certain day and are attributed to same marketing campaign.

Using cohorts allows for faster training as the number of datapoints is remarkably smaller. It might also make predictions more accurate compared to player level as some variance is cancelled out due to the aggregation. Features are chosen to capture both transactional as well as behavioral trends of the cohorts to give the models the optimal grounds for predicting the lifetime value.

The data is queried for the first seven days from when the player downloaded the game, so that each metric is broken down on daily level (d1_revenue, d2_revenue... d7_revenue). A single data point consists of players that downloaded the game on a certain date and are attributed to same marketing campaign, network as well as operating system. An example of cohort could be players who downloaded the game on iOS device on 01/01/2021 in the United States and are attributed to campaign X that was shown by network Y. Before training the models, the data is scaled to values between (0,1) using MinMaxScaler from Sklearn-library. This is done to avoid any decrease in model performance that having features with very different ranges could cause.

To stay consistent with previous literature, this paper uses cross validation. For each of the five folds, 4/5 of the dataset was used to train one instance of each compared model, and predictions are made with the remaining 1/5. A shortcoming of this is that the setting differs from an actual setting that would be used by a company, and the results should only be used to compare the models. One instance of each model is trained

The data is highly skewed, as vast majority of players do not make any purchases. As can be seen in figure 6, even within spenders, majority of cohorts bring relatively little revenue, and there are very few cohorts in the highest spend buckets. From the model perspective this also means that there are relatively few datapoints on the high spenders, which also decreases the model's ability to identify such cohorts. Such outlier cohorts also increase the errors remarkably and make interpreting the error metrics difficult. Especially RMSE, which is affected more by large differences between prediction and actual values, can react to these cohorts. The errors are also likely very different for the highest spenders when compared to non-spenders, for instance.

Also, the revenue model of the game affects LTV predictions, for example when it comes to the ad revenue share of total revenue. Ad revenues tend to be more deterministic, as the game

creator can control the amount and frequency of ads shown. This also means that the ad revenues are easier to predict in contrast to in-app purchases, which happen more randomly and require more activity from the user.

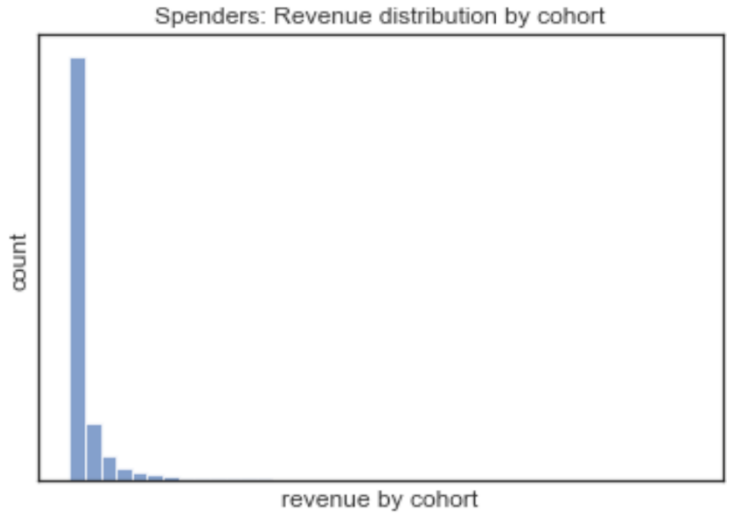

Figure 6. Revenue distribution for spender cohorts

Figure 7. shows that there is clear pattern as installs increase during weekends. This shows that weekends affect the players' behavior, which supports the hypothesis that the model would improve if more features capturing this kind of changes are included.

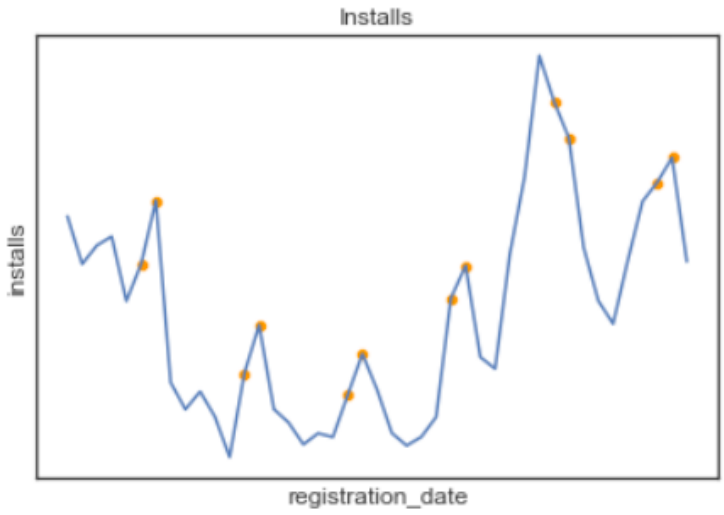

Figure 7. Installs by registration date, Saturday and Sunday marked with yellow dots.

