Gilberto Vilar de Carvalho Santos
Marcelo de Avila Afonseca

User Lifetime Value prediction using Machine Learning in a Free-to-Play mobile game

Brazil

December 2020

Gilberto Vilar de Carvalho Santos
Marcelo de Avila Afonseca

User Lifetime Value prediction using Machine Learning in a Free-to-Play mobile game

Final work in the field of Machine Learning to obtain the Bachelor's diploma in Mechatronics Engineering at Escola Politécnica da USP

Escola Politécnica da Universidade de São Paulo

Mechatronics Engineering

Bachelors

Supervisor: Fabio Gagliardi Cozman

Co-supervisor: Matheus Nogueira

Brazil

December 2020

Gilberto Vilar de Carvalho Santos
Marcelo de Avila Afonseca
User Lifetime Value prediction using Machine Learning in a Free-to-Play mobile game/ Gilberto Vilar de Carvalho Santos
Marcelo de Avila Afonseca. – Brazil, December 2020-
Supervisor: Fabio Gagliardi Cozman

Final Work – Escola Politécnica da Universidade de São Paulo
Mechatronics Engineering
Bachelors, December 2020.

I. LTV. II. CLV. III. USP. IV. prediction V. Lifetime Value. VI. machine learning.

Gilberto Vilar de Carvalho Santos
Marcelo de Avila Afonseca

User Lifetime Value prediction using Machine Learning in a Free-to-Play mobile game

Final work in the field of Machine Learning to obtain the Bachelor's diploma in Mechatronics Engineering at Escola Politécnica da USP

Fabio Gagliardi Cozman
Orientador

Professor

Professor

Brazil
December 2020

We dedicate this work to our family and friends that supported us through this journey.

Acknowledgements

Gilberto Vilar de Carvalho Santos.

My most heartfelt thanks to all those involved in the creation of this work. Specially to my parents Gilberto Joaquim and Vera Lúcia and my sister Bárbara, whose unyielding support and patience throughout my graduation have been my cornerstone. Next I would like to thank all of my friends and colleagues, who have been by my side through thick and thin over these last six years. A special thanks to my partner in this thesis, Marcelo de Avila Afonseca and all colleagues from both Politecnico di Torino and Mechatronics 2015 class at USP.

I would also like to show my appreciation and respect for the professors and staff of the Escola Politécnica and the University of São Paulo, specially Dr. Prof. Fábio Gagliardi Cozman, our mentor and guide in this enterprise; Professor Rafael Traldi Moura and Arturo Forner Cordero, both my scientific initiation supervisors.

Finally, I would also like to thank Wildlife Studios, the company that agreed to provide part of its huge dataset that allowed this work to be performed. Special thanks to Matheus Nogueira Salgado, Senior Data Scientist and expert on LTV prediction from Wildlife, that supported us through this journey, with meaningful insights and guidance.

Marcelo de Avila Afonseca.

Firstly, I would like to thank my parents, Melina and Marcelo, for all the support during my graduation and for always encouraging my career choices, giving me advice when I most needed. Also, for pushing me to try different paths and to constantly challenge myself. Secondly, I would like to thank Gilberto Vilar de Carvalho Santos, who, besides being a great friend, took the challenge of developing this thesis with me.

Lastly, I also thank Escola Politécnica of the University of São Paulo for opening several doors in my personal and professional life, enabling me to have a critical point of view over the world and to be a problem solver, always aiming for excellence. I am specially thankful for the school enabling me to study abroad, which has always been a dream of mine.

All the hard work was definitely worth it.

“Work hard, have fun, make history.” - Jeff Bezos

Abstract

Na crescente indústria de jogos para celular, o modelo de negócios *Freemium* se tornou bastante popular, permitindo que indivíduos baixem e joguem gratuitamente, tendo a opção de gastar dinheiro em itens virtuais no aplicativo, se desejarem. Portanto, identificar e reter os usuários com altos gastos tornou-se crucial para as empresas de desenvolvimento de jogos maximizarem os lucros, uma vez que uma pequena parcela dos jogadores gera a maior parte das receitas. Nesse contexto, o Lifetime Value (LTV) é a métrica mais usada para identificar esses usuários e direcionar o orçamento de marketing em um cenário de tomada de decisões de negócios.

Nesta tese, é realizada uma análise completa de um conjunto de dados real do jogo para celular *Castle Crush*. O conjunto de dados foi fornecido pela empresa brasileira *Wildlife Studios*. Os resultados são uma análise exploratória de dados seguida por insights acionáveis e uma comparação consolidada de vários algoritmos de Aprendizado de Máquina usados para criar um modelo de previsão de LTV.

O estudo mostra que as características mais importantes para a previsão do LTV são diretamente relacionadas às compras (número de compras e receita líquida por compra), o que está de acordo com a literatura. Features estáticas relacionadas aos usuários têm baixa correlação com LTV de longo prazo, mas foram úteis na elaboração dos modelos de regressão e forneceram percepções interessantes para a empresa. Eventos e ações no jogo, como iniciar ou vencer batalhas, também mostraram baixa correlação com a variável alvo em estudo. Por fim, em relação aos algoritmos aplicados, os modelos Lasso, Ridge e Multilayer Perceptron (MLP) foram os que apresentaram melhor desempenho em quase todas as métricas de erro, com o Lasso se destacando por seu alto desempenho com usuários pagantes e por sua baixa complexidade.

Key-words: Machine Learning, predição de LTV, Feature importance, Customer Lifetime Value.

Abstract

In the growing mobile gaming industry, the *Freemium* business model has become quite popular, enabling customers to download and play games for free, having the option to spend money on in-app virtual items if wanted. Therefore, identifying and retaining the high spending users has become crucial to game developing companies to maximize profits, since a small share of players drives the largest part of revenues. In this context, Customer Lifetime Value (LTV) is the most used metric to identify those users and drive marketing budget in a business decision-making scenario.

In this thesis, a complete analysis of a real dataset from the mobile game *Castle Crush* is performed. The dataset was provided by the Brazilian company *Wildlife Studios*. The results are a exploratory data analysis followed by actionable insights and a consolidated comparison of several regression Machine Learning algorithms used to create a LTV prediction model.

The study shows that the most important features for LTV prediction are derived from purchase-related ones (number of purchases and net revenue per purchase), which is in line with literature. Static users features have low correlation with long term LTV but were useful in the prediction model development and provided interesting insights to the company. In-game events and actions, such as starting or winning battles, also showed low correlation with the target variable. Lastly, regarding the applied algorithms, Lasso, Ridge and Multilayer Perceptron (MLP) models were the best performers in almost all error metrics, with Lasso standing out for its high performance with paying users and for its low complexity.

Key-words: Machine Learning, LTV prediction, Feature importance, Customer Lifetime Value.

List of Figures

List of Tables

List of abbreviations and acronyms

F2P	Free-to-Play
LTV, CLV	Customer Lifetime Value
ULV	User Lifetime Value
UAC	User Acquisition Cost
ML	Machine Learning
RFM	Recency-Frequency-Monetary
RF	Random Forest
LR	Linear Regression
NN	Neural Network
DT	Decision Tress
MLP	Multilayer Perceptron
AUC	Area Under the Curve
NMSE	Normalized Mean-Square Error
RNMSE	Root Normalized Mean-Square Error
RMSLE	Root-Mean-Square Logarithmic Error
NRMSE	Normalized root-mean-square Error
SMAPE	Symmetric Mean Absolute Percent Error
IAP	In-app Purchases
EDA	Exploratory Data Analysis
KDD	Knowledge Discovery in Databases
KPI	Key Performance Index

Contents

Part I - Introduction

1 Introduction

1.1 Motivation

1.1.1 Mobile Gaming Industry

The increased adoption of smartphones and the advancements in mobile networks created the perfect scenario for the diffusion of mobile games. Instead of being an extension of PC or console gaming, Mobile gaming has become a unique type of experience with specific characteristics, such as higher availability and personalization. With the market showing significant growth in the past years, several game developers have entered the market, changing the competition levels and the general structure of the segment.

The initial big change in the mobile gaming industry started in 2006-2007 (FEIJOO et al., 2012), when the first wave of smartphones combined with broadband connection created the ideal scenario. In the end of 2007, Apple's iPhone was released and included several features that enabled users to have a different interaction with games, such as touch-screen, a large display and an application store. Since then, downloading from application stores has become the industry standard for accessing different tools and entertainment. This significant familiarity with smartphones and the digital environment enabled developers to move from traditional platforms and explore the benefits of mobile gaming industry.

As indicated by NewZoo's Global Games Market Report, the gaming industry has surpassed the mark of $100 billion in revenues in 2020, with the Mobile segment contributing to almost 50% of the total. which can be seen in Figure 1. The study indicated that this revenue mark represented an increase of 9,3% when compared to 2019, probably related to significant engagement increase due to COVID-19 restriction measures.

This noticeable increase in the overall gaming market and in the share of Mobile gaming in the industry was also depicted by SuperData, a Nielsen Company. Their 2019 study indicate on one hand the growing share of Mobile devices on free-to-play gaming and on the other hand a decline on total market size for both free-to-pay console and free-to-play PC modalities (see Figure 2).

Besides the rising participation of Mobile devices in the gaming industry's revenues, it is also relevant to understand the main actors that take part in this segment and how verticalization levels are changing in recent years.

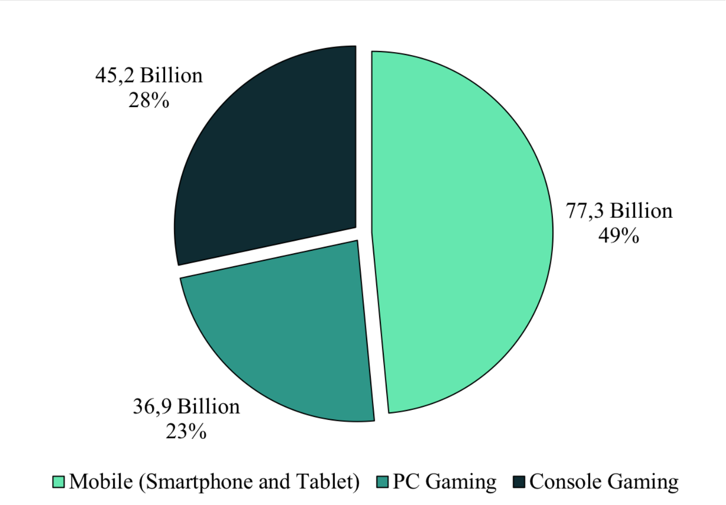


Figure 1 – 2020 Total Gaming Revenue (Billions of US Dollars)
Source: Adapted from NewZoo's 2020 Global Games Report

As shown in Accenture's The Pulse of Gaming 2014 report, the traditional gaming industry value chain was mainly composed by three actors: Developers, Publishers and Distributors. Nonetheless, as indicated in the study, companies have been increasing their capabilities to participate in different stages of the product life-cycle. For instance, Portfolio Management Companies engage in development and publishing activities, being also responsible for brand and investment management. Additionally, the so-called Content Providers are in charge for the publishing and distributing stages, which encompass customer service and vendor relationship managing.

This traditional structure is clearer in PC and Console gaming. However, as the Mobile category advances, the distinctions between industry participants is less noticeable, since companies are more verticalized, integrating both the developer, publisher and distributor roles. The virtual and online environments facilitate the content distribution, considering that applications are mainly commercialized in digital stores on devices.

Likewise, the past years also revealed relevant changes in the business models for games. The industry has four main monetization strategies, summarized in Figure 3. The Physical Distribution was the main strategy used in console and PC gaming in the past decade, corresponding to direct sales of the physical games on retail locations, for example. As digital stores diffusion increased, the Digital Distribution became popular, with examples being payed applications that can be downloaded on mobile devices. In recent years, the third type of business model became quite popular in the general entertainment industry, not only in gaming, with its main examples being Netflix and Spotify.

Finally, the fourth type of monetization is called Free-to-play (F2P), which corresponds to games that can be downloaded, installed and played for free, but players have the option to spend money on In-App Purchases (IAPs) (SIFA et al., 2015). These purchases are virtual items, which can be virtual currency, new characters, decorative features or any item that boost the players' performance and enhance their gaming experience (MARCHAND; HENNIG-THURAU, 2013; SIFA et al., 2015; HANNER; ZARNEKOW, 2015).

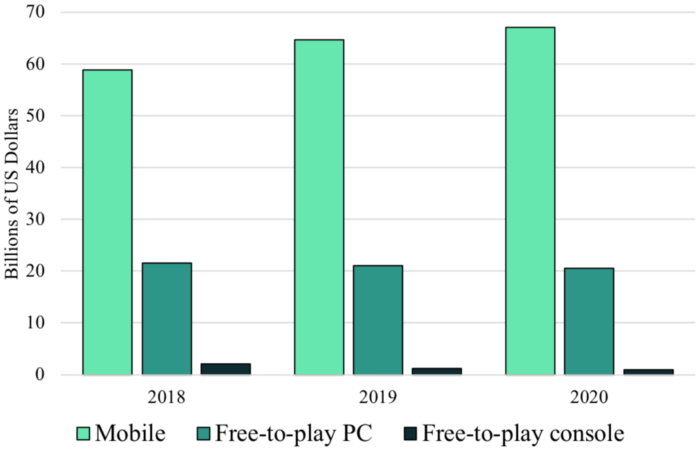
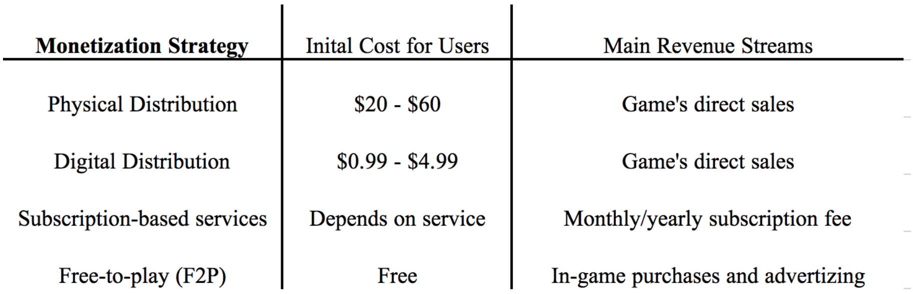


Figure 2 – Free-to-play market and forecast for 2020
Source: Adapted from SuperData, a Nielsen Company, 2019 Year in Review

The models' "free" nature decreases the financial barriers for users, since anyone that has access to internet can download it and start playing (VOIGT; HINZ, 2016). For this reason, F2P games usually attract a huge base of users that grows disproportionately with the revenue generated by them (HANNER; ZARNEKOW, 2015). It means that, although these games attract a lot of people, only a small portion of them are actually converted into paying users (around 5% (GONZÁLEZ-PIÑERO, 2017)). Therefore, a large share of overall revenue derives from a small portion of players.



Monetization Strategy	Inital Cost for Users	Main Revenue Streams
Physical Distribution	\$20 - \$60	Game's direct sales
Digital Distribution	\$0.99 - \$4.99	Game's direct sales
Subscription-based services	Depends on service	Monthly/yearly subscription fee
Free-to-play (F2P)	Free	In-game purchases and advertizing

Figure 3 – Main Monetization Strategies for the gaming industry
Source: Adapted from Accenture, The Pulse of Gaming 2014

Finally, in F2P games, to reach a higher level of profitability, game companies need to attract a huge number of users. It means that marketing expenditures become the main source of costs, since marginal cost per user is negligible in this kind of business (VOIGT; HINZ, 2016). Therefore, this companies must identify the sources of its most valuable customers in order to make a better allocation of its marketing efforts. Customer Lifetime

Value (LTV or CLV) is the main metric used to measure customer value over time, but this is only one of its uses. LTV will be better described in the following section.

1.1.2 Customer Lifetime Value (LTV): definition and importance

As introduced in the past section, Customer Lifetime Value (LTV) is a very important metric used by companies to better evaluate its customers and make strategic decisions, specially in Free-to-play games. This is sustained mainly by the fact that customers In-App expenditures differ a lot, so it is important to identify as soon as possible who are the users that will provide the higher cash flows and those groups who are worth it to invest in marketing campaigns.

Initially, to better understand the motivation and usage of the LTV concept, it is relevant to provide a formal definition. In a business perspective, Life Time Value is defined as the customer's present value in terms of generated revenue and incurred costs for acquiring and servicing the customer. The most general equation comes from the sum of the discounted free cash flows generated by the person over time:

$$LTV = \sum_{t=1}^{T} \left(\frac{Rev_t - C_t}{(1+d)^t} \right) - UAC \tag{1}$$

Where t is the period of cash flow; T is the total number of periods of projected life for the customer under consideration; Rev_t is the revenue from the customer in period t; C_t is the marginal cost of servicing the user to generate the revenue Rev_t in period t; d is the discount rate and UAC is the *User Acquisition Cost* (VOIGT; HINZ, 2016; CANNON; CANNON; SCHWAIGER, 2010).

With that in mind, it is noticeable that customers can expend different amounts while using a certain application and, at the same time, users from different locations and profiles might have different acquisition costs. Moreover, even within paying users there is a high heterogeneity in terms of revenue generation (VOIGT; HINZ, 2016). Thus, it can be concluded that some players are more profitable than others, since their disparate behaviors and different associated costs impact directly the companies' profitability.

Additionally, another aspect that increases the importance of identifying the most valuable costumer is the high competitiveness in the F2P industry in recent years. The low barriers to entry in the gaming industry attracts a huge amount of F2P games to be launched on the market. With more games on market, the chance of retention for a given game is decreased, as well as the cost for publishing and acquiring new players is increased (HANNER; ZARNEKOW, 2015).

After understanding the main idea behind LTV and its relevance in the gaming industry, it is interesting to dig in some examples of usage of that concept (SIFA et al., 2015; MONEREO, 2005; KELLY; MISHRA; JEQUINTO, 2014). The first category regards User Acquisition, which include choosing the right distribution channels to reach the most profitable profiles and allocating the marketing budget campaigns accordingly to the player's predicted return. The second group of use cases are related to User Service and can include evaluating if new in-game features released over time increased overall LTV. Finally, the third category is Pricing Strategy and Promotions, and a few interesting examples are targeting promotions to users with high rates of LTV growth or providing discounts to selected high spending users.

Player Lifetime Value prediction is, therefore, fundamental to help game companies develop profitable marketing strategies, make good business decisions and maximize returns from games. It can be computed through several methods, since historical benchmarks to complex machine learning algorithms (MONEREO, 2005). To do so, game companies need to know how to leverage the massive quantity of data generated from games, mainly from F2P game where the user base is huge, to identify the "best" players' desires and behavior over time.

1.1.3 Company Overview - *Wildlife Studios*

Wildlife Studios is one of the 10 largest mobile gaming companies in the world. Founded in Brazil in 2011, the studio has grown to become a truly global organization. *Wildlife* has offices in the United States, Brazil, Argentina and Ireland, and are still expanding. Today, its portfolio of over 70 games engages billions of players around the world. In December 2019 the company was valued at U$1.3 billion, showing the high volumes of its funding rounds and the potential of this team and business model.

1.1.4 Mobile Game Overview - *Castle Crush*

Castle Crush is an online strategic game available developed by *Wildlife Studios* and based on the free-to-play monetization model. The goal of the player is to destroy the opponent's castle. During the game, each player has a deck of 14 cards, to be chosen between all cards within the player's entire deck. Each card represents a character with a specific attack power, health range and other variable skills. Those cards are to be selected and dragged into three different lanes of the field, where the opponents' characters fight each other. Winning a battle, the player receives prizes in form of trophies, coins (soft currency), gems (hard currency) and chests, that may have a combination of cards, gems and coins. The number of trophies determine the level/arena of the player.

The game revenue stream comes basically from IAPs (in-app purchases). The player can directly acquire gems and, with them, acquire coins, buy cards, open chests and acquire the piggy bank. The latter is a special feature that allows players to accumulate gems and cards for each opened chest (earned in battle or purchased in the store).

As other free-to-play games, the very first days of player interaction with *Castle Crush* are the most important, since is when the churn rate is higher. For this reason, Wildlife needs to identify, as accurate as possible, the most valuable players within 3 days of time span after installation, in order to develop good pricing and in-app promotion strategies.

1.2 Objectives

The main objectives of this thesis are to develop and implement a machine learning algorithm able to predict User Lifetime Value (LTV) after 180 days of use of a Castle Crush's player. The prediction must be done with 7 days data of user-activity after activation date. Initially, a work of data analysis will be done to identify the main characteristics of the users and to identify patterns of usage, determining the most relevant features. Also, this step of data analysis will be useful to better understand the game's data base and the relationships between its main features.

Additionally, some insights obtained after the Exploratory Data Analysis part will be given, in order to provide interesting observations that can be used BY Wildlife in their business.

The algorithm implementation will follow the whole KDD process (Knowledge Discovery in Databases) including: data exploration, preprocessing, feature engineering, data mining, result evaluation and interpretation. In the result evaluation part, the objective is to make considerations regarding the most suitable model to the problem and the given dataset.

2 Literature Review

As stated in the previous section, LTV is a crucial metric when talking about marketing budget allocation and business decision-making. This metric becomes even more important in *freemium* business models, where the customer base is huge and knowing the most valuable group of people is a challenging task.

In this section, an overview of LTV prediction will be presented, including its application in the most variable fields and the specific challenges regarding Free-to-Play games. Finally, state of the art models will be discussed, showing its weaknesses and strengths with practical examples and implementations. With this in mind, this literature review is divided in four categories according to the types of the most common models and algorithms used for LTV prediction: Average Models, Parametric Models, Markov's Chain Models and Machine Learning Models.

2.1 Average models

Average models can be seen as the most basic implementation of LTV prediction. According to Burelli (2019) (BURELLI, 2019), these models are based on the computation of an average customer lifetime value for the entirety of the company's customer base or for a given cohort.

Berger and Nasr (1998) (BERGER; NASR, 1998) are the first researchers that attempted a categorization of average prediction models. The authors give a set of practical examples of applications of LTV in an increasing order of complexity. The most basic model proposed by them assumes that the customer retention rate and the costs of retention are constant over time and both costs and revenues happen periodically at a constant rate. Within these conditions the formula for CLV is the following:

$$CLV = GC * \sum_{i=1}^{n} \frac{r^i}{(1+d)^i} - M * \sum_{i=1}^{n} \frac{r^{i-1}}{(1+d)^{i-0.5}} \tag{2}$$

Where CG is the customer's expected gross contribution margin per period, M is the promotion costs per customer per period, n is the prediction horizon expressed in number of periods, r is the retention between one period and the next, and d is the discount rate. Other examples presented are a variation of this formula, applying probability distributions to the projected profitability and costs per customer, giving a little more robustness to the model.