3.1.2 Features

The basic dataset consists of 83 features. All the data is from the first seven days since install, but some features are on daily level (marked with d1-d7). Each row in the dataset represents one cohort.

Table 2. Model features

Feature	Description	Datatype
Registration date	Day the game was downloaded	Datetime
Revenue (d1-d7, sum, avg, median, std, 90th percentile)	In-app purchases, by day and total statistics	Float
Ad revenue (d1-d7, sum, avg, std, 90th percentile)	Ad revenue generated	Float
Time played (d1-d7, sum, avg, median, std)	Amount of time spent on the game	Float
Purchase count (d1-d7, sum, avg, median, std)	Number of purchases	Float
Session count (d1-d7, sum, avg, median, std)	Number of sessions	Float
Last seen d1-d7	Count of player that have not played after dX	Float
Ad impressions (sum, avg, median, std)	Number of times ads have been seen	Float
Ad clicks (sum, avg, median, std)	Number of times ads have been clicked	Float
Video reward (sum, avg, median, std)	In-game rewards given for viewing ads	Float
Installs	Total number of installs	Integer
Spenders	Number of purchasers	Integer
Seasonal SIN/COS transformed		
Week number with sine transformation	-1 to 1	Float
Week number with cosine transformation	-1 to 1	Float
Month number with sine transformation	-1 to 1	Float
Month number with cosine transformation	-1 to 1	Float
Seasonal one-hot encoded		
Month (one-hot encoded)	1 or 0	Integer
Week (one-hot encoded)	1 or 0	Integer
Weekend (one-hot encoded)	1 or 0	Integer

3.1.2 Challenges of using past data in model testing

As freemium mobile games are optimized and changed constantly, one should be cautious when testing models with historical data. In some cases, there can be quite remarkable changes in the monetization of a game, which could lead to the historical data before the change to be outdated if the player behavior changes. However, evaluating the magnitude of such effect is also very hard as one must wait 360 days to get the real numbers and evaluating the effect of past changes leaves the question open whether this time is similar.

Using irrelevant past data can be harmful as the predictions do not match reality. Also considering that one will not know this earlier than a year after, the damage caused by making decisions based on irrelevant numbers can be remarkable. It is also noteworthy that even if such risks are clear for the data scientist developing the model, they might not translate to the actual user, who may take them as granted, and it is important to communicate this clearly.

Due to abovementioned reasons, it is vital to take changes in the game into account when training and testing the model. Also, prediction windows should be chosen according to how the game monetizes and what one wants to find out. Shorter prediction windows may allow for higher accuracy but increases uncertainty of what happens after. For example, in the case of user acquisition, where knowing whether an investment is going to break even is crucial, the prediction window should be adjusted according to it.

3.2 Models

Manual testing showed that in many cases the default hyperparameters worked well in other models than deep-MLP. Deep-MLP has a wide variety of hyperparameters to tune. Among most important aspects are the network size and shape, optimizer, activation functions, regularization as well as initialization of the model.

During the model development various hyperparameters and architectures were tested, but eventually similar model to what Chen et al. (2018) and Kurki (2020) used, gave the best results. Changing the number of layers or their size did not improve performance. Different regularization methods, such as including dropout layers and adding L1 and L2 regularization to the layers did not improve the performance. Out of loss functions (MAE, MSE, MSLE), MAE performed the best with both evaluation metrics, MAE and RMSE.

The model has an input layer of same size as the data, three hidden layers with 300,200 and 100 neurons respectively as well as an output layer of size 1 as the model is used for regression. All the layers use ReLU-activation, the optimizer used is Adam and weights are initialized with He initializer. Early stopping was used to avoid overfitting and to speed up the model.