In the field of free-to-play games, Monero (2005) (MONEREO, 2005) divides this average calculation into two steps, whats is called a *top-down* structure of LTV (see Figure 4). The *Monetization/time* represents the average value generated by a cohort of players (or the entire user base) over a given period of time. The *Lifetime* block is a measure of how much time, on average, people from that cohort are still active. The author shows a set of metrics that can inserted into each of those blocks in order to estimate the LTV of a given user.

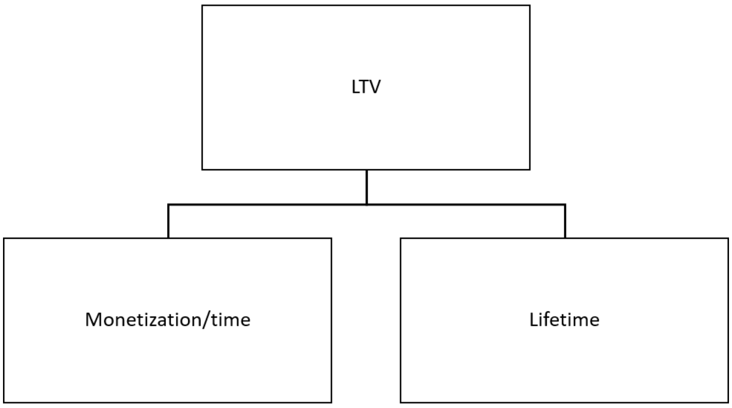


Figure 4 – Average Models diagram

For monetization per time, the most common metrics are ARPU, ARPDAU and

ARPPU. ARPU means Average Revenue Per User and is usually calculated in a monthly basis. ARPDAU is the Average Revenue Per Daily Active User and it filters only users that did not churn yet. Finally, ARPPU means Average Revenue Per Paying User, filtering only the value generated by paying users. Monero also describe simple methods to model the lifetime block, which are all variations on how to model the retention curve of a given group, linear or logarithmic, for example. The multiplication of those two factors gives an estimation of a user LTV based on its cohort past behavior (see Equation 3), and that is why Burelli (2019) and other authors also call this models History-Based Models. The following equation is an example of the multiplication between a monetization metric and a lifetime metric (retention curve, in this case).

$$LTV = \sum_{i=0}^{n} ARPDAU * ret(i) \tag{3}$$

Where $ARPDAU$ stands for average revenue per daily active user, as decribed before, and $ret(i)$ is the value of the chosen retention function at the ith day. It is worth mentioning how simplistic those methods are and, although not extremely accurate, they are easily readable and based on established business KPIs such as APRDAU and retention.

Another common model for LTV prediction is called RFM, which stands for Recency, Frequency and Monetary Value. This type of average model also relies on the previous structure, but dividing the *Monetization/time* block into two separate metrics: Monetary Value and Frequency. Recency is an established metric in direct marketing and it measures how recent was the last interaction of the customer in given point in time and it is conceptually related to the *Lifetime* block. This concept is well explored by Dwyer (1997) (DWYER, 1997) and, along with purchase frequency and average purchase monetary value, has been widely used to predict customer behavior (BURELLI, 2019). Dwyer outlines how even a short period of purchase inactivity might mean the end of the relationship, depending on the business scenario.

Different from the first average models presented in this section, which relies in a very mathematical determination of LTV, the RFM is mostly applied as a clustering model. In general, the objective is to rank the customer base according to their potential LTV. According to Hughes (2000) (HUGHES, 2000) the simplest models classify customers into five groups based on each of these three variables. This gives 5 x 5 x 5 or 125 cells. His studies show that customers' response rates vary the most by their recency, followed by their purchase frequency and monetary value.

Shih and Liu (2003) (SHIH; LIU, 2003) propose a method based on RFM to evaluate CLV. As a first step the method relies on a group of expert evaluation to identify the relative importance of the recency, frequency and monetary variables using analytical hierarchical processing. The customers are than clustered based on the RFM space and the resulting clusters are ranked through a simple weighted sum of the three normalized variables. The results are then validated by simple decision trees and the hierarchical process proves to be a comparable method. Mailing or other marketing communication programs, for example, are prioritized based on the scores of different RFM groups (GUPTA et al., 2006).

As stated before, the Average Models' field contain a set of well established methods for LTV modeling. They are known to be easy to implement and highly readable, which makes them a quick tool to evaluate customer's value over time. However, it is clear that those methods do not show high accuracy, since they rely on average metrics based on past customer behavior. Moreover, this models are only able to process purchasing information, which are not that common in the context of *freemium* business, where only a small percent of customer base is composed by paying users.

2.2 Parametric models

A Parametric (ou Probabilistic) Model in the context of LTV prediction is a model that fits customer past behaviors into probabilistic distributions. In this case, the methods overcome the problem of simplicity attributed to Average Models, since they are not based on average metrics, but on stochastic processes. In a simple manner, they are able to make predictions with a certain confidence about whether an individual will still be an active customer in the future and, if so, what his or her purchasing behavior will be (GUPTA et al., 2006).

2.2.1 Pareto/NBD

Pareto/NBD is a framework originally proposed by Schmittlein, Morrison, and Colombo (1987) (SCHMITTLEIN; MORRISON; COLOMBO, 1987). The Pareto distribution is the combination of an Exponential and a Gamma distribution while NBD stands for Negative Binomial Distribution, which is conceptually the inverse of a Binomial distribution or the combination of a Poisson Process and a gamma distribution. In the Pareto distribution, specifically, the Exponential parameter is gamma distributed across the customer base. Analogously, in the NBD the Poisson parameter is gamma distributed across the customer base (see Equations 4, 5).

$$Pareto \sim Exp(\mu \sim Gamma(s, \beta)) \tag{4}$$

$$NBD \sim Poisson(\lambda \sim Gamma(r, \alpha)) \tag{5}$$

Therefore, the two distributions are controlled by four parameters (r and α for NBD and s and β from Pareto), which are then optimized in order to fit the customer base past behavior.

The authors find that the customers "deaths" follow a Pareto distribution, while the number of purchases made be an active customer follow the NBD, based on six assumptions:

(i) While active, the number of transactions made by a customer in a time period of length t is distributed Poisson with transaction rate λ.

(ii) Heterogeneity in transaction rates across customers follows a gamma distribution with shape parameter r and scale parameter α.

(iii) Each customer has an unobserved “lifetime” of length τ. This point at which the customer becomes inactive is distributed exponential with dropout rate μ.

(iv) Heterogeneity in dropout rates across customers follows a gamma distribution with shape parameter s and scale parameter β.

(v) The transaction rate λ and the dropout rate μ vary independently across customers.

(vi) Monetary Value of transactions follows a Gamma/Gamma distribution with parameters u and w.

Each customer i has its own transactional information $X_i = (x_i, t_i, T)$, where x_i is the number of transactions made until T, t_i is the time instant of the last purchase with $0 < t_i < T$, $[0, T]$ is the window of observation and M_i is the average monetary value per transaction. The final objective is to find optimal parameters s, β, r and α that model the transaction information X_i of the customer base. To do so, Schmittlein suggests two approaches: Maximum Likelihood and Fitting Observed Moments; in case a model without prior data, Schmittlein discusses also the possibility to handpick the parameters based on expert domain judgments. Once the parameters have been identified, each customer's future number of purchases is predicted using the two aforementioned distributions and the customer's x, t and T (see Figure 5).

Glady el al. (2009)(GLADY; BAESENS; CROUX, 2009) propose a modified Pareto/NBD approach for predicting customer lifetime value. Usually, in Pareto/NBD models, the number of transactions and the future profits per transaction are estimated separately. This study proposes an alternative method that shows how the dependence between the number of transactions and their profitability can be used to increase the accuracy of the prediction. The case study is based in a dataset from the retail banking sector.

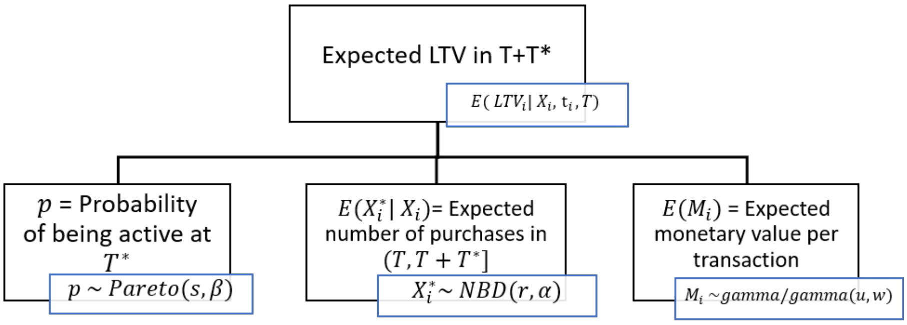

Expected LTV in T+T*
$E(LTV_i| X_i, t_i, T)$
p = Probability of being active at T^*
$p \sim Pareto(s, \beta)$
$E(X_i^*| X_i)$= Expected number of purchases in $(T, T+T^*]$
$X_i^* \sim NBD(r, \alpha)$
$E(M_i)$ = Expected monetary value per transaction
$M_i \sim gamma/gamma(u, w)$

Figure 5 – Pareto/NBD model diagram

Jasek et al. (2019) (JASEK et al., 2019) make a comparative analysis of probabilistic models for LTV prediction in the context of online retail shopping. The authors find that probabilistic models have achieved overall good and consistent results on the majority of the studied transactional datasets, Pareto/NBD models being considered stable with significant lifts from the baseline Status quo model. Moreover, they concluded that Pareto/NBD variants have underperformed multiple criteria and would not be fully useful for the studied datasets without further improvements.

2.2.2 BG/NBD

One of the main issues with the Pareto/NBD is its computational complexity as the estimation requires multiple evaluations of the Gauss Hypergeometric Function. This problem is mitigated by the modified BG/NBD model by Fader et al. (2005) (FADER; HARDIE; LEE, 2005). BG/NBD stands for Beta-Geometric/Negative Binomial Distribution and is an alternative framework derived from Pareto/NBD model. According to Fader, the two models yield very similar results in a wide variety of purchasing environments but it is much easier to implement, leading to the suggestion that the BG/NBD could be viewed as an attractive alternative to the Pareto/NBD in most applications.

The model has the five following assumptions:

(i) While active, the number of transactions made by a customer follows a Poisson process with transaction rate λ. This is equivalent to assuming that the time between transactions is distributed exponentially with transaction rate λ.

(ii) Heterogeneity in λ follows a gamma distribution with parameter r and α.

(iii) After any transaction, a customer becomes inactive with probability p. Therefore the point at which the customer “drops out” is distributed across transactions according to a (shifted) geometric distribution.

(iv) Heterogeneity in p follows a beta distribution with parameters a and b.

(v) The transaction rate λ and the dropout probability p vary independently across customers.

The optimization procedure follows the same steps as the Pareto/NBD (section 2.2.1, but with other four parameters to tune: r, α, a and b. Maximum Likelihood is applied and results comparing Pareto/NBD and BG/NBD are shown. The study shows that BG/NBD is a good alternative for Pareto's model in most business applications.

Jasek et al. (2019), quoted in section 2.2.1, also applies BG/NBD in their comparative analysis. As the Pareto/NBD, the BG/NBD also shows great stability compared to other adapted models. Morover, the author show how easy it is to be implemented and optimized compared to the other models.

As it can be seen in the previous paragraphs, the two models presented in section 2.2 are very known and widely implemented in the field of CLV prediction and in multiple business scenarios. However, these methods have two important limitations in the context of *freemium* business, particularly in Free-to-Play games. The first is that they need a lot of transactional (or monetary) data in order to fit the probabilistic distributions and, as stated in the previous sections, these business models are known to have only a small percentage of paying users. The second limitation is that they are unable to process non-transactional data, which hinders companies ability to leverage the huge amount of data regarding user interaction with the business application (game in-app interactions, for example).

2.3 Machine learning models

This section has the objective to give a basic introduction to the main machine learning models used in LTV prediction problems. Additionally, whenever possible, examples will be given to give a better contextualization of the algorithms and to picture how they can be applied.

2.3.1 Tree-based

Tree-based algorithms are a very popular model used in supervised learning, mostly in classification problems. They can be applied for predicting numerical values (regression trees) or categorical values (categorical trees). There are several variations of these algorithms, which have different advantages and disadvantages, according to the conditions being worked with. The most common types of tree-based algorithms are Decision Trees, Random Forest and Tree Boosting methods.

2.3.1.1 Decision Trees:

Decision Trees are the foundation of all tree-based models and are composed by three main elements: nodes, links and leafs. Nodes represent the features or attributes that define the database being studies. The links or branches indicate a decision being made regarding that attribute. Finally, the leafs, which can be seen as final stage nodes, represent an outcome of the prediction model.

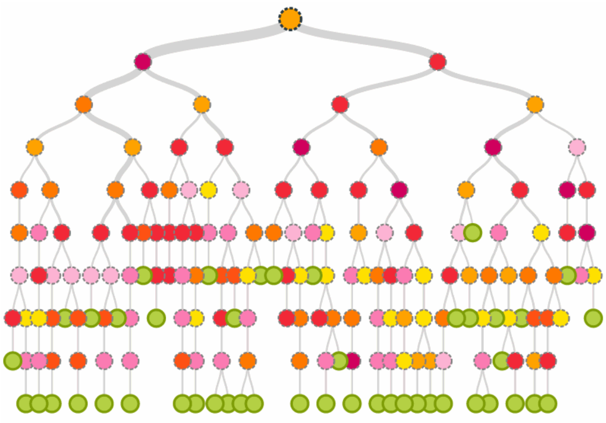


Figure 6 – Decision Tree example

The Decision Tree is traversed from the root node, located at the extreme top, to the leafs, located at the bottom. To categorize an unknown instance, the model starts at the root of the decision tree and follows the branch indicated by the result of each test until a final node (leaf node) is arrived.

As introduced before, Decision Trees can be categorized as Classification Trees (to predict categorical values) or as Regression Trees (to predict real numbers) (AGARWAL; SHARMA, 2011). In the latter, the predicted response is given by the mean response of training observations that are part of the same terminal node. In contrast, for classification, the final prediction is done according with the most commonly occurring class in the node. This flexibility of using Decision Trees to handle discrete and continuous attributes is a big advantage of this type of algorithms.

2.3.1.2 Random Forest:

Random Forest algorithms consist in a large number of decision trees used together, training each of the decision trees independently with a different sample of the observations. There are several benefits related to the use of random forest. While individual trees tend to overfit to the training data, using multiple trees helps to mitigate that problem. Besides that, the model is less affected by perturbations in the dataset, having a more robust characteristic. An example of the general structure of a random forest can be seen in Image 7.

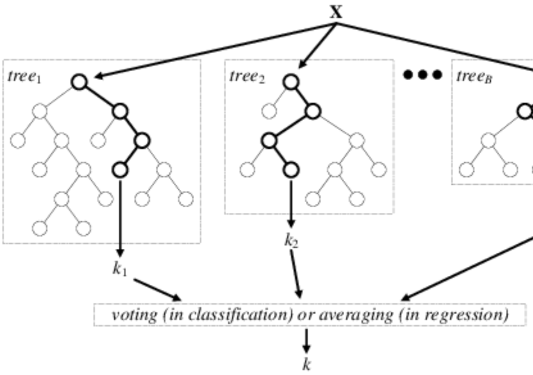


Figure 7 – Random Forest example

Regarding the application of Random Forests (RF) on Costumer Lifetime Value (CLV) prediction, several authors described the usage to be satisfactory (CHAMBERLAIN et al., 2017; DRACHEN et al., 2018). More specifically, Charmerlian et al. (2017) added that by augmenting the RF feature set with unsupervised customer embedding improved CLV predictions when compared to the authors' benchmark.

Furthermore, Drachen et al. (2018) also applied Random Forests in detrmining CLV specifically in mobile games. The authors' study is quite interesting, since they apply RF for both regression and categorical tasks, showing the versatility of this type of algorithms (DRACHEN et al., 2018). In addition, the authors also used other types of classifiers, such as Adaboost and XG-Boost, which will be described in the following sections. The usage of multiple classifiers is quite common in game analytics (DRACHEN et al., 2018). Also, the article states that the best overall classifier used by the authors was the Random Forest one, indicating the high potential of usage in these types of datasets.

2.3.2 Boosting

Boosting is a technique that can be applied in to a wide range of algorithms, such as decision trees and regression. The main goal of boosting, is to improve the performance of other classification algorithms. Boosting has been successful in predicting customer churn in several companies (VAFEIADIS et al., 2015), which has some similarities with our overall problem of customr lifetime value.

For instance, in the tree-based algorithms described before, instead of fitting the dataset with a single decision tree or combining multiple decision trees (Random Forests), boosting models consist of building consecutive small trees, with each node focused on correcting the residuals of the past tree.

This is specially relevant for preventing overfitting problems, which usually occur when applying decision trees in some situations. By applying the algorithm to residuals, we improve the model effectiveness gradually. This also presents another characteristic of the Boosting models, which is the slower rate of learning. In a study elaborated by

Vafeiadis et al. (2015), the authors compared different classification models (regression, decision trees, etc) with its boosted versions. The authors' concluded that for all tested classifiers, there was an improvement in accuracy from 1% to 4%.

(i) Adaboost: Adaboost is one of the most popular boosting algorithms and it is a specific designed for classification problems using decision trees as classifer (VAFEIADIS et al., 2015). It's basic working structure consists of identifying the mis-classified data and increasing their weight in the analysis. This guarantees that the next node will focus more on what generated error in the past tree.

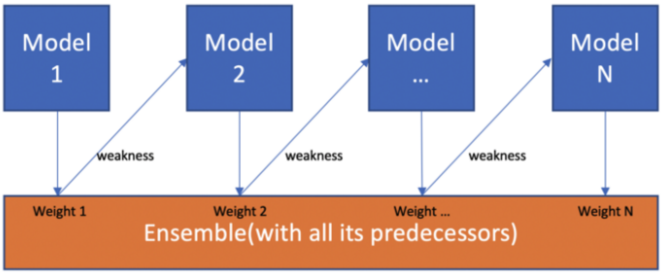


Figure 8 – Adaboost general structure example

2.3.3 Artifical Neural Networks

Artificial Neural Networks (ANNs) are well known machine learning techniques that reproduce the mechanism of learning in biological organisms. The ANNs are composed by neurons, which are the nodes that receive inputs and provide outputs after some processing is done. Nodes can be interconnected, creating a network of several layers that communicate with each other.

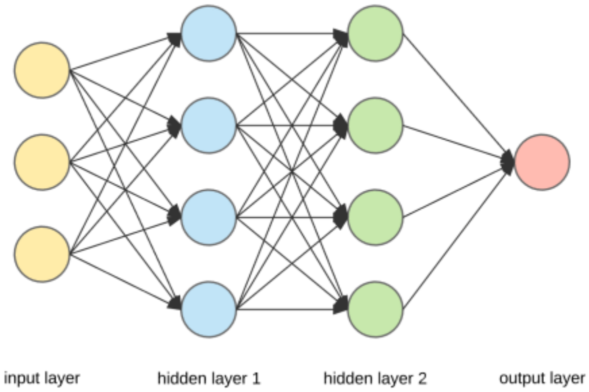


Figure 9 – Neural Network representation with two hidden layers

There are several variations of models that are built using the neural network structure. One example is the Deep Neural Network model, which contains multiple non-linear hidden layers that enables it to deal with very complicated relationships between inputs and outputs (SRIVASTAVA et al., 2014). In some situations, though, these elaborated relationships can lead to overfitting and to long processing time.

In order to deal with that, Srivastava et al. (2014) propose a technique called Dropout, which consists of randomly dropping nodes (neurons) and their respective connections from the neural network during the training process. According to the authors, this prevents units from co-adapting too much, which can give major enhancements over other tunning methods for ANNs.

2.3.4 Multiple models comparison

The methods that are currently dominant within the Free-to-Play games industry are either average based or some form of Pareto/NBD. However, we can see an emerging trend in recent years of more studies being published on the application of different machine learning algorithms to CLV prediction. These methods have a number of advantages over classical statistical methods in this context, as they make no assumptions on the distributions of the input data and easily allow multiple features to be included in the model (BURELLI, 2019).

As presented in the previous sections, machine learning is a wide and growing field of research, providing a large set of algorithms. Those algorithms have their specific statistical appeal and optimization routines, presenting different results depending on input data and objective. This section has the objective to show how machine learning algorithms perform against each other in a LTV prediction scenario.

Tsai et. al. (2013) propose a comparative study of hybrid ML techniques in LTV prediction problems. The input data is from a real stainless pipe manufacturer based in Taiwan. The objective is to compare two types of commonly-used hybrid models that performs the classification task in two stages:

- Type 1: Classification + Classification
- Type 2: Clustering + Classification

Where the classification tasks are performed with Decision tree, Logistic regression and Multilayer-Perceptron, while the clustering stage was done through k-means or Self-Organizing Maps (SOM) algorithms.

I each type of hybrid model, The first stage aims to filter outliers and unrepresentative data (those that cannot be properly classified or clustered) (TSAI et al., 2013). The second performs the final classification, assigning each customer into "valuable" or "not valuable" group. The ground truth is defined with a general score given for each customer based on the weighted RFM (Recency-frequency-monetary) model. The first quintile

(20%) contains the valuable customers ("1") and the rest are the low value customers ("0"). For all models and techniques, the assessment is done through the metrics accuracy, false positive error rate (Type I) and false negative error rate (Type II) with a 10-fold cross-validation test set.

The ML technique for first stage of each model is defined as a single baseline, which means that not all combinations of techniques are evaluated. The second stage (classification) is then implemented with the best baseline technique. Decision tree and k-means performed batter for the baseline of the first and second type of model, respectively. Decision tree also performed better for both models' second stages. The final best hybrid models comparison is shown in Figure 10.

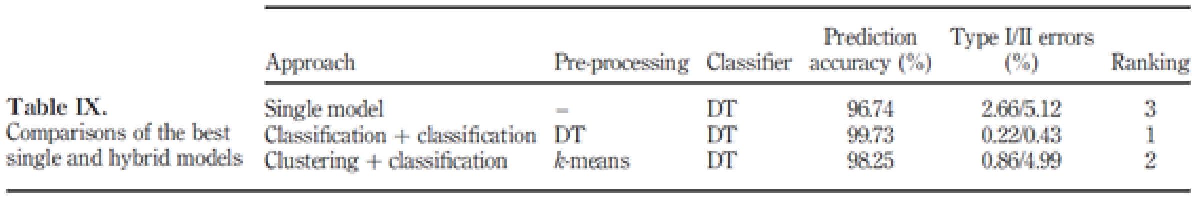


Table IX. Comparisons of the best single and hybrid models

Approach	Pre-processing	Classifier	Prediction accuracy (%)	Type I/II errors (%)	Ranking
Single model	–	DT	96.74	2.66/5.12	3
Classification + classification	DT	DT	99.73	0.22/0.43	1
Clustering + classification	*k*-means	DT	98.25	0.86/4.99	2

Figure 10 – Tsai et. al. best hybrid models comparison
Source: Tsai et. al. (2013) (TSAI et al., 2013)

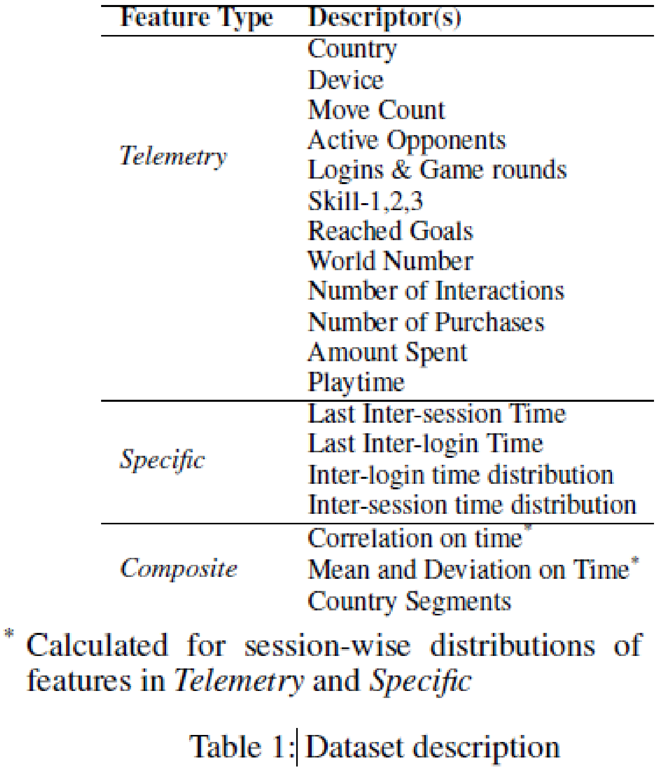
The model Classification+classification was the best performer in all evaluation metrics. In addition, Decision Trees provide a more interpretable model and actionable information about relevant features (although it was not highlighted in the article). The authors also state that this kind of approach and results may vary depending on the real case data in which they are applied. Finally, they completely rely on the RFM model to define customers' value. More techniques like Support Vector Machines, Random Forests and other Neural Networks should be tested besides DT good results.

Sifa et. al. (2015) focus on predicting the purchase behavior of players of a F2P game. The objective is to predict purchasing players using a 2-step approach:

1. Classification task: predict whether a purchase will occur or not.

2. Regression task: predict the number of purchases a user will make.

As described above, the task set follows a simple structure that is somehow related to an Average Model framework, explained in Section 2.1. The first task is a conventional binary classification problem, aiming to classify players into paying and not paying users. The second task aimed to predict future number of purchases for a given paying user, according to the first task classification results. This procedure was applied in a 100000 player dataset using a set of features derived from 30-day of user activity observation. The features are divided into 3 groups: *Telemetry*, *Specific* and *Composite* (see Figure 11).



Feature Type	Descriptor(s)
Telemetry	Country
	Device
	Move Count
	Active Opponents
	Logins & Game rounds
	Skill-1,2,3
	Reached Goals
	World Number
	Number of Interactions
	Number of Purchases
	Amount Spent
	Playtime
Specific	Last Inter-session Time
	Last Inter-login Time
	Inter-login time distribution
	Inter-session time distribution
Composite	Correlation on time*
	Mean and Deviation on Time*
	Country Segments

* Calculated for session-wise distributions of features in *Telemetry* and *Specific*

Table 1: Dataset description

Figure 11 – User level features
Source: Sifa et. al. (2015) (SIFA et al., 2015)

In the classification task, Random Forest performed best on F1-score and G-mean against Support Vector Machine and Decision Tree. SMOTE-NC (Synthetic Minority Over-sampling Technique-Nominal Continuous) was also applied along with each these algorithms in order to deal with the high data imbalance, which is characteristic from F2P games (see Section 1), ans improved accuracy. In the same section, the authors show how the increase in the number of observed days of user activity improved models performance (1, 3 and 7 days, in this case).

In the second task, the authors assume that number of purchases follow a Poisson distribution (similar to Pareto/NBD described in Section 2.2.1) and apply a Poisson Regression Tree (PRT) to predict the target (integral) variable. PRT presents great interpretability as all tree-based models and provides insightful information for business decision-making regarding feature importance (SIFA et al., 2015). Sifa founds that total purchase amount, number of purchases and world number (user in-game level) are the most relevant features for the model (see Figure 12).

Drachen et. al. (2018) presented a similar work in LTV prediction. The study evaluates the influence of specific types of social interactions typical of casual mobile games the on the prediction task. The premium users and Customer Lifetime Value predictions are done by applying classifiers and regression models, respectively, using a 200000 player dataset from a well known puzzle mobile game (DRACHEN et al., 2018). In

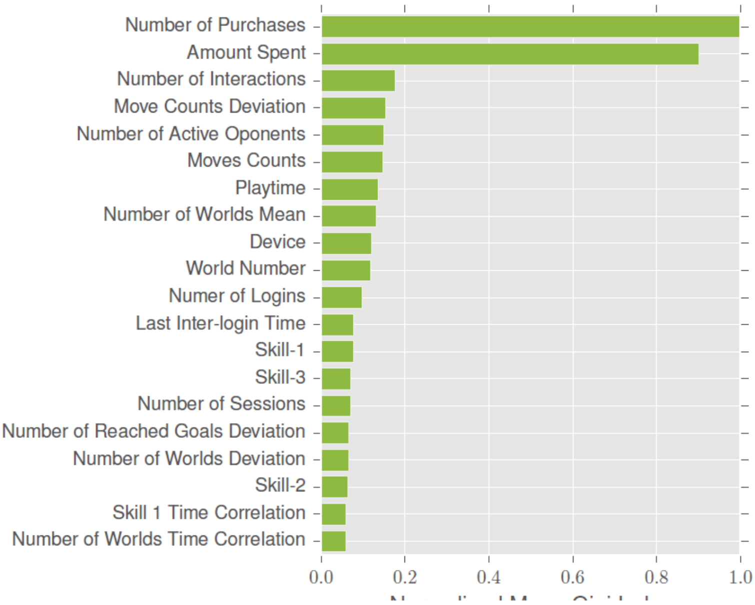


Figure 12 – User level features importance
Source: Sifa et. al. (2015) (SIFA et al., 2015)

the first step, Random Forest and XGBoost perform the best across all periods of user activity observation, according to results of AUC and AUPR evaluation metrics. In the classification task, the authors also apply SMOTE to balance the data. In the regression task, XGBoost outperformed Random Forest on both NRMSE and R^2, although neither of them presented sufficient variance explanation (roughly 9%) (DRACHEN et al., 2018).

Chen et. al. (2018) made a comparative analysis of Deep Learning and Parametric models (see Sections 2.3.3 and 2.2) in LTV prediction within the F2P business scenario. The parametric models applied are Pareto/NBD, BG/NBD and MGB/CNBD-k all combined with gamma distributions, while Multilayer Perceptron (MLP) and the Convolutional Neural Network (CNN) were the deep learning methods implemented. The dataset is from Age of Ishtaria, a role-playing, freemium, social mobile game with several millions of players worldwide, originally developed by Silicon Studio.

The featuring engineering step included only transactional data (RFM) for parametric models, while it also included player behavioral features for deep learning algorithms. Parametric models showed some limitation to deal with huge amounts of data, since they rely only in the RFM model and probabilistic distributions, taking into account only paying users. Deep learning methods outperformed parametric ones on RMSLE, NRMSE, SMAPE and % error, although all of them presented a percent error lower than 10%. Moreover, CNNs presented the best performance over all methods implemented, showing the great potential in predicting LTV (6% percent error). CNNs are also simpler to implement since they can be applied to raw data (time series pictures of the probability density function of each player, see Figure 13) without feature preprocessing as MLP needs (CHEN et al., 2018). The author indicated a set of further studies including other deep learning structures as Recurrent Neural Networks such as LSTM, that implements fully connected layers able to use time-series records as inputs.

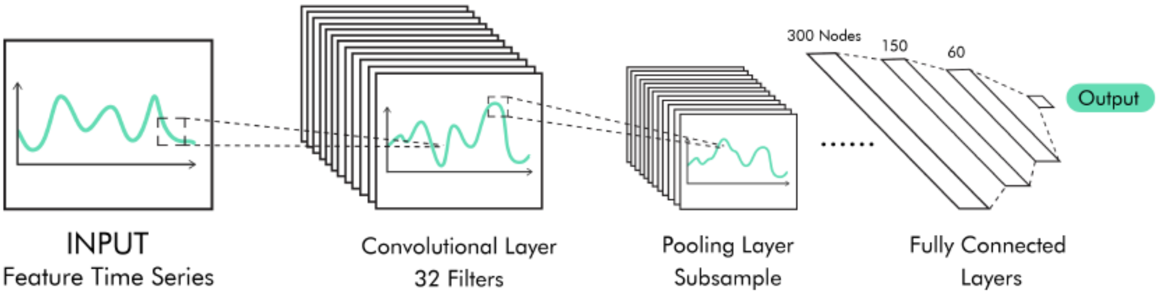


Figure 13 – CNN time series pipeline
Source: Chen et. al. (2018) (CHEN et al., 2018)

Sifa et. al. (2018) came back with another work proposing a direct LTV prediction on an individual player, without the aforementioned step-by-step approach (classification and regression). They also used data from a mobile game operating through the freemium business model (only IAP revenue stream). The objective was to predict LTV of a player 360 days after installation, using only the first 7 days of user-activity data. To do so, a set of machine learning methods are evaluated: Decision Trees, Random Forests, Linear Regression and Multilayer Perceptron, all of them combined with SMOTE to handle data imbalance. The best result on NRMSE was MLP with SMOTE, both on the whole player dataset and among paying users, showing the great potential of deep learning in LTV prediction.

In general, SMOTE seems to improve algorithms performance and the authors also present the most important features using the tree-based models results (see Figure 14). As expected, based on previous articles, Total Purchase Amount and No. of Purchases were the most significant ones, along with behavioral features like Avg. Game Currency, Game Currency Median Round and levels of skill. The authors also analyze “hit hate”, which is how well the sorted LTV is able to capture premium users. The study shows that MLP correctly predicts 70% of premium users and it reflects the good LTV accuracy mainly within premium users. According to Sifa, predicting LTV for zero value users is still a challenge and features like socioeconomic, psychological and personality attributes may improve results of NRMSE for all users.

Kurki (2020) performs and empirical analysis on churn and LTV prediction in a non-contractual mobile game. In the LTV prediction side the author also evaluates a set of machine learning models combined with SMOTE to handle data imbalance. Linear Regressor and Random Forest Regressor surprisingly outperformed Gradient Boosting and MLP on the regression task, based on NRMSE and MAE. However, RF is recommended by the author due to its interpretability and managerial applicability using the feature

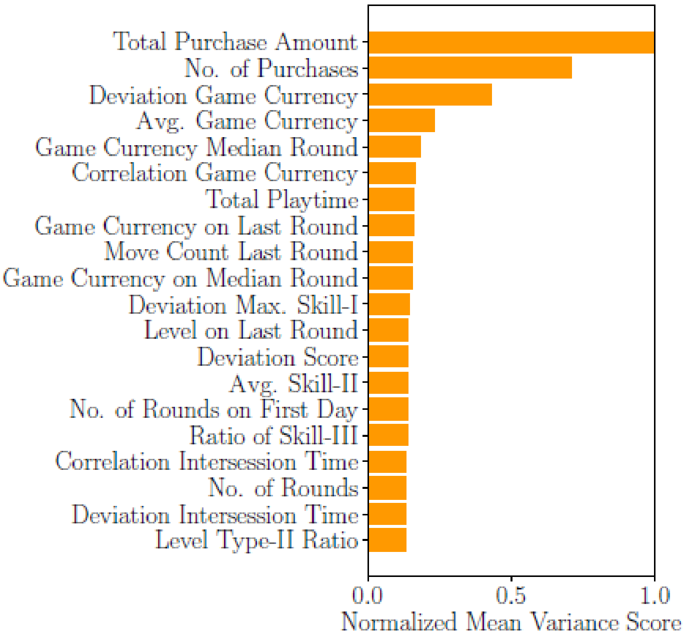


Figure 14 – Features importances

Source: Sifa et. al. (2018) (SIFA et al., 2018)

importance's information. Another key finding is that machine learning models perform significantly better with a 7-day observation period, even when compared with 14 days of observation, which is counterintuitive.

Part II - Methodology

3 Methodology Description

This section aims to introduce the Methodology chapter, showing the steps taken until the final model selection. Each step will be theoretically and broadly explained here, while practically described in detail in the following sections, showing results achieved with the real dataset provided by the company.

The whole methodology of this work is based on the KDD process, which is a commonly used data science framework to extract useful information from large datasets. KDD stands for Knowledge Discovery in Databases (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) and was firstly formalized by Fayyad et. al. (1996) as:

"The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data."

The process contains a set of steps that helps researchers organize their discovery in phases and focus on finding the desired information (see Figure 15). The authors also emphasize that this process in interactive and iterative, being very flexible according to the user's decision-making approach.

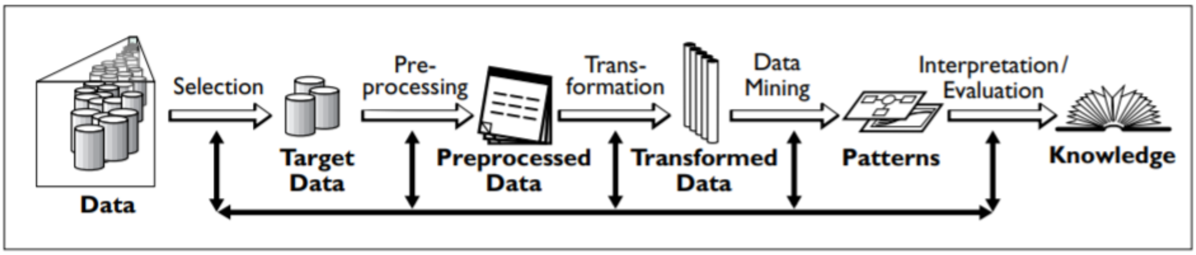


Figure 15 – KDD process
Source: Fayyad et. al. (1996) (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996)

The very first step is Data Selection, which is basically extract and filter raw information from the mother database. Usually, this step involves SQL (Standard Query Language) to query information on cloud and build the raw dataset (Target Data). The objective is to create a Target Dataset, which involves selecting a dataset or focusing on a subset of variables or data samples on which discovery is to be performed. This step may already include some kind of feature selection and feature engineering depending on how tables as organized.

Generally speaking, the second step is Data Preprocessing, which includes a set of basic operations. The most commonly operations include: handling missing data, removing noise or outliers, feature selection and feature engineering. The final goal is to prepare the dataset with the desired information to be interpreted in the Data Mining step. However, in this work, an additional step is included right before and after Preprocessing: Exploratory Data Analysis.

Exploratory Data Analysis (EDA) is the term used to define an approach to analyze datasets in order to extract its main characteristics, often with visual methods. It was promoted by Tukey (1977) to encourage statisticians to explore the data and possibly formulate hypotheses that could lead to new data collection and experiments (TUKEY, 1977). In statistics, according to Behrens (1997), EDA is a well-established tradition that provides conceptual and computational tools for discovering patterns to foster hypothesis development and refinement (BEHRENS, 1997).

In this work, EDA will be used to visualize what raw and processed data can tell about the relationships between input and output variables, in order to generate useful insights about the game, players behavior and LTV. Figure 16 shows a more generalist schema describing a data science process that includes Exploratory Data Analysis.

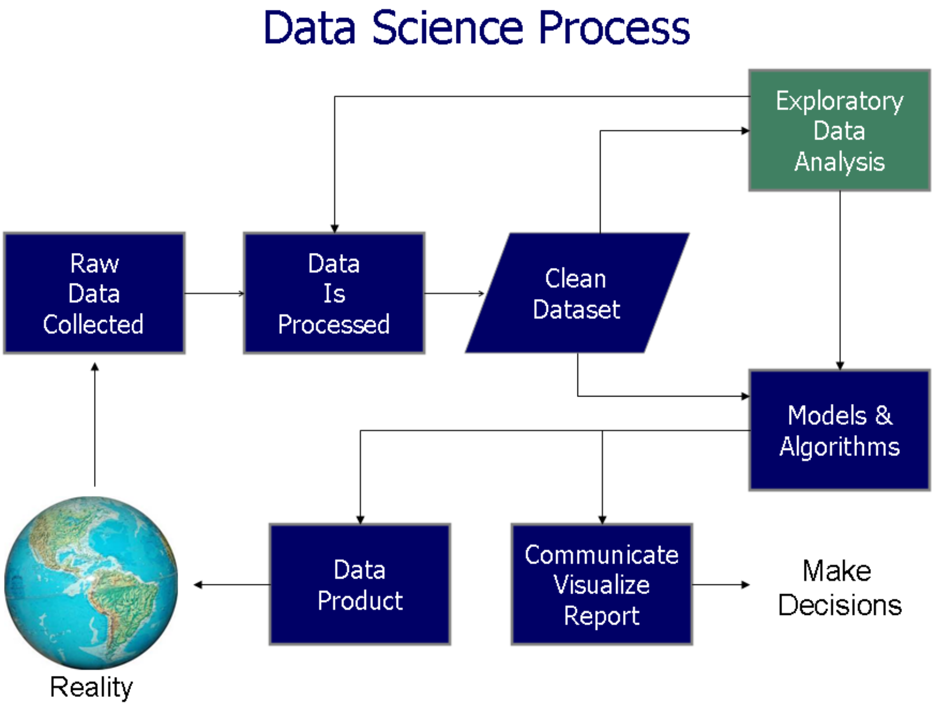


Figure 16 – Data Science Process including EDA
Source: Wikimedia

The next step is Data Transformation, which is the final step before applying the machine learning models. This process aims to handle any data transformation to give the data all characteristics that are necessary to input it to a model. This step is intrinsically interactive, since each algorithm requires a specific input format. Normalization, standardization and conversion of categorical data into numerical data are frequently applied in this step.

The Data Mining step is where intelligent models are developed and implemented in order to identify patterns on the input data. Usually, this process involves the comparison of a set of models that are recurrently optimized according to specific metrics. According to Fayyad et. al. (1996), Data Mining includes searching for patterns of interest in a particular representational form or a set of such representations, including classification rules or trees, regression, clustering, sequence modeling, dependency, and line analysis That is when insightful information is extracted from data (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Finally, the Data Mining results are interpreted and evaluated by the user to generate knowledge. It includes visualize the patterns and possibly returning to previous steps to add or remove relevant or irrelevant variables, respectively. This knowledge is then used in business decision-making or to create a data product (see Figure 16).

4 Data extraction

In this section, the methodology used in the first extraction of data from the company's database will described. The description regards the first interaction with the data via SQL and features previously created according expert domain advice.

4.1 Dataset description

The available data to be worked with consists of three datasets provided by Wildlife and related to the game Castle Crush. The first dataset contains User related features, including the user identification (ID), country, mobile device model, etc. The second, includes Purchase related features, indicating all purchasing activities within the game and revenues generated from those purchases. Finally, the third dataset comprises event related features, which are actions that users can take within the app, such as starting battles, winning or losing battles and engaging with in-game elements.

4.2 Data selection

The biggest challenge in the *Data Extraction* phase was dealing with a huge amount of data, with the used datasets consisting of millions of data points and hundreds of features. To cope we that we decided to focus exclusively on the data points of 2018, which according to Wildlife was the year that Castle Crush had the least number of updates that could affect user behaviour. Additionally, from the three datasets we selected a total of 109 features that we considered to be the most relevant for the LTV problem and that best described and characterized Castle Crush. Then, we joined the chosen features using the user ID number as a primary key.

Below, each selected feature is described and grouped together according to its origin dataset. To simplify, we used the terms *D+x* and *dsx*, where *D* represents the game activation day of each user and *x* the number of days after it. For instance, *D+1* represents one day of the activation day and so on. Additionally, the feature *purchases_ds0* represents the total number of purchases on the game activation day, while *purchases_ds5* shows the total number of purchases 5 days after the activation day, for example.

4.2.1 User information

Table 1 describes the user related features that were selected for the problem.

Feature Name	**Description**
fiu	user's id number
activation_date	user's game activation date
country	user's country
platform	mobile device platform (Android or iOS)
source	app donwload source (organic, facebook installs, etc)
device	mobile device model name (iPad, iPhone, etc)
device_model	mobile device model id
os_version	mobile device's operating system version

Table 1 – Features from the User related database

4.2.2 Purchase information

Table 2 shows the purchase related features that were created from the datasets. For each user, we determined the number of purchases and the net revenue for the first days of playing the game. Also, the table represents the LTV metric being used in this work. This metric was created using the net_revenue features from the Purchase database. The feature LTV_3, for example, represents the sum of the net revenue of each user on the first three days since the game was activated. The same pattern is replicated at LTV_28, LTV_180 and LTV_360. It is interesting to point out that in theory, the Lifetime Value of a customer should also account the costs incurred during the lifetime of the customer. However, to simplify the problem and to deal with the database that was available, only the generated revenues were accounted for.

Feature Name	Description
purchases_dsx	# of user's purchases at D+x days
net_revenue_dsx	net revenue at D+x days
LTV_3	user total spending at day 3
LTV_7	user total spending at day 7
LTV_28	user total spending at day 28
LTV_180	user total spending at day 180
LTV_360	user total spending at day 360

Table 2 – Features from the Purchase related database

4.2.3 Event information

Table 3 shows the event related features chosen to compose our final dataset.