3.3 Evaluation of the results

3.3.1 Error metrics

Choosing the right error metric is crucial when assessing model performance. As error metrics condense a lot of information into one number, they only highlight one characteristics of the model performance (Chai and Draxler, 2014). Chai and Draxler (2014) continue that this is also why there is no consensus on what error metrics are the optimal, and use varies depending on the problem. This thesis uses mean absolute error, root mean square error and R2 to compare the models. The choice is based on the large extent to which the error metrics in question are used in modeling as well as enabling comparison with previous research. Also, Chai and Draxler (2014) conclude that using several error metrics is often required to assess model performance.

3.3.2 Mean absolute error

Mean absolute error (MAE) is a commonly used metric. It is defined as the absolute difference between the predicted and actual values. For example, if in this paper MAE was 100, it would mean that the model under- or overestimates the actual values by 100 USD on average. The advantages of MAE are its simplicity and easy interpretability, as it can be measured in real life units, unlike for example RMSE. However, this scale dependence means that it can only be used in cases that use the same units, for example money. According to Chai and Draxler (2014), MAE is best suited for problems where the errors are uniformly distributed. From this point of view, as the errors for freemium games LTV: s tends to be very skewed, MAE should also be interpreted with this in mind.

$$MAE = \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

3.3.3 Root mean squared error

Mean squared error (MSE)-based error metrics are widely used in the LTV literature. Several variations are used: (Kurki 2020; Sifa et al. 2018) used RMSE, while (Drachen et al. 2018; Sifa et al. 2018; Chen et al. 2018) measured the accuracy with NRMSE. NRMSE is RMSE where the results is normalized, so it is unit-free unlike RMSE. This makes the error easier to interpret and allows comparison across datasets. Due to the errors being squared, RMSE is more affected by outliers compared to MAE. Chai and Draxter (2014) mention that this property makes RMSE better for model comparison, as MAE may not react to changes adequately.

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

3.3.4 Coefficient of determination

Coefficient of determination, R2, tells how much of the variance of the target variable is explained by the predictors (Nagelkerke, 1991). If R2 was 1, it means that all the variance in target is fully explained by the model features. In this case all the data points would be on the regression line. On the other hand, when R2 is 0, the predictors do not have any prediction power. An important note when dealing with R2 is that it does not imply causality. R2 is calculated by dividing the explained variance of the model with total variance.

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - \bar{y})^2}{\sum_i(y_i - \bar{y})^2}$$

4 RESULTS

4.1 Model performance

Table 3. Results of LTV prediction

Model	Metric	Basic	Seasonal one hot	Seasonal sin/cos
Deep-MLP	MAE	**311.80**	314.21	316.66
LASSO	MAE	**380.40**	412.98	389.85
RF	MAE	**409.70**	413.82	412.47
XGB	MAE	429.25	436.17	**429.00**
Deep-MLP	RMSE	**1284.95**	1295.77	1299.00
LASSO	RMSE	1473.21	**1472.53**	1472.82
RF	RMSE	1521.52	1521.05	**1512.46**
XGB	RMSE	**1554.76**	1555.11	1561.11
Deep-MLP	R2	**0.59**	**0.59**	0.58
LASSO	R2	**0.53**	**0.53**	**0.53**
RF	R2	**0.50**	**0.50**	**0.50**
XGB	R2	**0.47**	**0.47**	**0.47**

In this experiment, deep-MLP stands out considering both MAE and RMSE. This suggests that there is some complexity in the data that simpler models cannot detect, which is in line with findings of (Sifa et al. 2018; Chen et al. 2018; Miettinen 2019; Runge, 2020).

Linear regression performs the second best when considering both MAE and RMSE. This suggests that despite its simplicity, linear regression performs well in LTV predictions. This might be explained by the high feature importance of money spent by the players. For implementation point of view, linear regression is an easy model in a sense that training the model is fast and its simplicity makes is easier to understand also by less technical people.

The seemingly small MAE can be partly explained by the skewedness of the data as large share of cohorts do not generate any revenue. This means that if one for example had 99 zero revenue cohorts and 1 that generates 33 100 USD, if the model predicted all cohorts to be zero spenders, the MAE would be 331. As this example shows, MAE should be interpreted with care and preferably as just a way to measure of magnitude. MAE is not ideal metric when considering for example the biggest spenders, as such extreme datapoints likely have very different absolute errors. From this point of view, thorough analysis on how the error varies across different levels of spend would be beneficial to better understand the model.