Feature Name	Description
TutorialStart	user starts tutorial
TutorialStartPartx	user reaches part x of the tutorial
TutorialFinish	user finishes tutorial
StartBattle_sum_dsix	# of battles started at D+x
PiggyBankModifiedPoints_sum_dsix	# of times Piggy Bank points were modified at D+x
OpenChest_sum_dsix	# of times user opened the Chest at D+x
StartSession_sum_dsix	# of times user started session at D+x
WinBattle_sum_dsix	# of battles won at D+x
LoseBattle_sum_dsix	# of battles lost at D+x
EnterDeck_sum_dsix	# of times user entered Deck page at D+x
StartGameplayModeBattle_sum_dsix	# of times user started alternative battle mode at D+x
UpgradeCard_sum_dsix	# of times user upgraded a card at D+x
EnterShop_sum_dsix	# of times user entered the game shop at D+x
BuyCard_sum_dsix	# of times user bought a card at D+x
StartTournamentBattle_sum_dsix	# of times user started a tournament battle at D+x
WinTournamentBattle_sum_dsix	# of times user won a tournament battle at D+x
LoseTournamentBattle_sum_dsix	# of times user lost a tournament battle at D+x
ChangeArena_sum_dsix	# of times user advanced on Arena level at D+x
JoinTournament_sum_dsix	# of times user joined tournament at D+x
QuitTournament_sum_dsix	# of times user quit tournament at D+x
OpenPiggyBank_sum_dsix	# of times user opened PiggyBank feature at D+x

Table 3 – Features from the Event related database

4.2.4 Raw Master Dataset

After selecting all the features chosen to be worked with, a *Raw Master Dataset* was created. In total, 111 features were selected and the primary key was the user identification number, showed as *fiu* in the raw databases. This *Master Dataset* was the one used in the Exploratory Analysis described in the following section.

Part III - Analysis

5 Exploratory Data Analysis

Following the description made in the previous section, the features extracted from the company's database were divided in 3 categories:

- User-related features
- Purchase-related features
- Event-related features

Each feature of each category will be carefully analyzed in two steps. First, the statistical distribution and relevance (if any) of the feature in the dataset will be plotted. Second, the feature's relationship with the target variables (LTV variables) will be extracted and analyzed. Each feature's section will be closed with a detailed conclusion regarding the corresponding attribute, in order to generate actionable insights to the company.

Additionally, due to privacy matters, several charts that contained sensible information to Wildlife were modified to normalized values or had the representative axes omitted.

5.1 User-related features

5.1.1 Country

Figure 17 shows the user distribution across the Top 20 counties in the selected dataset. India is the country with most users, having almost twice the quantity of the second country, Brazil. The letters WW mean Worldwide, representing the users that were acquired to the platform through global marketing campaigns. The chart also shows that a significant portion of the user base is concentrated on a few countries. This can be specially insightful for Wildlife to better focus their efforts on acquiring new users and expanding their customer base on parts of the world that have not been representative so far.

According to Figure 18, Iceland, Hong Kong and Thailand are the Top 3 paying countries on average. However, when we compare Figure 17 with Figure 18, we can see that those Top 3 countries don't have a significant representation on the number of users. Also, the United States is ranked number 13 in average LTV180 and also has a significant share of the user base, indicating the potential importance of a user being from the U.S.

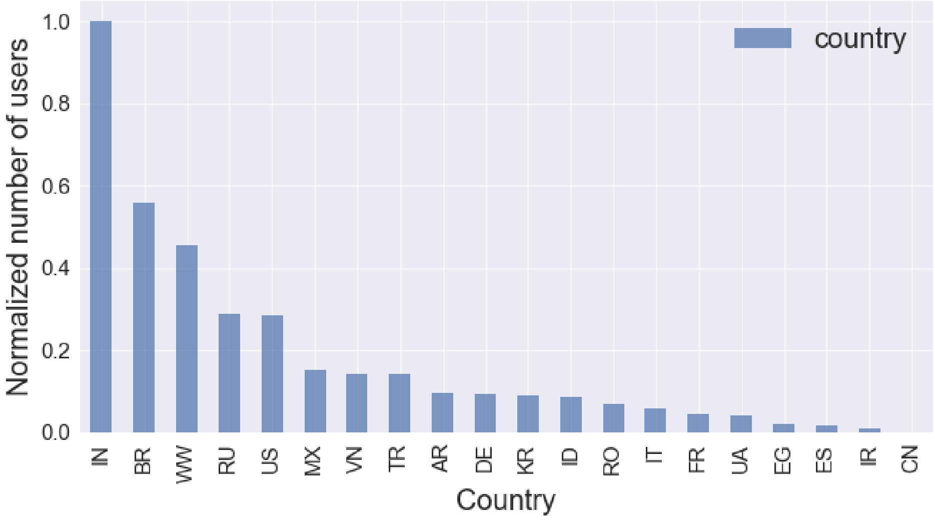


Figure 17 – Top 20 country distribution

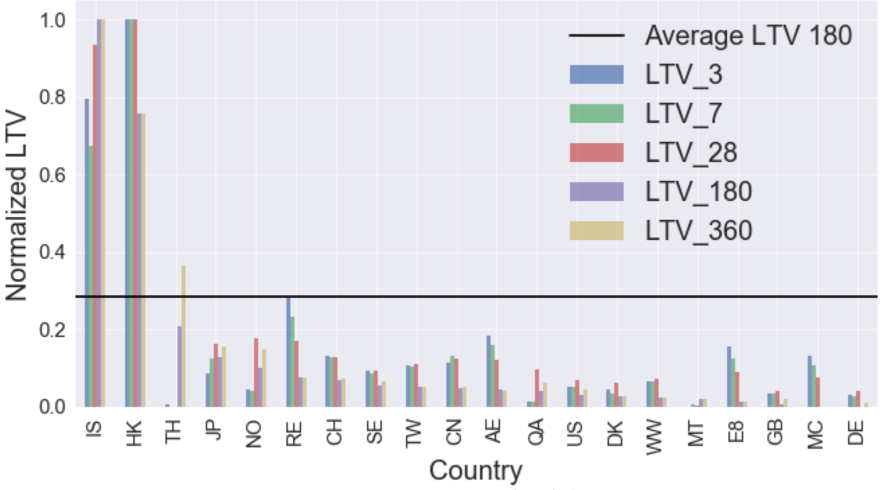


Figure 18 – Top 20 countries ranked by average LTV180

Although it is interesting to check the highest spenders, we see that being a top spender on average does not mean having a high and positive correlation with LTV value. The correlation heat map (Figure 19) shows that all of these countries have an extremely low correlation with LTV. It proves that average metrics involving a huge and diversified dataset like this are not that useful for LTV prediction.

Figure 20 shows the 10 countries with the highest correlation with LTV180. Thailand is the most correlated country, followed by Hong Kong and the denomination WW (Worldwide). Also, the United States and India appear in the fourth and fifth positions and both have a significant proportion of the user base (see Figure 17), which means that they are frequent in the database and useful for LTV prediction at the same time. However, the correlation itself still seems to be very low compared to other features analyzed in the literature, when taking into account the country features independently.

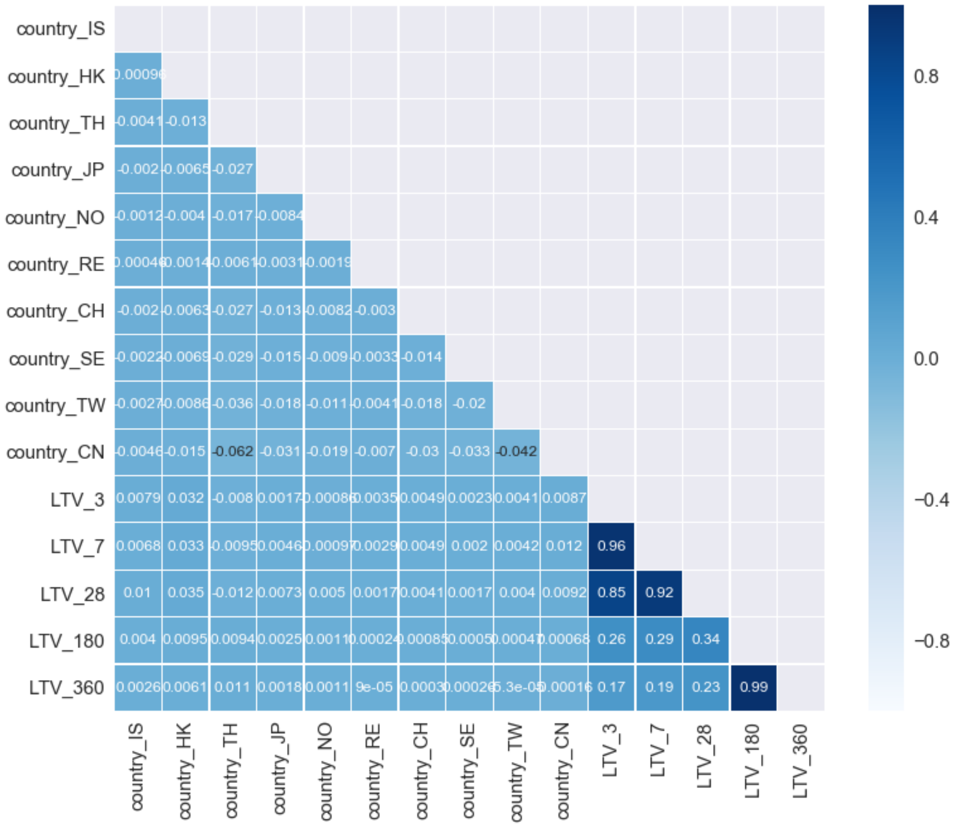


Figure 19 – Correlation heatmap for top 10 average LTV 180 countries

5.1.2 Platform

Figure 21 shows that 88% are Android users and only 12% uses iOS. It means that the user base in very imbalanced regarding the platform feature, although in line the expected superiority of Android in number of owners, since Apple cellphones are known to be more expensive.

Figure 22 contains the average LTV variables by platform and shows that iOS players have significantly higher value, with an average LTV_3 already higher than the user base average LTV_180, which indicates that those users are more valuable to Wildlife. Furthermore, the steepness of the curve connecting the top of each LTV bar is much higher for iOS players than for Android's, which means that Apple users spending activity is much more intense over time.However, as we saw in Figure 21, we have only 12% of users belonging to the high spending platform group. This means again that the regressor will have only a few examples of high spenders regarding platform segmentation.

The heatmap plotted in Figure 23 shows that the platform feature has a higher correlation with LTV when compared with the country feature, but still generally low to perform a regression task.

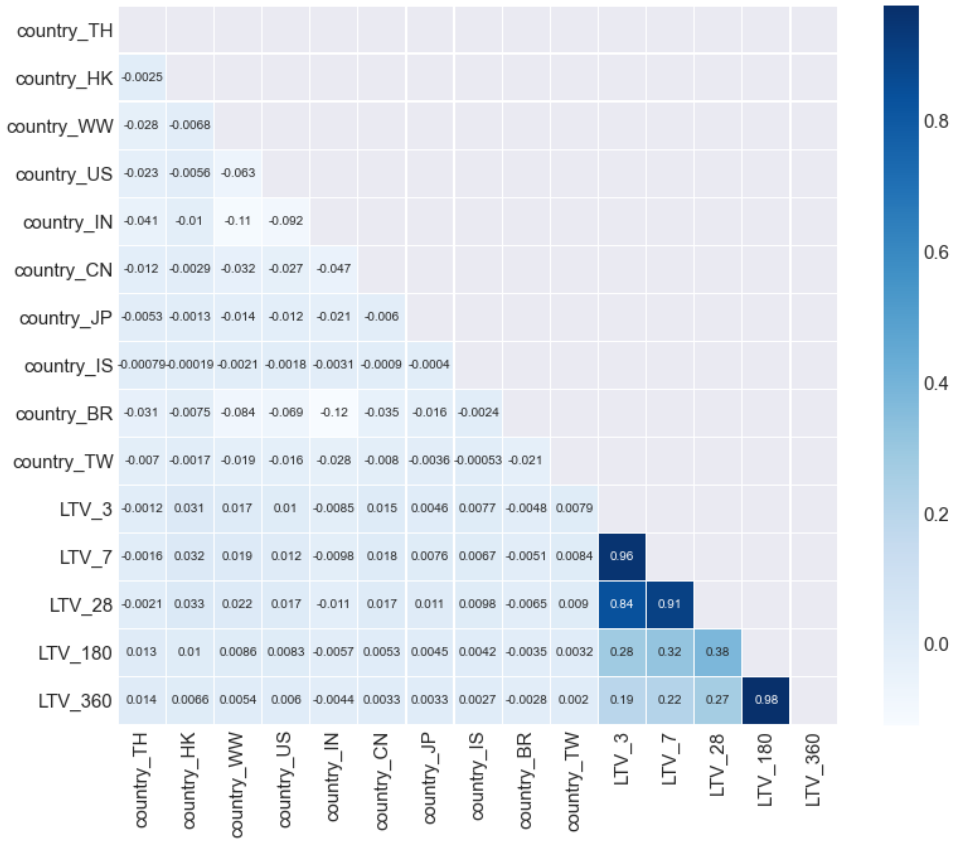


Figure 20 – Correlation heatmap for top 10 average LTV 180 countries

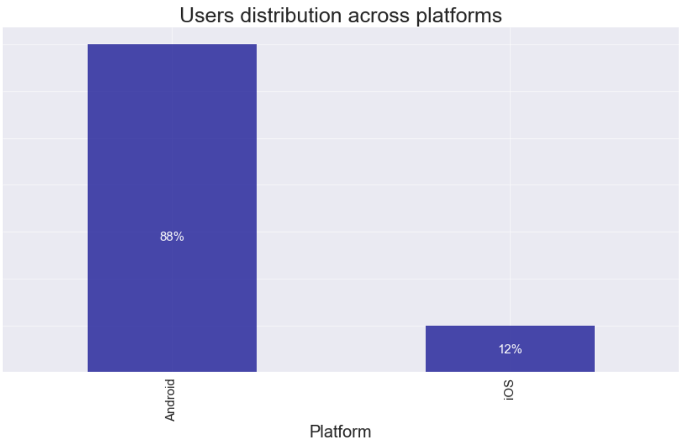


Figure 21 – Platform distribution across user base

5.1.3 Source

Figures 24 and 25 contain the source distribution across users and the top10 average LTVs across sources. We see that the majority of users start playing organically, which means that they usually do not download the game from paid sources. In Figure 25, due to privacy concerns, the sources names' were omitted, however, we can notice that Sources 1 and 2 clearly have a significant dominance when it comes to LTV. This means that Widlife can increase its focus when it comes to the players with the highest LTV and narrow it

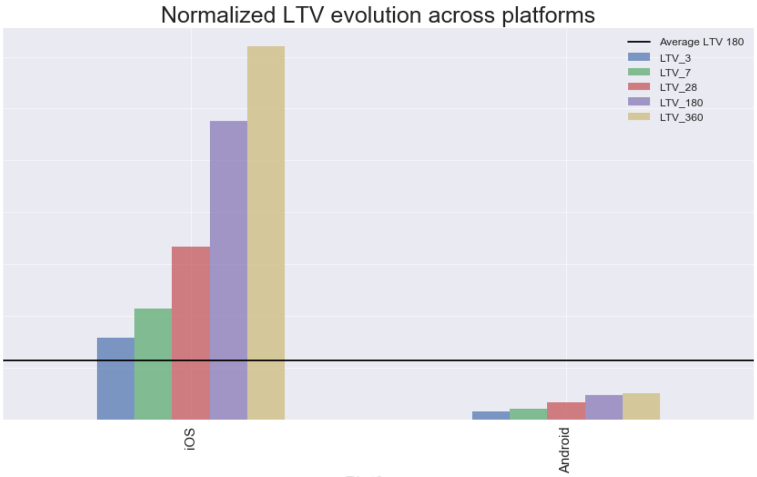


Figure 22 – Platform's average LTV

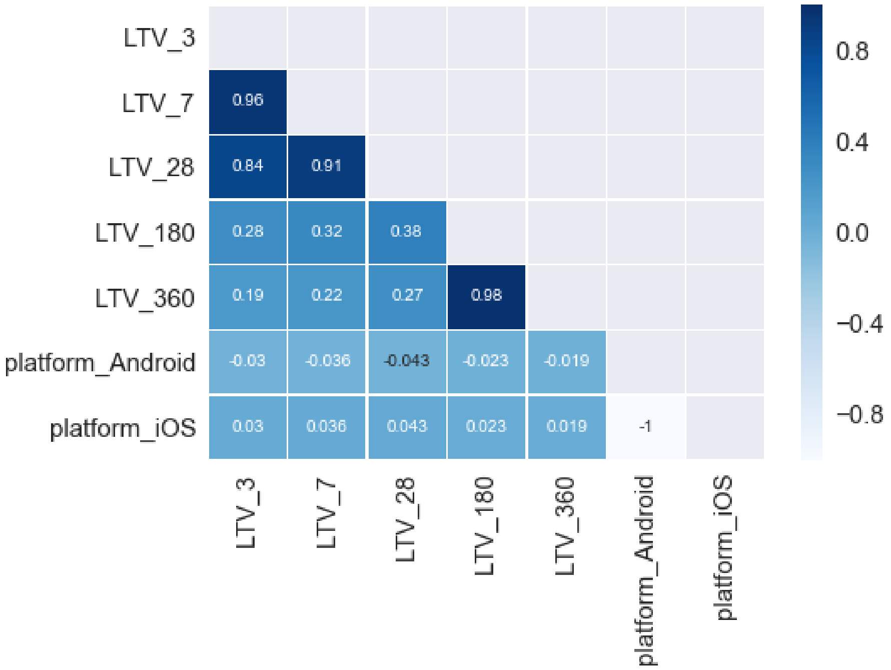


Figure 23 – Platforms correlation with LTV variables

to the first two sources. Also, Sources 1 and 2 show a clear evolution of LTV over time, being an important behaviour to be observed.

The correlation map for the 27 sources in Figure 26 does not show a clear correlation between them and the LTV variables. However, we can see that the organic source is, in general, negatively correlated with other sources.

We see that the organic source is really negatively correlated with the target variables, which is coherent with the previous results (the majority of users, which come from organic sources, are not paying users). Differently from country feature, the sources present in the highest average LTVs in Figure 25 are also within the most correlated

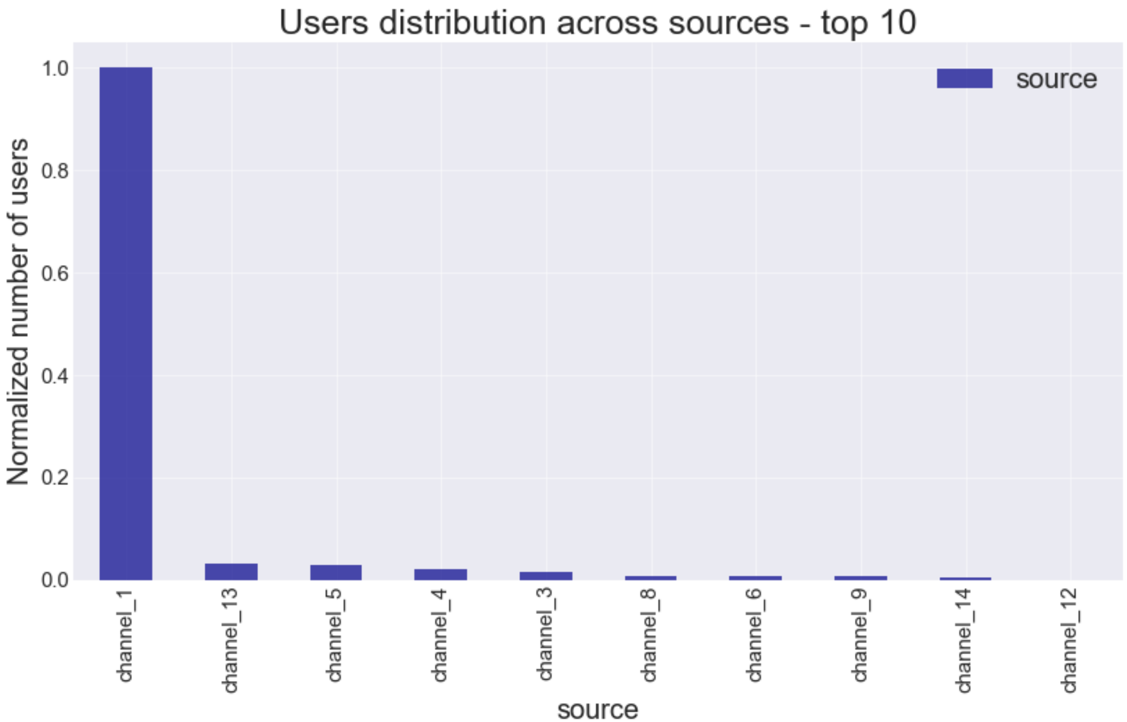


Figure 24 – Top 10 Sources in statistical significance

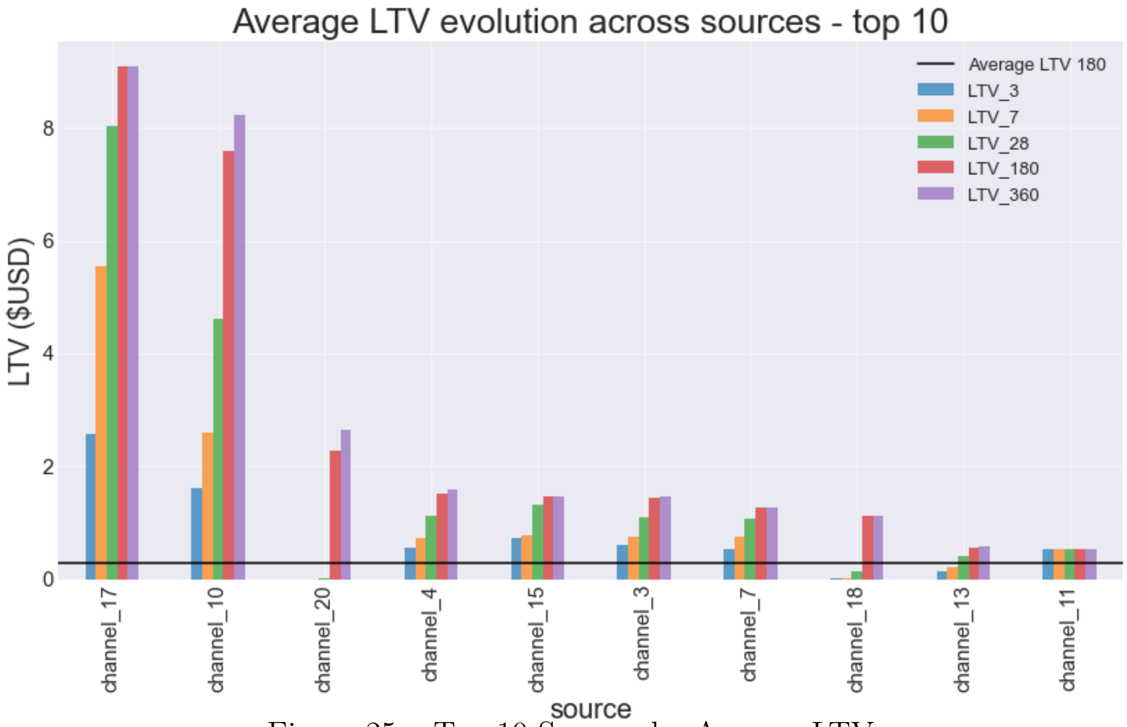


Figure 25 – Top 10 Sources by Average LTV

features. It means that, in this case, the average values of LTV regarding the source feature is meaningful to the LTV prediction task. Precisely, channel_10, channel_4 and channel_3 have the highest correlation with users LTV.

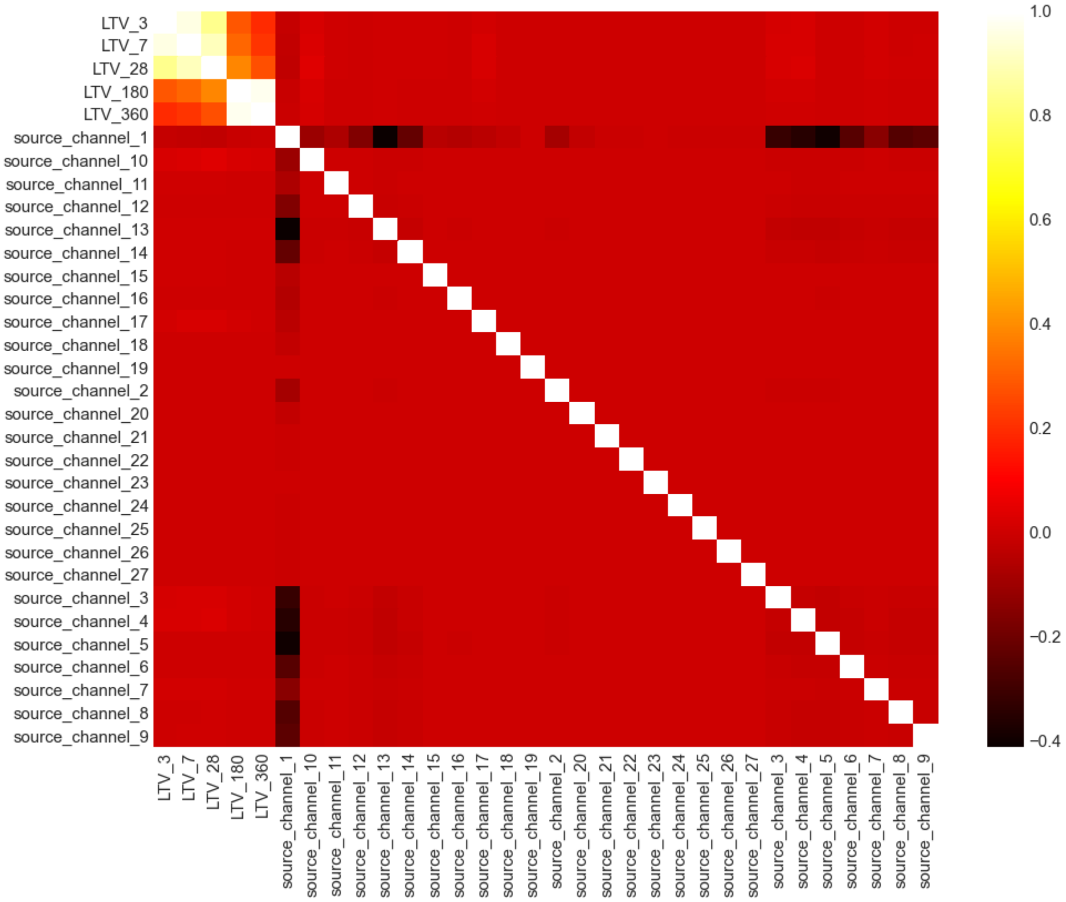


Figure 26 – All sources correlation with LTV variables

5.1.4 Device

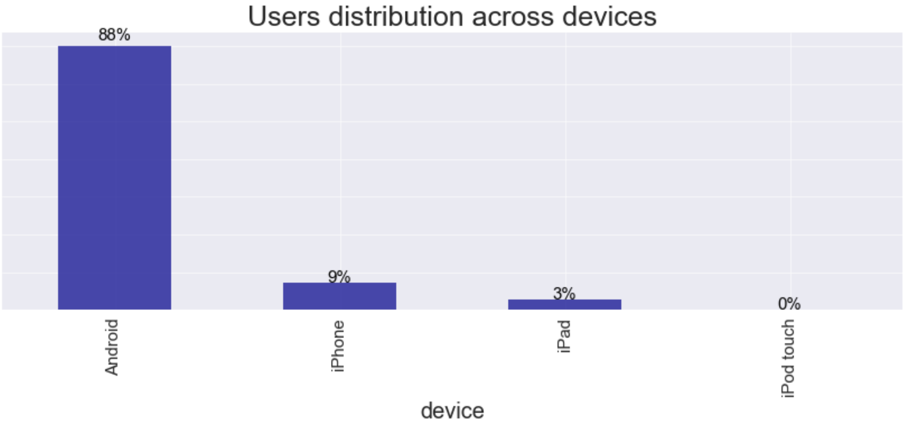


Figure 27 – Device feature distribution

Figures 27 and 28 show the distribution of the 5 device types across the user base. We see that the great majority of players use Android, which is coherent, since Apple devices are usually considered premium. Furthermore, this is in line with the results of platform analysis (see Section 5.1.2). The average LTV values per device shows that Apple

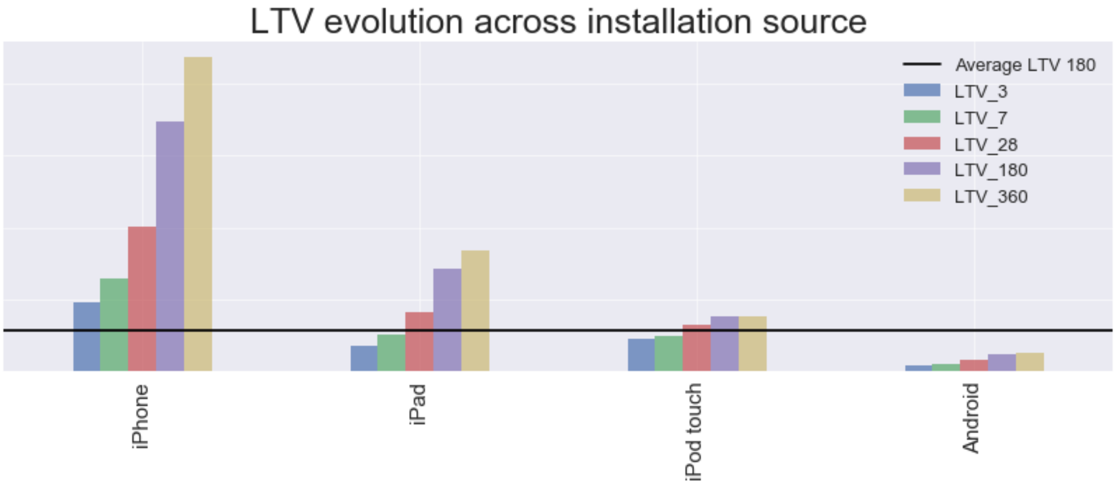


Figure 28 – Device LTV distribution

users are usually more valuable, which is also coherent with the platform analysis, that showed that iOS users are better spenders than Android's. The LTV evolution in time shows consistency across devices.

The correlation analysis between platform and device with the LTV variables is plotted through the heatmap in Figure 29. As intuitively pointed out, the correlation analysis shows high relation between platform and device attributes, which is an indicative that one of those features can be dropped in the preprocessing step. In this case, the platform information seems to have better correlations than device and probably better translate this information into LTV.

5.2 Purchase-related features

5.2.1 Number of purchases

Purchase distributions are plotted in Figure 30 for each single day of user observation. The number of purchases is higher in the first few days of use, which means that users seem to engage rapidly with the game (that is the opportunity to keep the player in the long run). Frequency of purchases starts to decrease over time until the seventh day observed. The exponential decay observed in all purchase charts is in line with the assumptions of the Pareto/NBD model discussed in Section 2.

The correlation map in Figure 31 graphically shows the correlation between each purchase feature. Clearly, there are interesting correlations between LTVs and purchase variables. Visually, the number of purchases made in the first few day seems to be more related to the target variables. Furthermore, as the number of purchases increases across users, for each feature, the LTV decreases. It means that users that make more purchases tend to be less valuable in the long run and this negative relation increases over time.

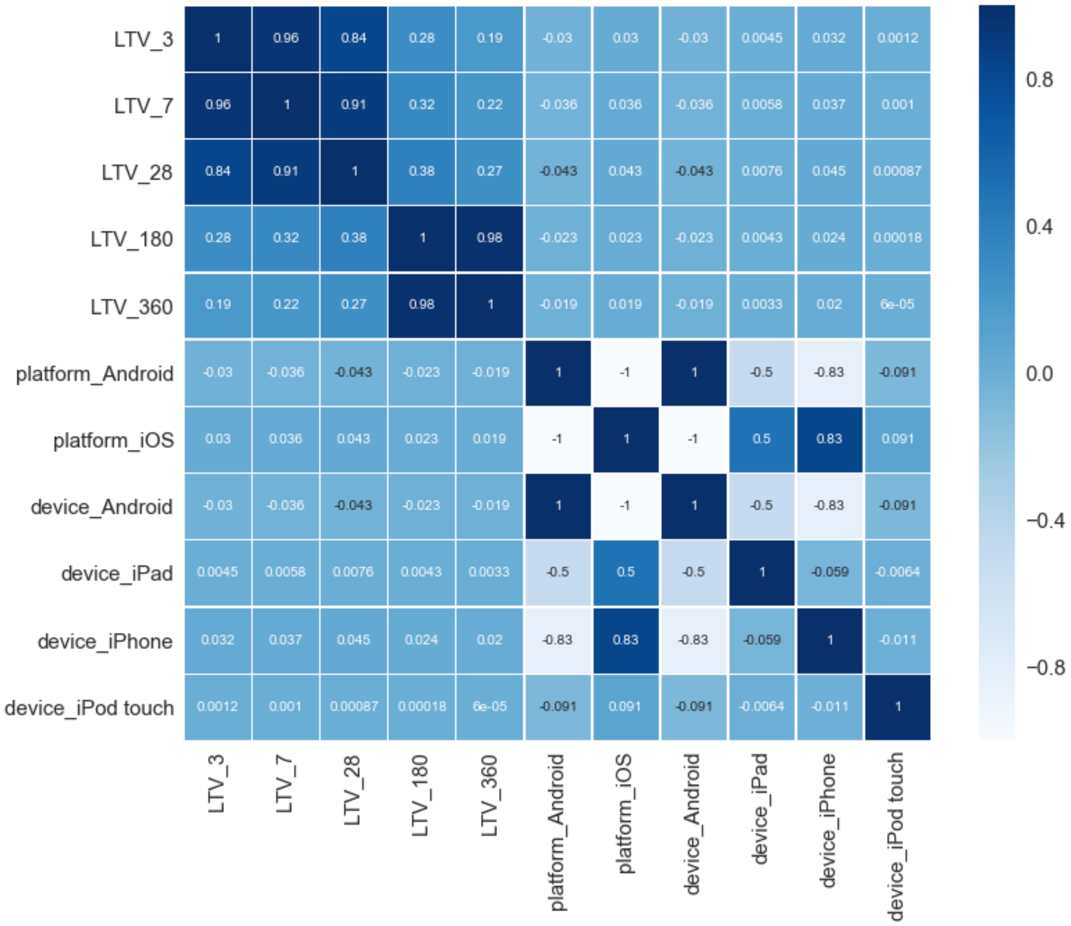


Figure 29 – Device feature correlation with LTV variables

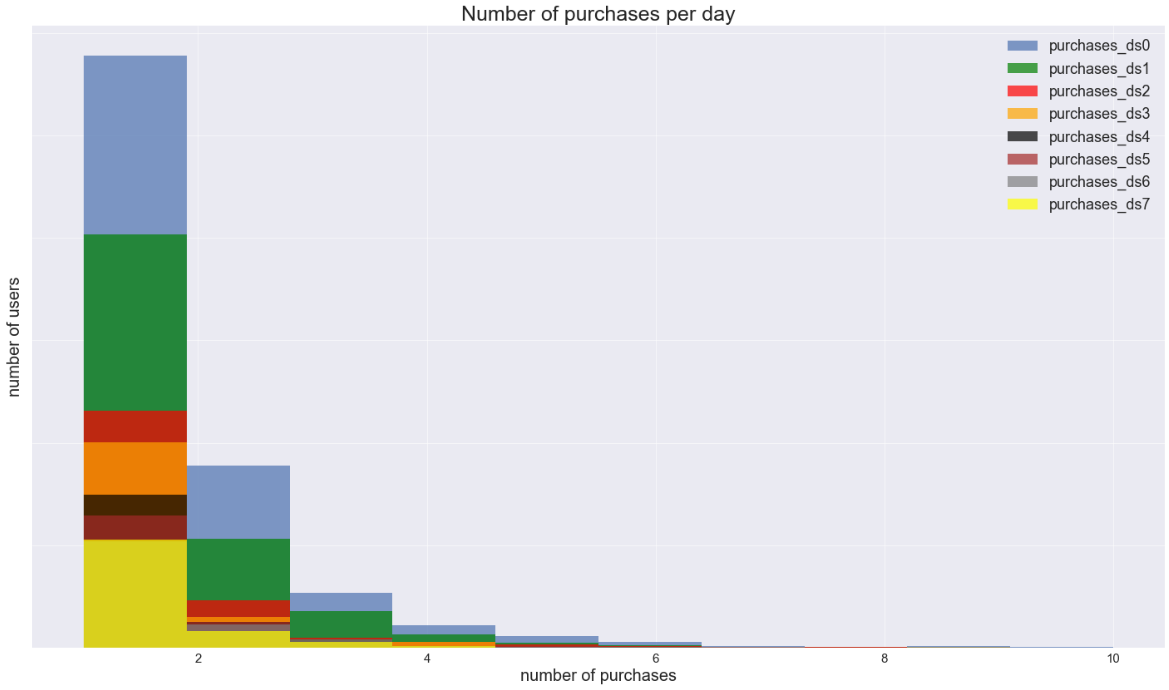
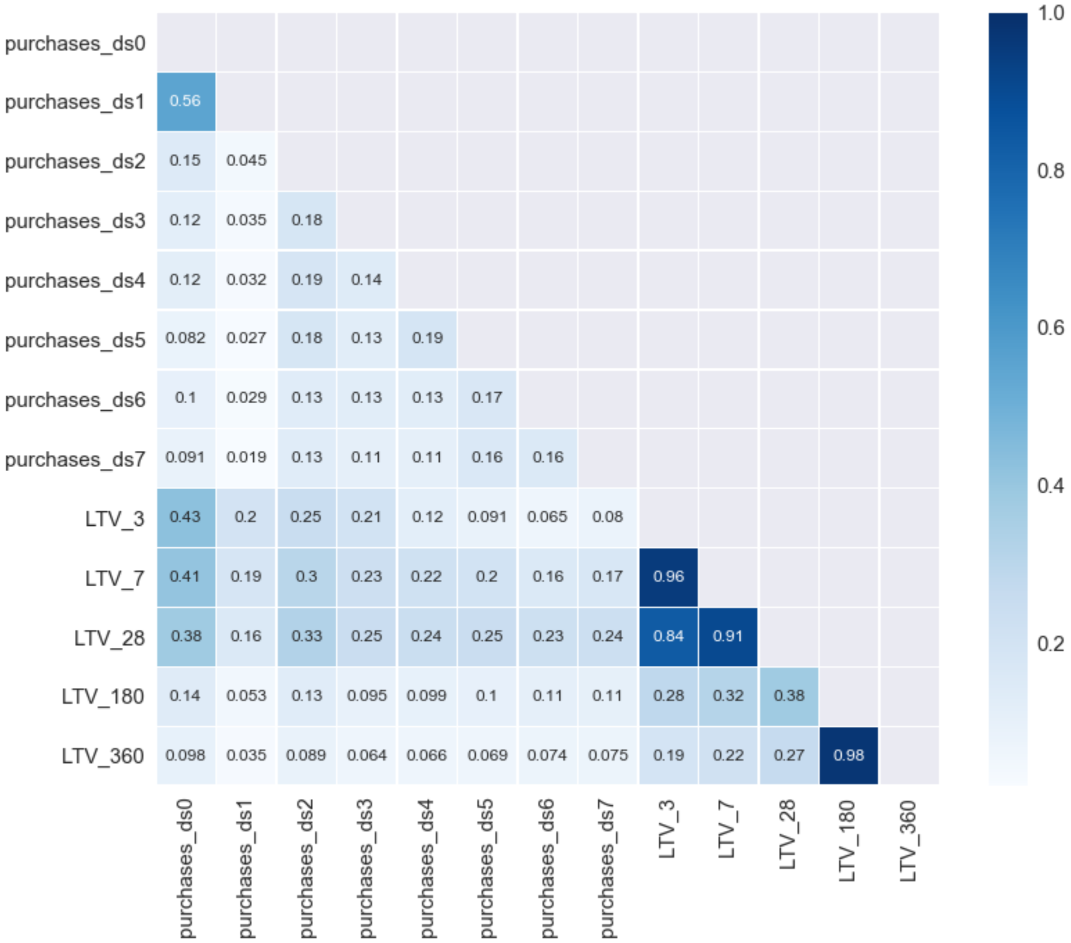


Figure 30 – Purchases histograms until 7th day of user activity

We see that correlation decreases as the purchasing day goes far from the installation day. It means that a person that makes a purchase right after the download day has more chances to become a valuable player in the future.



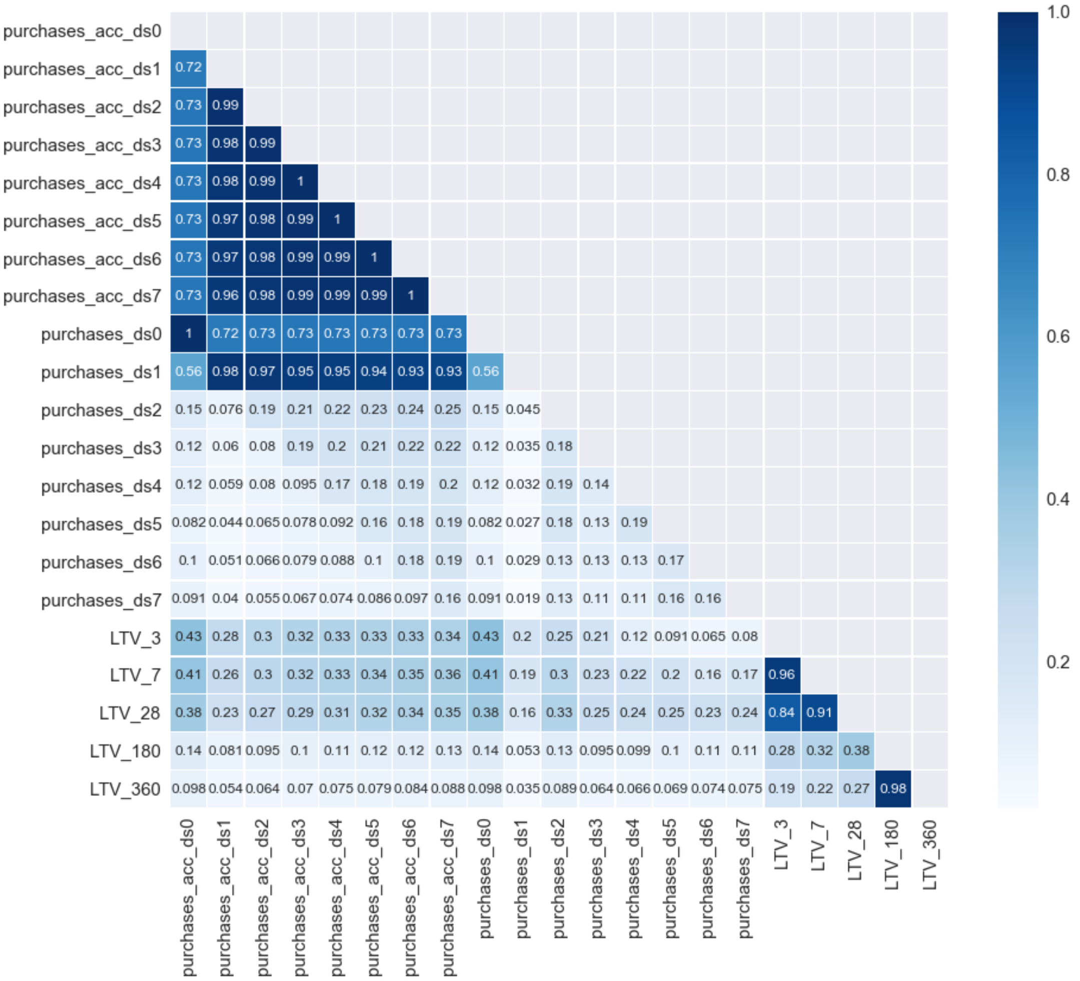
	purchases_ds0	purchases_ds1	purchases_ds2	purchases_ds3	purchases_ds4	purchases_ds5	purchases_ds6	purchases_ds7	LTV_3	LTV_7	LTV_28	LTV_180	LTV_360
purchases_ds0													
purchases_ds1	0.56												
purchases_ds2	0.15	0.045											
purchases_ds3	0.12	0.035	0.18										
purchases_ds4	0.12	0.032	0.19	0.14									
purchases_ds5	0.082	0.027	0.18	0.13	0.19								
purchases_ds6	0.1	0.029	0.13	0.13	0.13	0.17							
purchases_ds7	0.091	0.019	0.13	0.11	0.11	0.16	0.16						
LTV_3	0.43	0.2	0.25	0.21	0.12	0.091	0.065	0.08					
LTV_7	0.41	0.19	0.3	0.23	0.22	0.2	0.16	0.17	0.96				
LTV_28	0.38	0.16	0.33	0.25	0.24	0.25	0.23	0.24	0.84	0.91			
LTV_180	0.14	0.053	0.13	0.095	0.099	0.1	0.11	0.11	0.28	0.32	0.38		
LTV_360	0.098	0.035	0.089	0.064	0.066	0.069	0.074	0.075	0.19	0.22	0.27	0.98	

Figure 31 – Purchases and LTV correlation heatmap

All purchase analysis done so far regards the daily activity of each user. The cumulative number of purchases is analyzed in the following plots, in order to test for further correlation improvements. This variables are basically the accumulated sum of *purchases_dsx* until day x of observation. Figure 32 shows the correlation between the raw purchasing number, the accumulated values and the LTV target variables. Accumulated purchases presented a higher correlation with the LTV and, although highly correlated between themselves, the accumulated features are not that correlated with purchases itself.