RMSE on the other hand punishes extreme values more, so it could be considered as more robust metric out of the two used in this study. When developing such model, one should consider the imbalances in the data, as errors may be very different for high and low spenders. R2 scores show that the features explain at best 59% of the variance in the target, which suggests that the model has decent prediction power. Similarly to other metrics, also the seemingly decent R2 might be due to the skewedness of the data as having a lot of zero or low spenders improves the metric.

4.1.1 Feature importance

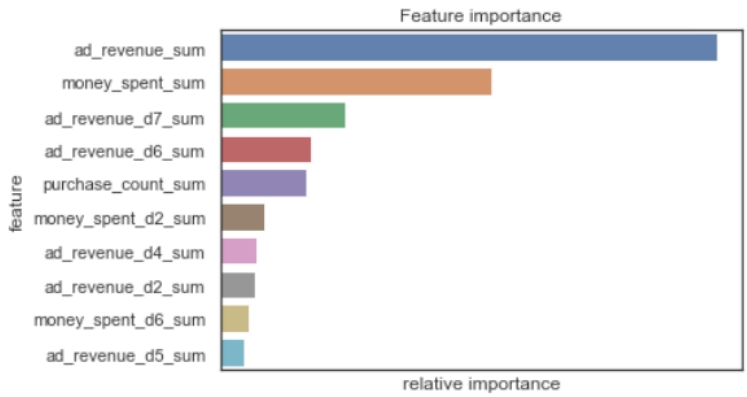

Figure 8. Relative feature importance for a single Random Forest-model

The relative feature importance of Random Forest shows that the revenue-based features play a big role in the predictions. The main difference between ad revenue and in-app purchase is that ad revenue is quite deterministic, as the game maker can affect how many ads players see, whereas purchases are more random and thus harder to predict.

4.1.2 Seasonal features

The results with seasonal features are not as clear as with basic ones. For the best performing MLP, including the seasonal features does not improve either of the error metrics error metrics. Including sine and cosine transformed features improves MAE for XGBoost and RMSE for RF slightly, whereas one hot- encoding improve LASSO: s performance in terms of RMSE. However, the improvements are so minor that drawing conclusions is hard.

It is unexpected that the seasonal features do not have stronger positive effect on the predictions as one would expect that for example in the winter when people tend to be more inside, playing would be more popular and thus also spending. On the other hand, it could be that then in the high spend period, also the spending is higher during the observation period which in turn adjusts the predictions. Also, the fact that majority of revenue comes from a very small share of spenders raises the question whether this group behaves similarly as the rest or if they tend to behave similarly regardless of the season or day.

It is also noteworthy that cultures and geography affect behavior. In this study, only one country is used but if the dataset included different cultural and geographical players, and the results could be different. One example could be that in many middle eastern countries, Friday is a holiday and Sunday workday, which is the opposite of what one has in Finland. Also seasons differ based on where one is. Even within the United States, there are different cultures and environments leading to weaker aggregate trends.

Considering that the model uses features tracking the money spent and advertising spend on daily level, it can already adjust the predictions for weekends based on those features. It is also important to notice that these results do not imply that seasonal patterns do not exist in the game, but rather that the methods used in this paper cannot capture them.

Also, if one would seek to predict how much is spent on a certain day or how, the seasonal features would likely be more important, whereas in this paper one is predicting the total revenue a cohort brings over a long time. This means that two aggregations are made, first when using cohorts instead of single players and second when one looks at d360 revenue, which contains the seasonality changes of a year. From this point of view, when the whole future of the cohort is treated as one bulk, it might be that the seasonal variation cancels out.

It is possible that the impact of seasonal features would be stronger when one does the predictions on player level. In the case of cohort LTV prediction, one can only know the date players started playing, but it does not allow monitoring their daily behavior after that, as one only has aggregate metrics available. In case of having data on player level, one could know the exact times and durations the players are active allowing using also features such as time of the day. Another issue is that predicting the single high spenders seems to be

challenging for a cohort model as majority of features are on aggregate level. If there is an outlier spender in the cohort, the signals are diminished as features are aggregated.