5.2.2 Net revenue

Keeping the same methodology as in the purchase feature exploration, Figure 33 shows that net revenue distributions follow a more sparse configuration across revenue values when compared with purchase distributions. Interestingly, there is no clear pattern indicating that high net revenues are less frequent than low daily revenue values.



	purchases_acc_ds0	purchases_acc_ds1	purchases_acc_ds2	purchases_acc_ds3	purchases_acc_ds4	purchases_acc_ds5	purchases_acc_ds6	purchases_acc_ds7	purchases_ds0	purchases_ds1	purchases_ds2	purchases_ds3	purchases_ds4	purchases_ds5	purchases_ds6	purchases_ds7	LTV_3	LTV_7	LTV_28	LTV_180	LTV_360
purchases_acc_ds0																					
purchases_acc_ds1	0.72																				
purchases_acc_ds2	0.73	0.99																			
purchases_acc_ds3	0.73	0.98	0.99																		
purchases_acc_ds4	0.73	0.98	0.99	1																	
purchases_acc_ds5	0.73	0.97	0.98	0.99	1																
purchases_acc_ds6	0.73	0.97	0.98	0.99	0.99	1															
purchases_acc_ds7	0.73	0.96	0.98	0.99	0.99	0.99	1														
purchases_ds0	1	0.72	0.73	0.73	0.73	0.73	0.73	0.73													
purchases_ds1	0.56	0.98	0.97	0.95	0.95	0.94	0.93	0.93	0.56												
purchases_ds2	0.15	0.076	0.19	0.21	0.22	0.23	0.24	0.25	0.15	0.045											
purchases_ds3	0.12	0.06	0.08	0.19	0.2	0.21	0.22	0.22	0.12	0.035	0.18										
purchases_ds4	0.12	0.059	0.08	0.095	0.17	0.18	0.19	0.2	0.12	0.032	0.19	0.14									
purchases_ds5	0.082	0.044	0.065	0.078	0.092	0.16	0.18	0.19	0.082	0.027	0.18	0.13	0.19								
purchases_ds6	0.1	0.051	0.066	0.079	0.088	0.1	0.18	0.19	0.1	0.029	0.13	0.13	0.13	0.17							
purchases_ds7	0.091	0.04	0.055	0.067	0.074	0.086	0.097	0.16	0.091	0.019	0.13	0.11	0.11	0.16	0.16						
LTV_3	0.43	0.28	0.3	0.32	0.33	0.33	0.33	0.34	0.43	0.2	0.25	0.21	0.12	0.091	0.065	0.08					
LTV_7	0.41	0.26	0.3	0.32	0.33	0.34	0.35	0.36	0.41	0.19	0.3	0.23	0.22	0.2	0.16	0.17	0.96				
LTV_28	0.38	0.23	0.27	0.29	0.31	0.32	0.34	0.35	0.38	0.16	0.33	0.25	0.24	0.25	0.23	0.24	0.84	0.91			
LTV_180	0.14	0.081	0.095	0.1	0.11	0.12	0.12	0.13	0.14	0.053	0.13	0.095	0.099	0.1	0.11	0.11	0.28	0.32	0.38		
LTV_360	0.098	0.054	0.064	0.07	0.075	0.079	0.084	0.088	0.098	0.035	0.089	0.064	0.066	0.069	0.074	0.075	0.19	0.22	0.27	0.98	

Figure 32 – Accumulated purchase correlation with daily purchase and LTV variables

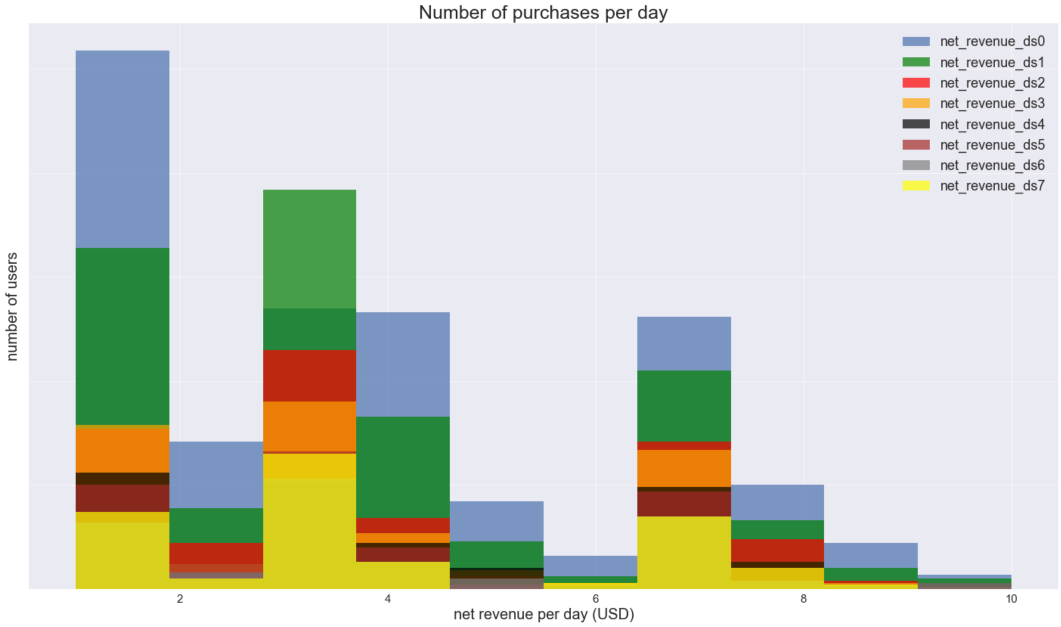


Figure 33 – Net revenue distribution across users by day of observation

Figure 34 proves that net revenue is a powerful feature in LTV prediction, having the highest correlations so far. The interesting insight is that net revenue generated in the first few days of activity is more relevant than revenue generated after some time. This is also coherent with the previous conclusion regarding number of purchases. Therefore, it also shows how important it is for the company to concentrate marketing efforts on players that just started to play. It seems like a potential spender is prone to spend in the first few days after downloading the game.

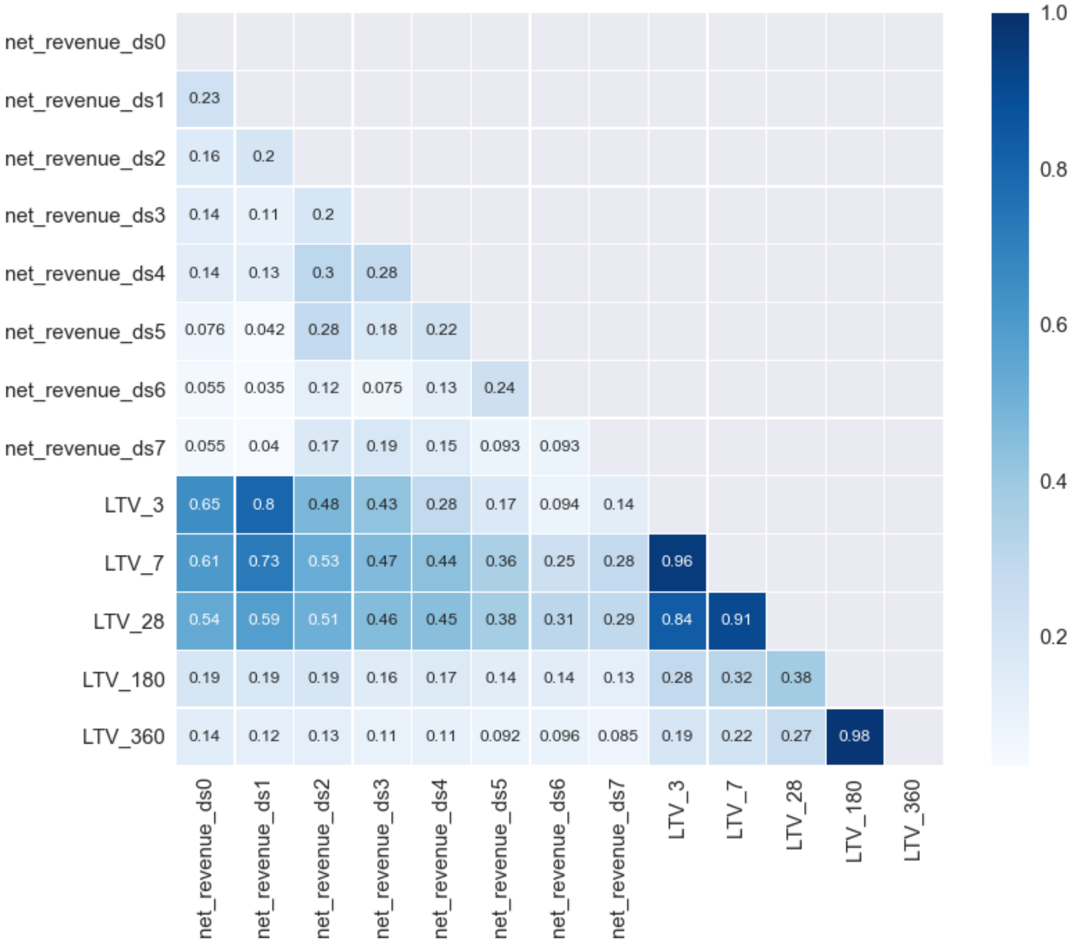


Figure 34 – Net revenue correlation with LTV variables

The accumulated net revenue (or LTV itself) generated by users until the seventh day of observation was also analyzed. The correlation between accumulates net revenue, net revenue and LTVs is shown in Figure 35.

Interestingly, the cumulative variables seems to have a strong relationship with LTV variables. If we check the previous analysis containing *net_revenue_dsx* against *net_revenue_acc_dsx*, we see that, comparatively, the second has a slightly higher correlation with the LTV. This can be seen in Figure 35, where the accumulated features show a darker blue color. Also, this superiority applies for the majority of correspondent purchase related features.

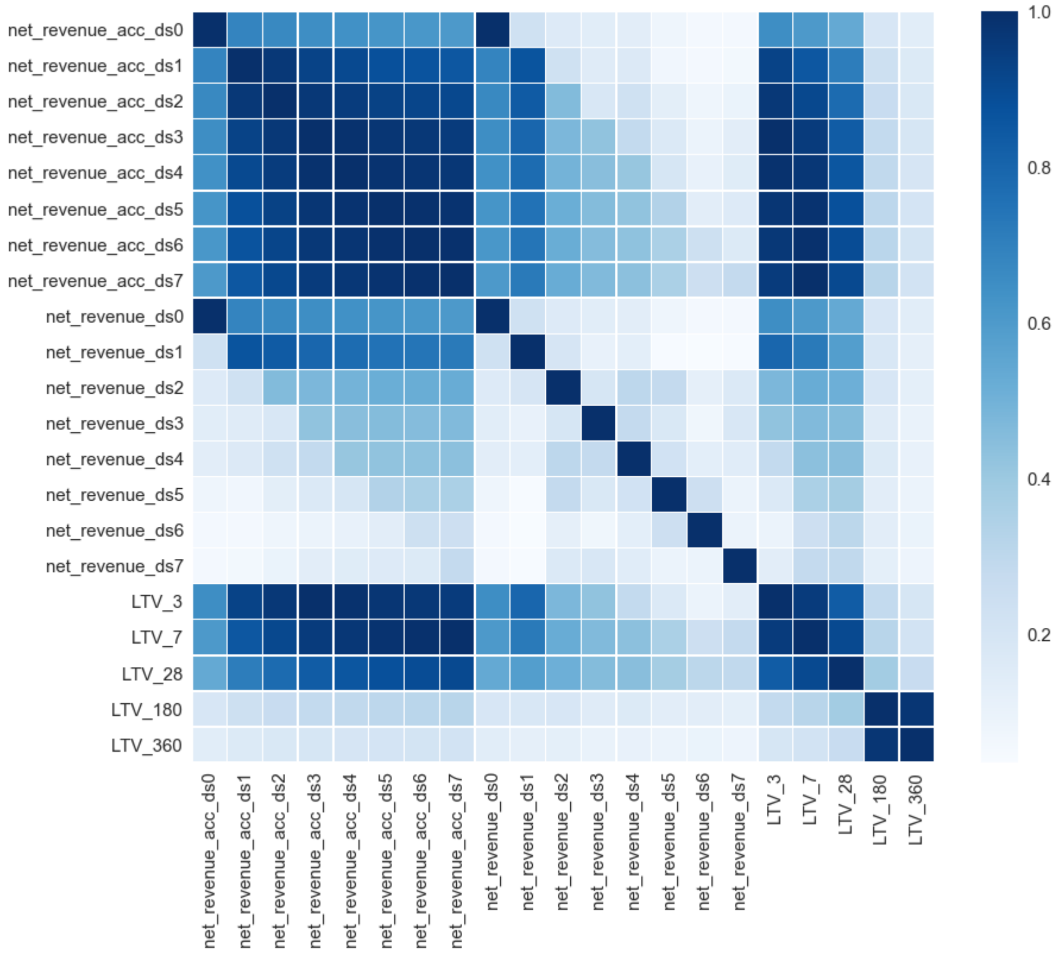


Figure 35 – Accumulated Net revenue correlation with LTV variables

Since revenue and number of purchases are intrinsically connected, we could analyze revenue/purchase to check how important the average value per purchase is on LTV prediction. To do so, 7 new columns were created: *net_revenue_dsx/purchases_dsx* for x in [0,7] as always. This columns give the idea of the purchase weight in value.

In Figure 37 we see that average revenue per purchase has also high correlation with the target LTV variables. Those features might be included in the training set during the feature engineering step.

As a final conclusion, the purchase-related features are obviously the most important for the regression task, as described in the literature review. The analysis also showed that accumulated information is more correlated with LTV values and, consequently, more relevant for the problem. Furthermore, revenue per purchase also showed a higher correlation than the two features alone, proving that the value of each purchase has more weight in LTV prediction and must be integrated in the Feature Engineering step. The correlation heatmap between them and the LTV variables is showed in Figure 38.

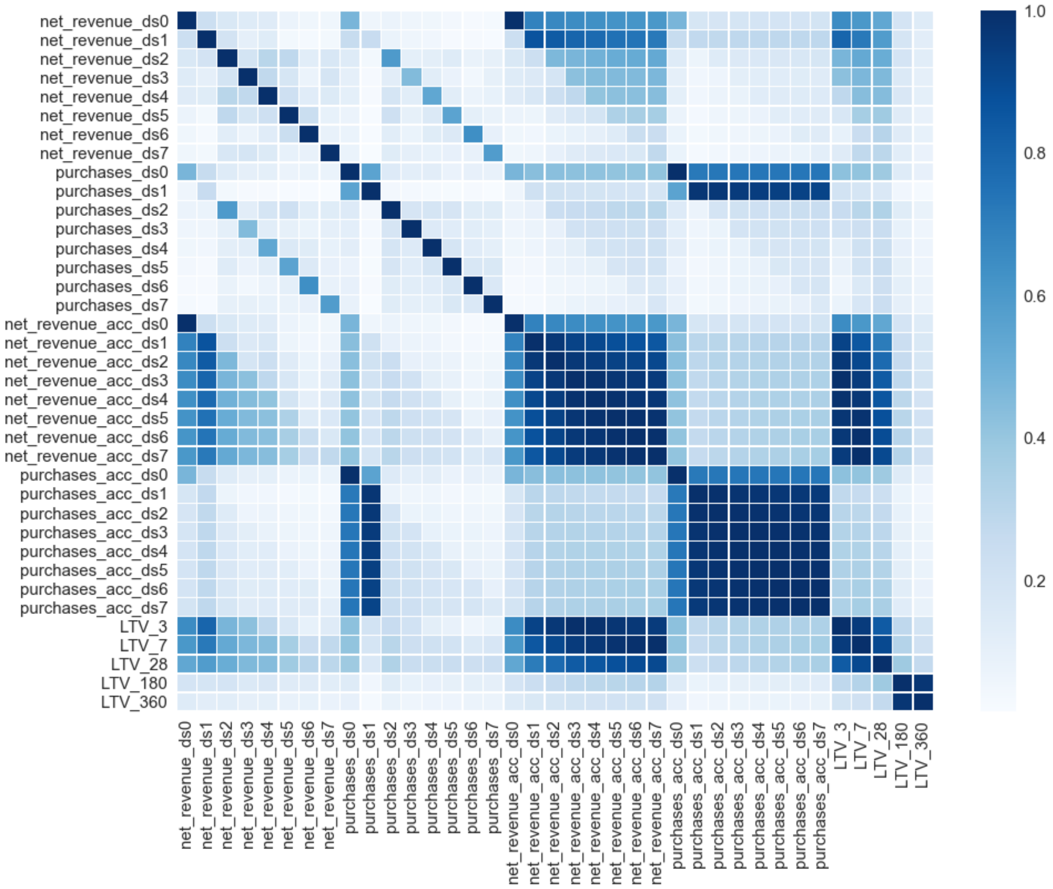


Figure 36 – All purchase-related features correlation

5.3 Event-related features

As previously stated, the Event-related features deal with in-game actions and attributes that the users can interact with. To better distinguish between these different elements, the following subsections describe them is smaller groups that are somehow related. Since the overall objective is to identify the correlation of features with the different LTV metrics created, several correlation matrices were created to have an improved perception of these relationships and to capture the most relevant elements.

5.3.1 Tutorial evolution

Inside Castle Crush, when a user first activates the application, a tutorial can be done to better familiarize the user with the basic instructions to play it properly. This tutorial consists of six parts.

Initially, to have a better overview of the user engagement in the tutorial provided to them, it is interesting to see how many of them actually take part of the tutorial. In the dataset analysed, a very low percentage of users actually started the tutorial. More precisely, only 0.04% of the users did start the tutorial.

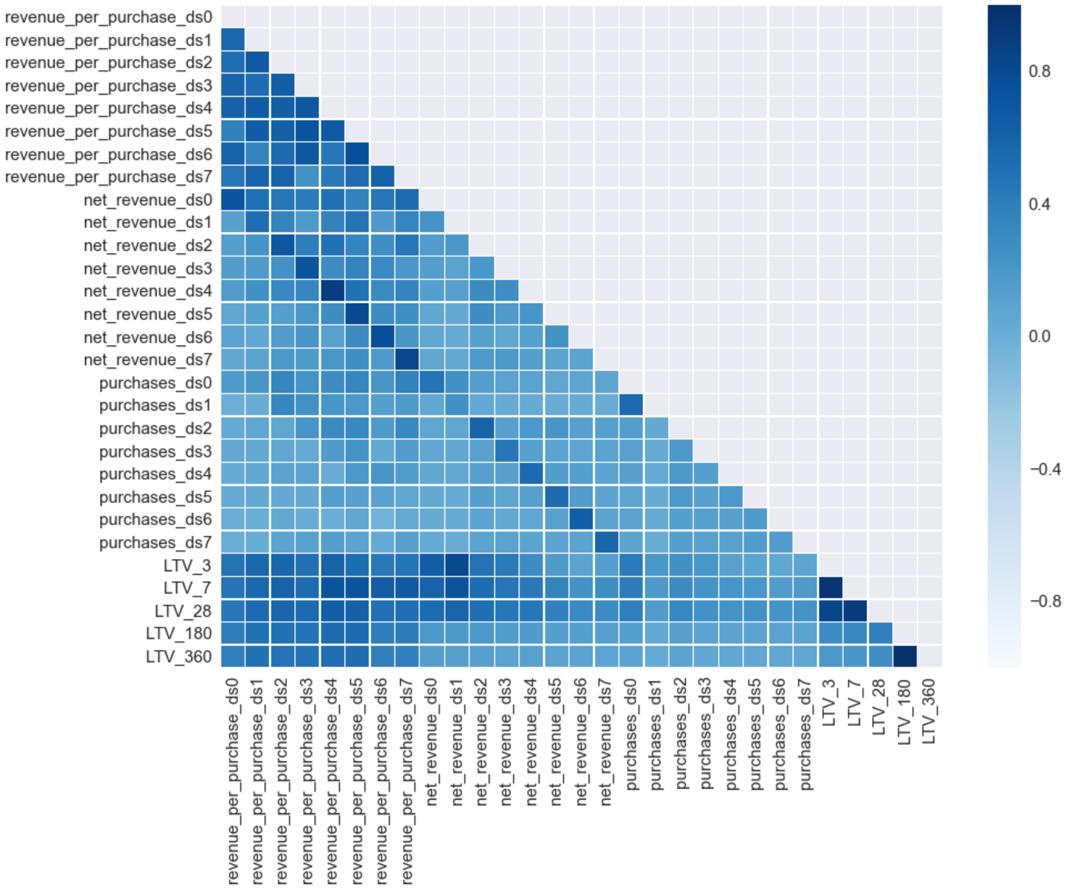


Figure 37 – All purchase-related features correlation

Also, Figure 39 shows that there is a very uniform quantity of users that start the six different parts of the tutorial, indicating that the majority players that usually start the first part, end up taking part in all the six ones.

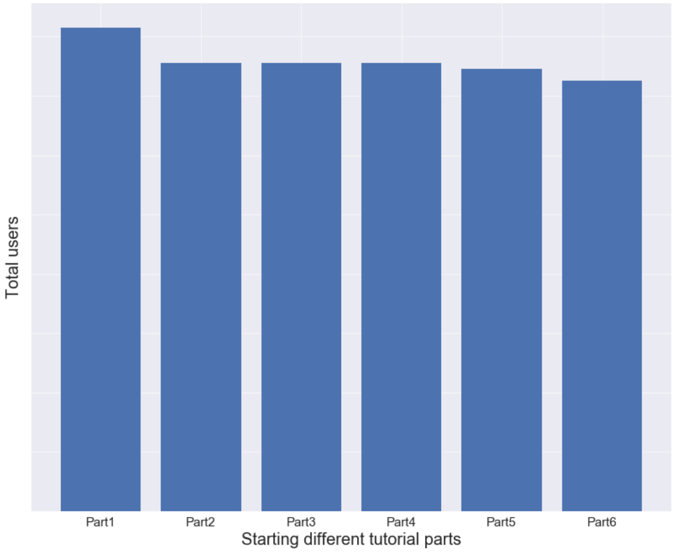


Figure 39 – Number of users starting the six different tutorial parts

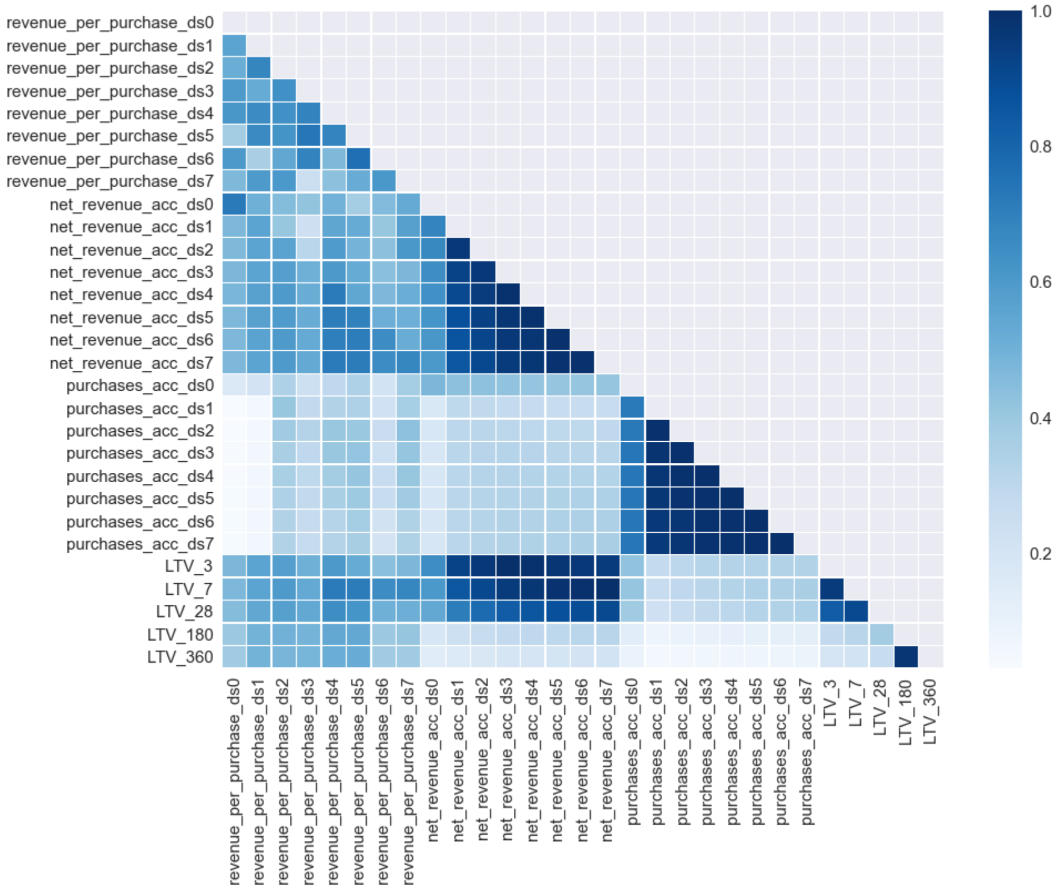


Figure 38 – Correlation of best purchase-related features

Additionally, the raw analysis indicated in Figure 40 shows that Tutorial is, in general, uncorrelated with LTV. However, the combination of tutorial conclusion may provide a useful information for LTV prediction. The cumulative features involving sequential Tutorial conclusions were developed in order to test the hypothesis that a user that concluded all steps of the tutorial is probably a valuable user. The correlation matrix of those variables is presented in Figure 41.

The light blue color showed in both correlation matrices reaffirms that Tutorial-related features are definitely not well correlated with LTV, even in the cumulative variation.

5.3.2 Battle and Session

The features indicated in this section are related to game battles (starting, winning or losing them) and starting in-app sessions. It can be seen in Figure 42 that the correlation values are extremely low for these features. This indicates that the number of battles or sessions started during the first days of the user playing the game have little influence on its LTV.

Similarly, Figure 43 pictures that the total number of victories or losses in battles

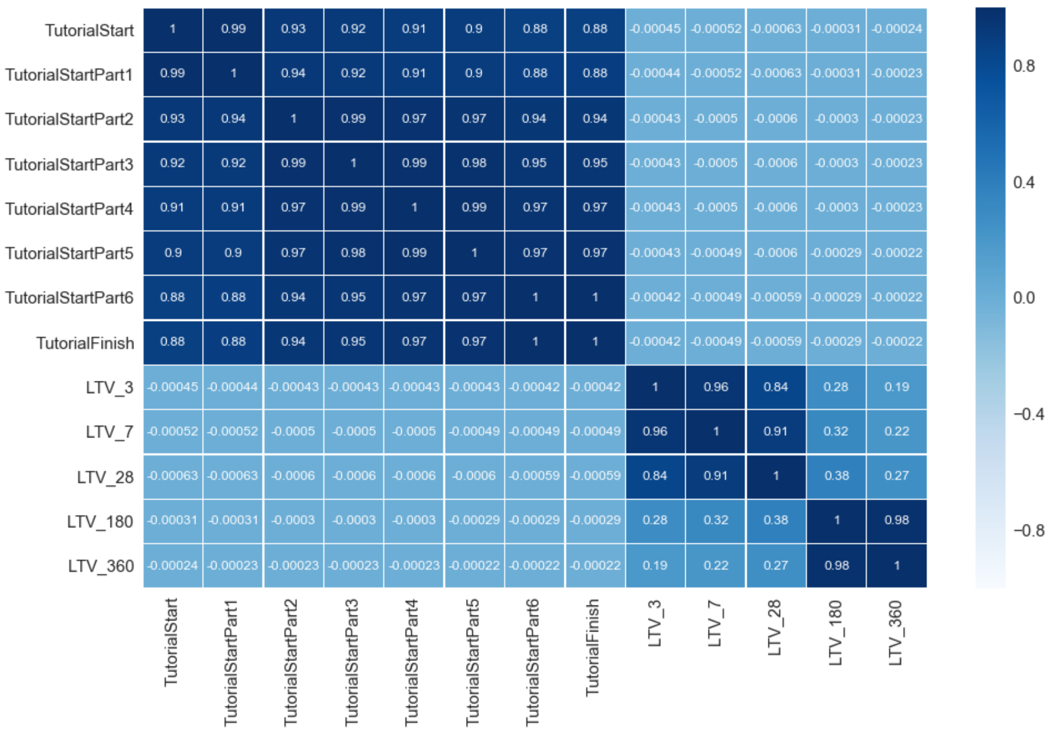


Figure 40 – Correlation of Tutorial-related features and LTVs

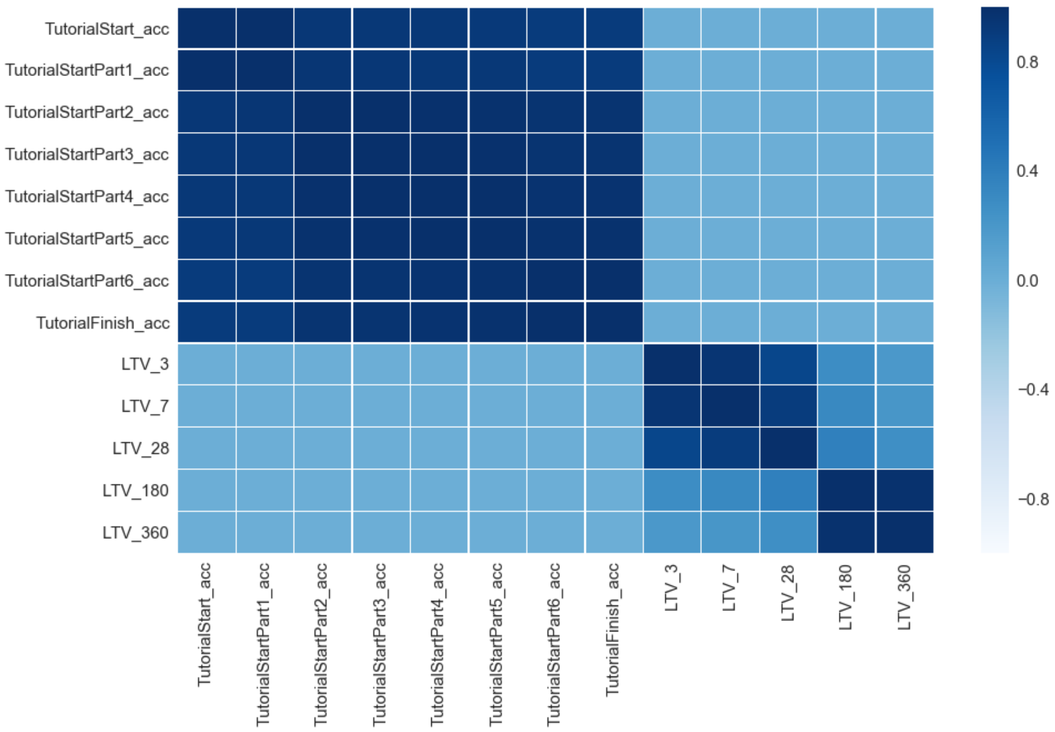


Figure 41 – Correlation of cumulative Tutorial-related features and LTVs

in the first days of playing does not seem to have a significant impact on users’ spending.

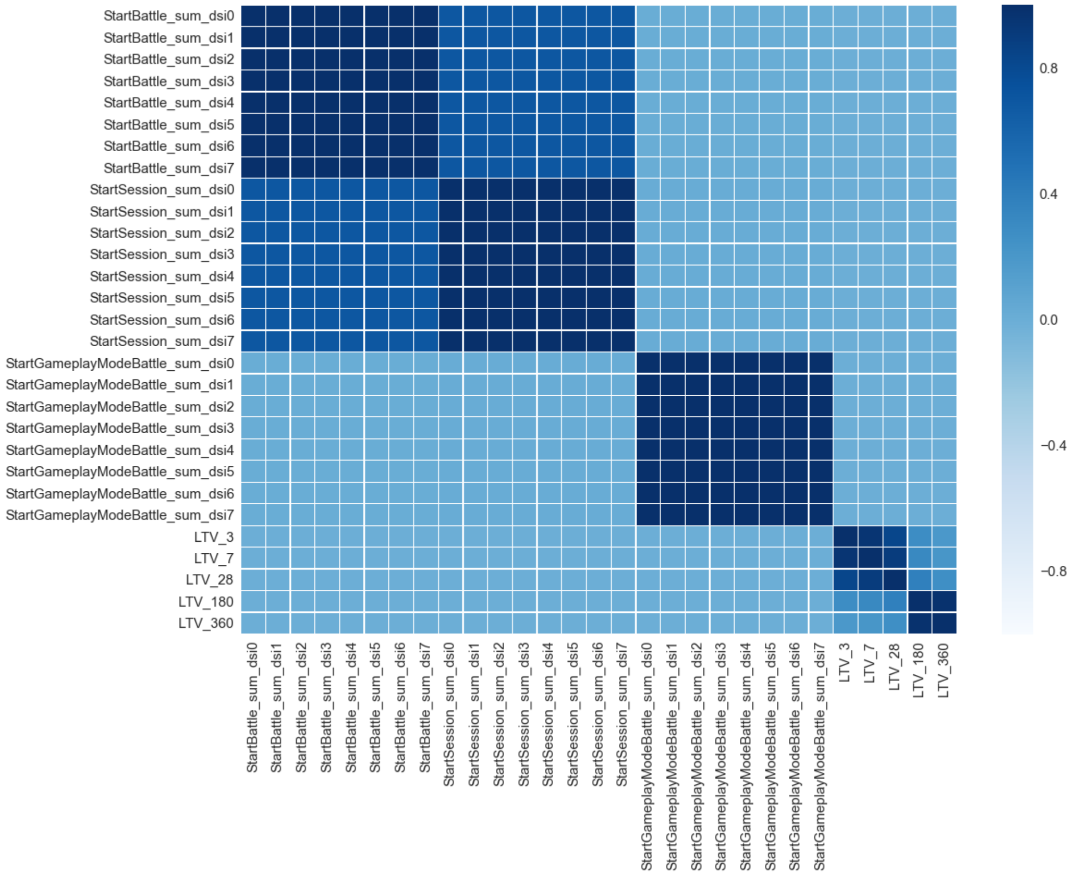


Figure 42 – Correlation of Battle and Session features and LTVs

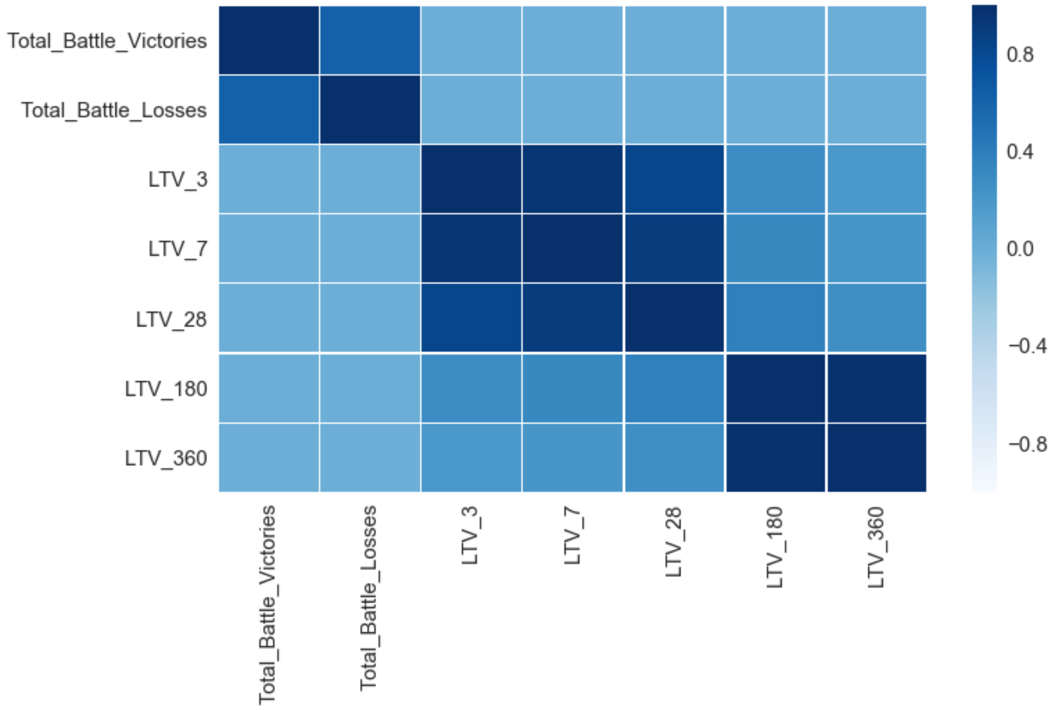


Figure 43 – Correlation of Winning or Losing Battles and LTVs

5.3.3 Piggybank and Chest

The Piggybank and Chest features also don't have a significant correlation with our variables of interest. From the correlation matrix presented in Figure 44, it is also visible that PiggyBank related features are somehow correlated wit OpenChest variations, indicating that users' behaviour within the game are related to each other in this case.

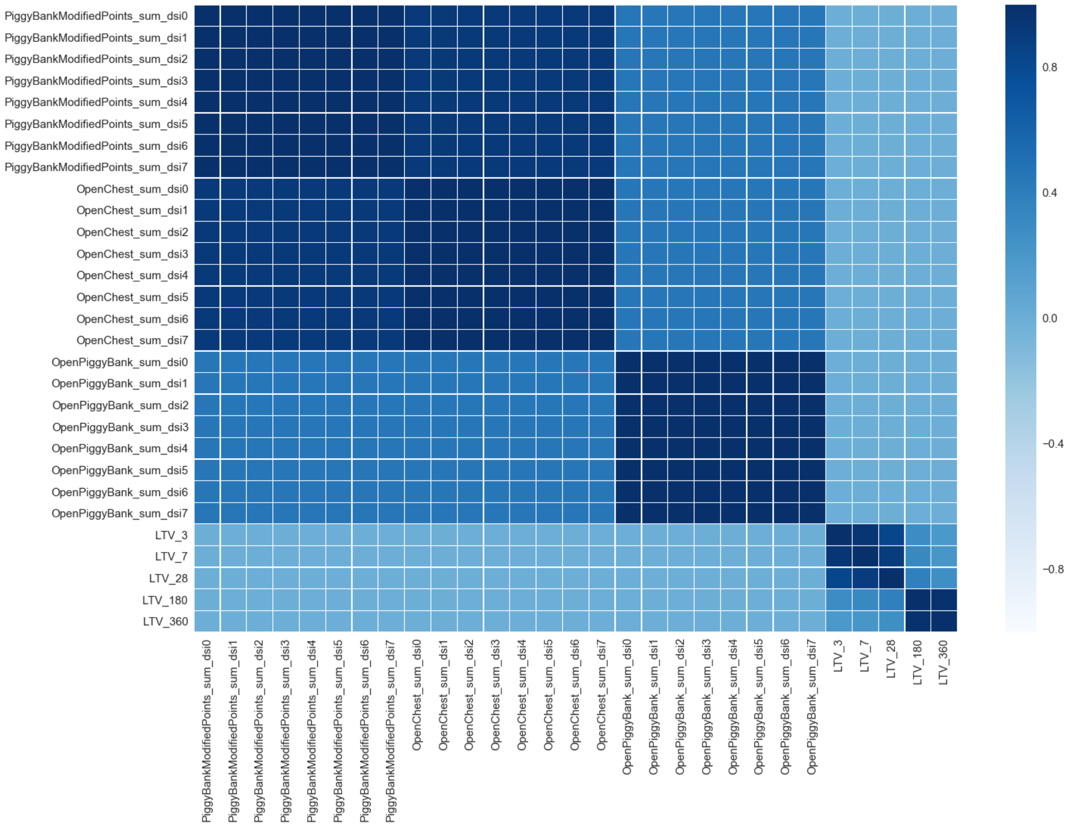


Figure 44 – Correlation of Piggybank and Chest features and LTVs

5.3.4 Cards

Upgrading or buying new cards does not have a meaningful impact on users spending in all the analyzed periods. Also, upgrading a card has a modest higher correlation with LTV than buying a new card, specially when looking at activation day (D+0). Correlations are shown in Figure 45.

5.3.5 Shop interaction

Within the app, the user can enter she shop to make purchases of game-related features. The first interesting behaviour to understand is the interaction with the shop in the first days of playing. As shown in 46, an extremely small part of players enters the shop in its first days, which restates that these features might not be the most appropriate for our LTV analysis.

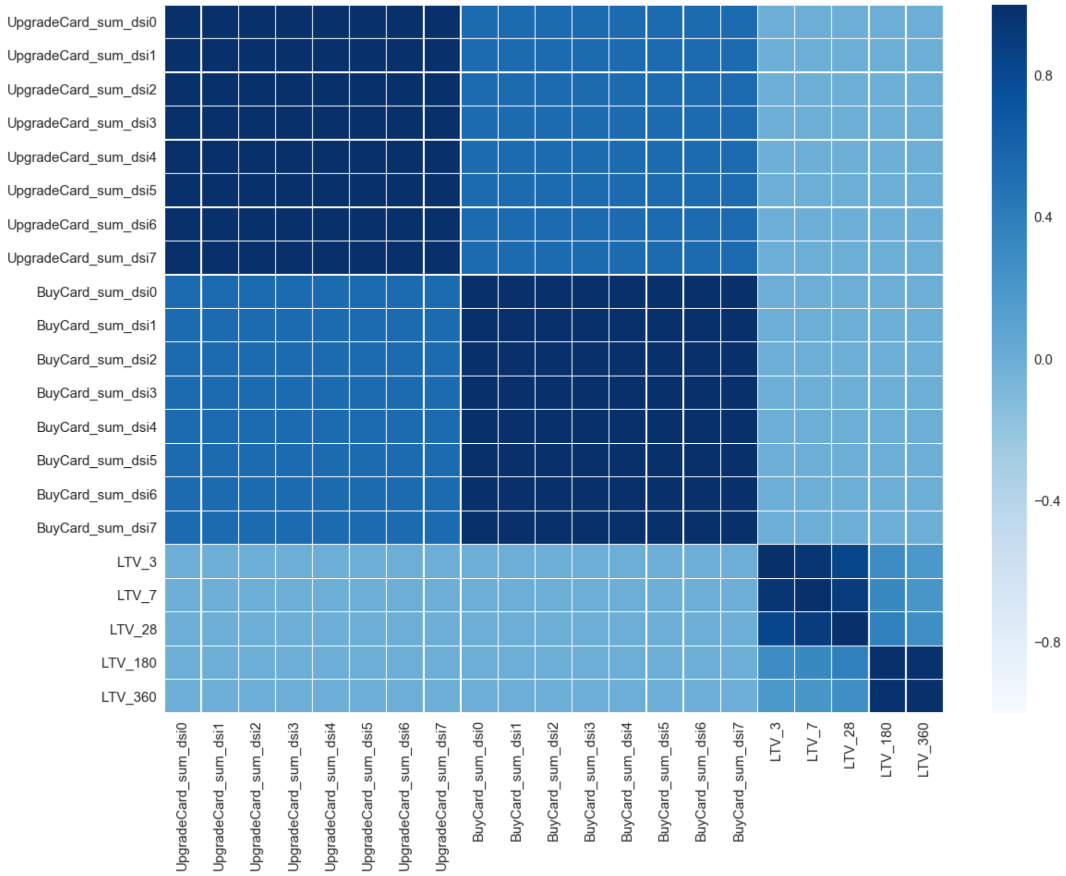


Figure 45 – Correlation of Cards features and LTVs

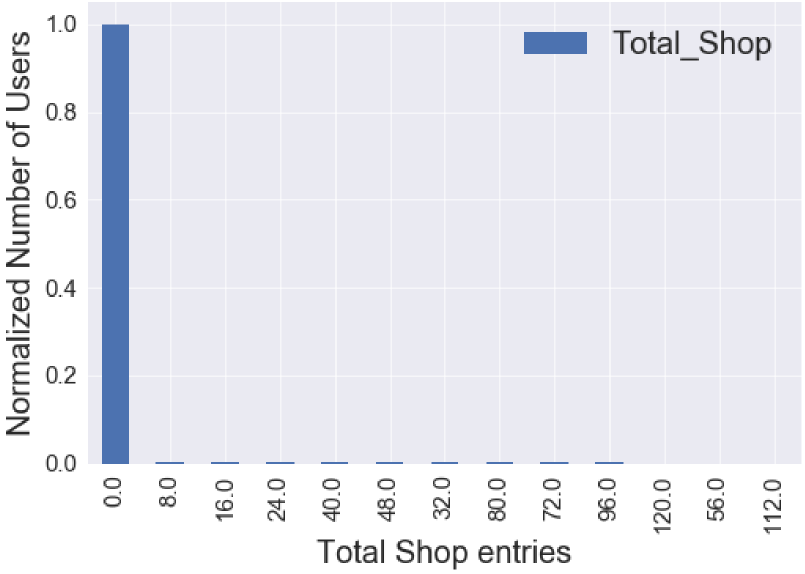


Figure 46 – User interaction with the shop in the first 7 days of playing (Normalized)

In fact, when plotting the correlation of entering the in-game shop with the LTVs, it is noticeable that the values are significantly low, as seen in Figure 47. Despite the small difference, entering the shop in earlier days is more correlated with players' expenditure than during the 7th day after installation, for example.