4.2 Challenges of LTV modeling

Another issue is that LTV models are often quite convenient for the user, as they provide seemingly clear answers to complicated and important business questions. This makes it lucrative to trust the model more than one should base on the past observations. However, LTV is also often still the best method available to get any estimates, making the situation challenging. Considering this there are two available options: to use the LTV despite its shortcomings or to implement changes in the way one operates so that one does not rely on predictions. This is also a major challenge, as even though predicting the future is very hard, for example many investment decisions are based on revenue forecasts in lack of better a better benchmark. In many cases however, it would be beneficial to try to simplify the prediction problem, and instead of predicting actual numbers, focus on modeling the uncertainties with Bayes-approach. Having a complicated LTV model could distract managers from thinking of the uncertainties if they are provided with a single number.

4.2.1 Interpretation of the model

One challenge for using an LTV model is that often the users are not data scientists, which imposes the risk that the user understands the model in a different way than the person who developed it. One should make it clear for the user what the errors are and what they mean in real life. When predicting d360 revenues with using only 7 days data, the errors are so large that the predictions should not be taken as ground truth but rather as indicator.

The fundamental problem of not knowing the impact of using LTV also makes it difficult to estimate whether good or bad marketing performance, for instance, is caused by the model or happens despite using it. There could be a case where a passive marketing strategy without using any model or other method to actively manage the process is performing the best.

5 CONCLUSIONS

Based on this study, deep-MLP is the most accurate model when predicting d360 LTV for a freemium game. Predicting LTV for 360 days with 7-day observation data is challenging and high variance in prediction errors should be considered when developing such models and considering how the results are used. Due to the highly skewed revenue distribution and similar behavior of spenders and non-spenders, the model errors should be evaluated with care. As this paper uses cross-validation, the results could be different in production setting, where the test and train setup would be different and model's ability to generalize emphasized.

Most likely the easiest way to improve the accuracy would be to shorten the prediction window. However, one should be cautious when choosing the window, and consider what are the goals and purposes of modeling the results. Other possibility is to increase the granularity and predict LTV for example on country or network level to smoothen out the impact of outlier users and thus reduce the error. Aiming at improving overall performance instead of campaign level seems to be more realistic approach considering the unpredictability of single cohorts.

5.1 Limitations and future research

Considering the difficulty of predicting d360 revenue on seven-day observations, the problem could be made simpler by reframing the problem as a classification one. Setting a threshold, let one say X% ROI on day 360, one could use a model to predict whether the cohort will reach this goal or not. This could potentially produce more accurate model, as the task is simpler than predicting exact revenues.

Another interesting approach would be to use a sequential model, like RNN, to predict the purchases and use that as basis for LTV. This could allow for capturing seasonality better compared to using a machine learning model to predict the aggregate revenue directly. Also using longer observation period could allow the model to learn seasonal patterns better.

References

Chen, C., Liu, Z., Zhou, J., Li, X., Qi, Y., Jiao, Y., Zhong, X., 2020. How Much Can A Retailer Sell? Sales Forecasting on Tmall. ArXiv200211940 Cs Stat.

Chamberlain, B.P., Cardoso, A., Liu, C.B., Pagliari, R., Deisenroth, M.P., 2017. Customer lifetime value prediction using embeddings, in: Proceedings of the 23rd ACM SIGKDD International Conference on Know

Bauer, J., Jannach, D., 2021. Improved Customer Lifetime Value Prediction With Sequence-To-Sequence Learning and Feature-Based Models. ACM Trans. Knowl. Discov. Data 15, 1–37. https://doi.org/10.1145/3441444

Burelli, P., 2019. Predicting Customer Lifetime Value in Free-to-Play Games. Data Analytics Applications in Gaming and Entertainment. Series: Data analytics applications, pp. 79–107. https://doi.org/10.1201/9780429286490-5

Bussière, D.,2016. Understanding of the Day of the Week Effect in Online Consumer Behaviour 8.

Chai, T., Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geosci. Model Dev. 7, 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

Chen, P.P., Guitart, A., del Rio, A.F., Perianez, A., 2018. Customer Lifetime Value in Video Games Using Deep Learning and Parametric Models, in: 2018 IEEE International Conference on Big Data (Big Data). Presented at the 2018 IEEE International Conference on Big Data (Big Data), IEEE, Seattle, WA, USA, pp. 2134–2140. https://doi.org/10.1109/BigData.2018.8622151

Chen, C., Liu, Z., Zhou, J., Li, X., Qi, Y., Jiao, Y., Zhong, X., 2020. How Much Can A Retailer Sell? Sales Forecasting on Tmall. ArXiv200211940 Cs Stat.

de Carvalho Santos, G.V., 2020. Feature importance analysis for User Lifetime Value prediction in games using Machine Learning: an exploratory approach.

Drachen, A., Pastor, M., Liu, A., Fontaine, D.J., Chang, Y., Runge, J., Sifa, R., Klabjan, D., 2018. To be or not to be... social: Incorporating simple social features in mobile game customer lifetime value predictions, in: Proceedings of the Australasian Computer Science Week Multiconference. pp. 1–10.

Fader, P., Hardie, B., Lee, K., 2005. RFM and CLV: Using iso-value curves for customer base analysis. J. Mark. Res. Am. Mark. Assoc. ISSN XLII, 415–430. https://doi.org/10.1509/jmkr.2005.42.4.415

Fader, P.S., Hardie, B.G.S., 2009. Probability Models for Customer-Base Analysis. J. Interact. Mark. 23, 61–69. https://doi.org/10.1016/j.intmar.2008.11.003

Fader, P.S., Hardie, B.G.S., 2009. Probability Models for Customer-Base Analysis. J. Interact. Mark. 23, 61–69. https://doi.org/10.1016/j.intmar.2008.11.003

Guitart, A., Tan, S.H., Río, A.F. del, Chen, P.P., Periáñez, Á., 2019. From non-paying to premium: predicting user conversion in video games with ensemble learning, in: Proceedings of the 14th International Conference on the Foundations of Digital Games. pp. 1–9.

Hoppe, D., Wagner, U., 2007. Customer base analysis: The case for a central variant of the Betageometric/NBD model. Mark. ZFP 29, 75–90.

Hwang, E.H., Nageswaran, L., Cho, S.-H., 2020. Impact of COVID-19 on Omnichannel Retail: Drivers of Online Sales during Pandemic (SSRN Scholarly Paper No. ID 3657827). Social Science Research Network, Rochester, NY. https://doi.org/10.2139/ssrn.3657827

Kooti, F., Lerman, K., Aiello, L.M., Grbovic, M., Djuric, N., Radosavljevic, V., 2016. Portrait of an Online Shopper: Understanding and Predicting Consumer Behavior, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. Presented at the WSDM 2016: Ninth ACM International Conference on Web Search and

Data Mining, ACM, San Francisco California USA, pp. 205–214. https://doi.org/10.1145/2835776.2835831

Kurki, J., 2020. Empirical analysis of prediction models for churn and customer lifetime value in freemium mobile games.

Miettinen, M., 2019. ProGame-Event-Based Machine Learning Approach for in-Game Marketing.

Muthukrishnan, R., Rohini, R., 2016. LASSO: A feature selection technique in predictive modeling for machine learning, in: 2016 IEEE International Conference on Advances in Computer Applications (ICACA). Presented at the 2016 IEEE International Conference on Advances in Computer Applications (ICACA), pp. 18–20. https://doi.org/10.1109/ICACA.2016.7887916

Nagelkerke, N.J., 1991. A note on a general definition of the coefficient of determination. Biometrika 78, 691–692.

Platzer, M., 2008. Stochastic models of noncontractual consumer relationships.

Rathi, T., 2011. Customer Lifetime Value Measurement using Machine Learning Techniques 34.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386.

Runge, J., 2020. Applications of Advanced Analytics to the Promotion of Freemium Goods. https://doi.org/10.18452/21962

Schmittlein, D.C., Morrison, D.G., Colombo, R., 1987. Counting your customers: Who-are they and what will they do next? Manag. Sci. 33, 1–24.

Sifa, R., Hadiji, F., Runge, J., Drachen, A., Kersting, K., Bauckhage, C., 2015. Predicting purchase decisions in mobile free-to-play games, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment.

Sifa, R., Runge, J., Bauckhage, C., Klapper, D., 2018. Customer lifetime value prediction in non-contractual freemium settings: Chasing high-value users using deep neural networks and SMOTE, in: Proceedings of the 51st Hawaii International Conference on System Sciences.

Valdivia, A., 2021. Customer Lifetime Value in Mobile Games: a Note on Stylized Facts and Statistical Challenges.

Weinberg, B.D., Berger, P.D., 2011. Connected customer lifetime value: The impact of social media. J. Direct Data Digit. Mark. Pract. 12, 328–344. https://doi.org/10.1057/dddmp.2011.2

Zhou, S., Montgomery, A., Gordon, G., n.d. Exploring customer Spending Behavior and Payday Effect using Prepaid Cards Transaction Data 17.