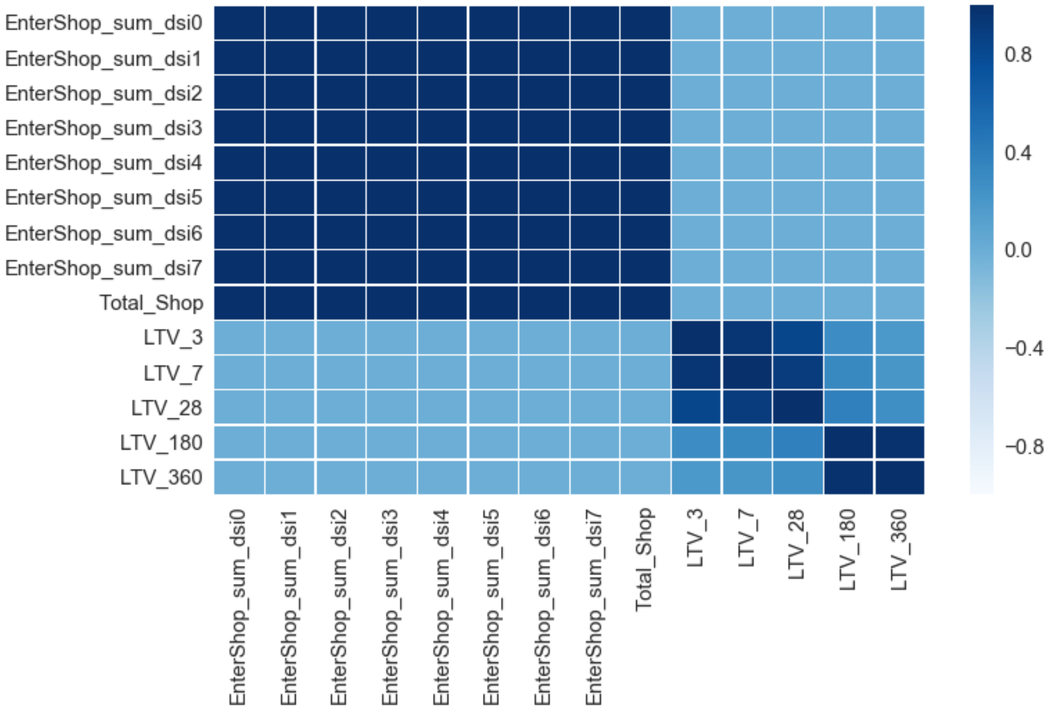


Figure 47 – Correlation of Shop interaction features and LTVs

5.3.6 Tournament

When analyzing the relationship of Tournament related features with the Lifetime Value of users an extremely low correlation was noticed. To better understand why, a computation of the amount of tournament battles started or joined was done.

For the first analysis, no users had participated in tournaments, which raised doubts regarding the size of the data sample collected. To deal with that, the dataset size was extended four times in size, however, the data revealed that only two users from the total of over a million data points analyzed had started or joined a tournament battle in the first days of playing. This showed that an extremely low percentage of the users participate in tournament battles. Therefore, due to the lack of representation of this group of features, it was decided to drop them from the analysis, which will be done in the Data Preprocessing section.

This insight is also quite interesting, indicating that users usually have a very low participation in tournament during the first days of playing the game. As an observation to Wildlife, some measures could be taken in order to make increase players' engagement, such as showing more in-game alerts regarding tournaments, making the participation of at least one tournament obligatory during the first days or giving prizes for users that join and start a tournament battle in the initial days of playing the game.

5.3.7 Arena evolution

As the user evolves in the game, he progresses by changing arenas. Intuitively, one would think that the amount of battles' victories and losses would influence the arena evolution of a player, making him more engaged and skilled in the game. In fact, as pointed out by Figure 48, there is a clear correlation between these features and, surprisingly, the amount of losses positively affects the arena evolution. This observation indicates that when the user plays battles, he becomes somehow more skilled in the game, even when he losses.

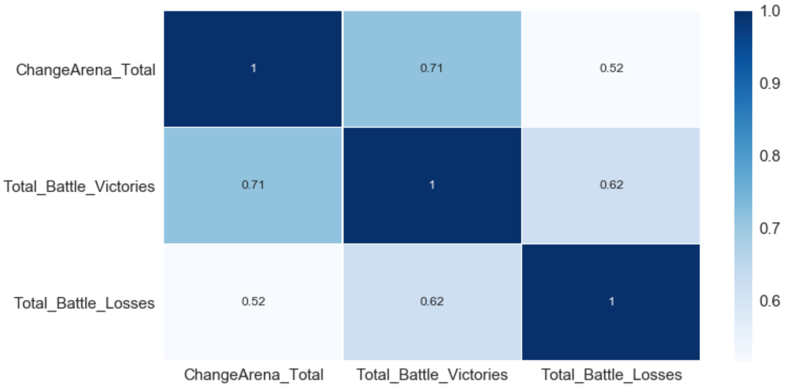


Figure 48 – Correlation of battles and arena evolution

Also, when checking the relationship of the users' arena evolution with our variables of interest (LTVs), we can see in Figure 49 that those features are extremely low correlated with the users' LTV, independent of the period of analysis.

6 Data Preprocessing

As stated before, the raw dataset must be preprocessed before serving as input to any Machine Learning algorithm. For seek of simplicity, the Data Transformation step, including data normalization or any other changes in the dataset, were also included in this section. Preprocessing was divided in 5 substeps, following the KDD pipeline described in Section 15:

- Dummification
- Feature Selection
- Feature Engineering
- Data Balancing
- Outliers treatment
- Scaling

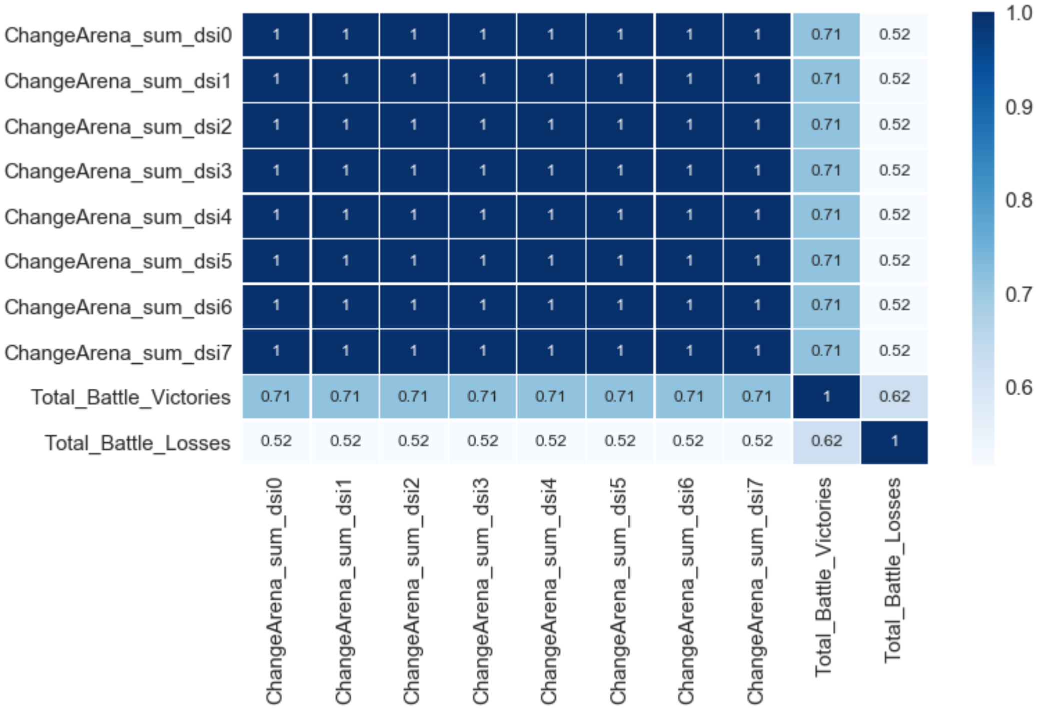


Figure 49 – Correlation of Arena features and LTVs

6.1 Dummification

Feature dummification is one kind of one-hot encoding, where we take a categorical variable and transform it into several columns taking only values of 1 and 0, with as many columns as different values we had in the original feature. The dummies strategy is the most commonly used and simplest transformation strategy of categorical features.

This process is commonly placed in the Data Transformation section within a KDD framework (see Section 15). However, since other preprocessing steps will involve feature selection based on correlation levels, this transformation step was placed in this section, following a temporal order. The dummification was done on features from Table 1 through the Pandas method *get_dummies*, broadly applied within the data science and machine learning fields.

6.2 Feature Selection

Feature Selection is the process where you automatically or manually select those features which contribute the most to the prediction variable or output in which you are interested in. In this case, since we have a lot of features analyzed (see Section 5), a feature selection methodology was developed based on 3 correlation criteria:

1. Correlation threshold: only features with more than 0.005 correlation with LTV 180 must be included

2. Minimum number of features: each feature category must have at least 1 representative included in the master dataset if correlation is greater than 0.0001

3. Maximum number of features: each feature can have a maximum number of 5 members across days sum and their correlation must be less than 0.8

These criteria were arbitrarily chosen taking into account the low correlations found during the Data Exploration section. The first criteria sets a threshold to limit non meaningful features for LTV prediction. The second criteria was developed with the objective to include at least some information about features with low (but not none) correlation with LTV. Finally, the third criteria aims no limit overfitting on machine learning models with highly correlated features.

To generalize these 3 criteria, 2 functions were developed and implemented in each feature. The first one, ranks features correlation module with LTV 180. The second performs a switch case that implements the 3 criteria information. The following section shows the details of feature selection for each feature category.

6.2.1 Country

Table 4 – country top10 correlation ranking

Feature	LTV_180 correlation
country_TH	0.013037
country_HK	0.010198
country_WW	0.008643
country_US	0.008271
country_IN	-0.005702
country_CN	0.005285
country_JP	0.004498
country_IS	0.004182
country_BR	-0.003507
country_TW	0.003207

Thailand, Hong Kong, WorldWide, US and India are the top 5 countries that have more than 0.005 of correlation with LTV 180. These countries were selected to be in the master training set, since they also fit in the other selection criteria.

6.2.2 Platform

Table 5 – platform correlation ranking

Feature	LTV_180 correlation
platform_iOS	0.023262
platform_Android	-0.023262

Table 5 shows platforms correlation with LTV 180 as showed in Section 3. Since it is a binary variable, only one column from the dummification of this feature is included in the master training set. In this case *platform_iOS* was arbitrarily chosen.

6.2.3 Source

Table 6 – source top10 correlation ranking

Feature	LTV_180 correlation
source_channel_10	0.018210
source_channel_1	-0.012242
source_channel_4	0.009803
source_channel_3	0.008396
source_channel_17	0.007760
source_channel_7	0.003305
source_channel_13	0.002543
source_channel_15	0.001255
source_channel_20	0.001182
source_channel_9	0.001088

The dummie features source_channel_10, source_channel_1, source_channel_4, source_channel_3 and source_channel_17 presented a correlation module greater than 0.005 to pass in the first criteria. All other criteria are also met.

6.2.4 Device

The Exploratory Data Analysis Section showed that Device and Platform are highly correlated features and they do not pass the third criterion. Therefore, in this analysis, device was dropped as a feature.

6.2.5 Purchases

Table 7 shows the correlation ranking among number of purchases across observation days.

Table 7 – Purchase number correlation ranking

Feature	LTV_180 correlation
purchases_ds0	0.140550
purchases_ds2	0.131740
purchases_ds6	0.109077
purchases_ds7	0.107009
purchases_ds5	0.101149
purchases_ds4	0.098712
purchases_ds3	0.094964
purchases_ds1	0.052707

Interestingly, number of purchases in the day of download presents the highest correlation with LTV 180. It means that a user that starts buying a lot on his first day is prone to keep spending through the rest of his lifetime within the game. the Top5 features from the table were selected to be in the master dataset: purchases_ds0, purchases_ds2, purchases_ds6 , purchases_ds7 and purchases_ds5.

6.2.6 Net revenue

As analyzed in Section 5, revenue-related features are the most important, since LTV is directly derived from these features. Table 8 show their correlation ranking with LTV 180.

Table 8 – net_revenue correlation ranking

Feature	LTV_180 correlation
net_revenue_ds0	0.192261
net_revenue_ds2	0.191409
net_revenue_ds1	0.187231
net_revenue_ds4	0.166582
net_revenue_ds3	0.155095
net_revenue_ds6	0.138864
net_revenue_ds5	0.138367
net_revenue_ds7	0.125537

Following the selection criteria, the Top5 features were selected to be in the master dataset: net_revenue_ds0, net_revenue_ds2, net_revenue_ds1, net_revenue_ds4, net_revenue_ds3, net_revenue_ds6, net_revenue_ds5 and net_revenue_ds7.

6.2.7 Event-related features

Event-related features prooved to be the less correlated with LTV, as shown in Section 5. The correlation range is between -0.0005 and 0.0005 on average, which does not fit the first threshold criterion. For this reason, these features were analyzed together, being considered as from the same group of features to be selected.

The second correlation criteria was applicable to all these features, and ds7 features were selected for each event related feature. Tournament-related features were dropped, since they showed only zeros, which mean that no member of this dataset participated in any tournament in the first 7 days of use. The correlations between the selected variables and LTV 180 are shown in table 9.

Table 9 – Selected event-related features correlation ranking

Feature	LTV_180 correlation
ChangeArena_sum_dsi7	-0.000263
OpenChest_sum_dsi7	-0.000251
EnterDeck_sum_dsi7	-0.000216
UpgradeCard_sum_dsi7	-0.000212
StartSession_sum_dsi7	-0.000207
PiggyBankModifiedPoints_sum_dsi7	-0.000198
StartBattle_sum_dsi7	-0.000180
EnterShop_sum_dsi7	-0.000178
WinBattle_sum_dsi7	-0.000175
LoseBattle_sum_dsi7	-0.000141
BuyCard_sum_dsi7	-0.000113

6.3 Feature Engineering

Feature engineering is the process of using domain knowledge to extract features from raw data. These features can be used to improve the performance and computational effort of machine learning algorithms.

As showed in the Section 5, accumulated variables regarding purchase and net revenue information and revenue per purchase presented high correlations with the target variable. In this section, these features are created and selected according to the same criteria from Section 6.5.

6.3.1 Accumulated purchases

Table 7 presents the correlation ranking of the cumulative sum of purchases across days of observation. As expected, all features meet the first criteria and the top 5 accumulated purchases were selected: purchases_acc_ds0,purchases_acc_ds7, purchases_acc_ds6, purchases_acc_ds5 and purchases_acc_ds4.

Table 10 – Accumulated purchases correlation ranking

Feature	LTV_180 correlation
purchases_acc_ds0	0.140550
purchases_acc_ds7	0.129952
purchases_acc_ds6	0.124418
purchases_acc_ds5	0.117146
purchases_acc_ds4	0.110723
purchases_acc_ds3	0.104237
purchases_acc_ds2	0.094955
purchases_acc_ds1	0.080740

6.3.2 Accumulated Net revenue

Table 11 presents the correlation ranking of the cumulative sum of net revenue across days of observation. As expected, all features meet the first criteria and the top 5 accumulated purchases were selected: net_revenue_acc_ds7, net_revenue_acc_ds6, net_revenue_acc_ds5, net_revenue_acc_ds4 and net_revenue_acc_ds3,

Since accumulated net revenue is the same as LTV, it directly influences LTV prediction and present the really high correlations with LTV 180.

Table 11 – Accumulated Net revenue correlation ranking

Feature	LTV_180 correlation
net_revenue_acc_ds7	0.317710
net_revenue_acc_ds6	0.311061
net_revenue_acc_ds5	0.302670
net_revenue_acc_ds4	0.293526
net_revenue_acc_ds3	0.284042
net_revenue_acc_ds2	0.265674
net_revenue_acc_ds1	0.238540
net_revenue_acc_ds0	0.192261

6.3.3 Net revenue per purchase

The average ticket in a purchasing day of a user is described with this feature. Table 12 shows the correlation between revenue per purchase and LTV 180 in each observation day. This feature is most correlated feature found so far and is expected to be important in the prediction task.

Table 12 – Net revenue per purchase correlation ranking

Feature	LTV_180 correlation
revenue_per_purchase_ds5	0.539521
revenue_per_purchase_ds4	0.539159
revenue_per_purchase_ds2	0.502345
revenue_per_purchase_ds1	0.495222
revenue_per_purchase_ds3	0.491884
revenue_per_purchase_ds7	0.415793
revenue_per_purchase_ds0	0.401631
revenue_per_purchase_ds6	0.399531

6.3.4 Last feature selection

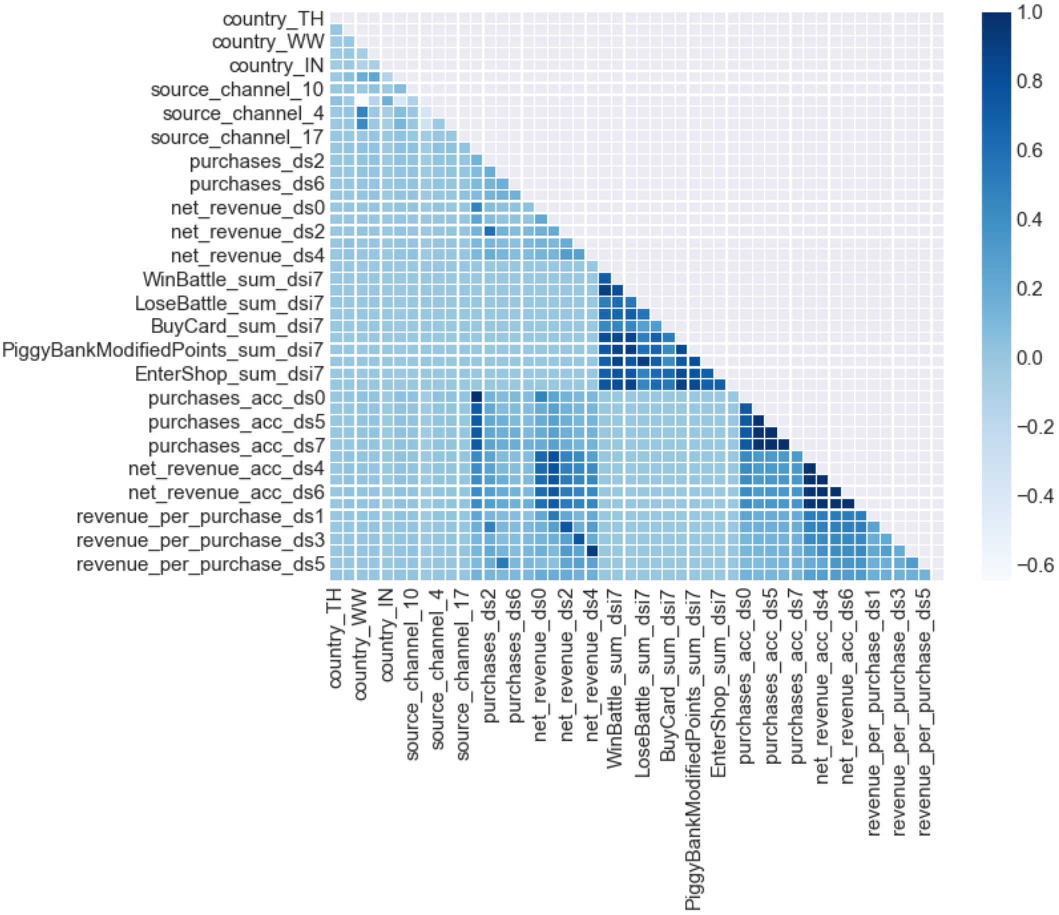


Figure 50 – Correlation matrix between features before final selection

The correlation heatmap plotted in Figure 50 shows that many features are highly correlated, as expected, since many of them have the same origin. To solve this problem and prevent overfitting, features pairs were analyzed individually and the one with lower correlation with LTV 180 was dropped. In this process, 16 features were removed. Table 13 shows the final set of features that were selected to be in the final master training set.

Table 13 – Final selected features

Feature
country_TH
country_HK
country_WW
country_US
country_IN
platform_iOS
source_channel_10
source_channel_1
source_channel_4
source_channel_3
source_channel_17
purchases_ds0
purchases_ds2
purchases_ds5
purchases_ds6
purchases_ds7
net_revenue_ds0
net_revenue_ds1
net_revenue_ds2
net_revenue_ds3
ChangeArena_sum_dsi7
BuyCard_sum_dsi7
StartSession_sum_dsi7
purchases_acc_ds0
purchases_acc_ds7
net_revenue_acc_ds7
revenue_per_purchase_ds1
revenue_per_purchase_ds2
revenue_per_purchase_ds3
revenue_per_purchase_ds4
revenue_per_purchase_ds5

6.4 Data Balancing

As stated in Sections 1 and 2, data imbalance is the main issue when we talk about LTV prediction in games. That is because usually only a small set of users are actually paying (LTV>0). (SIFA et al., 2015; DRACHEN et al., 2018; SIFA et al., 2018) solved this problem applying SMOTE (Synthetic Minority Over-sampling Technique) in order to generate synthetic samples from random combinations of the small set of paying users in the real dataset. The literature review proved that this step consistently improved models out-of-sample performance.

In this work, the Data Balancing step was performed in another way. Since Wildlife's database is huge, the master training set was balanced with real paying users. This process is more realistic and computationally less expensive comparing to SMOTE. To do so, 93000 paying users from 2018 were selected to balance the training set, replacing 93000 non paying users. The final table had roughly 50% paying and 50% non paying users.

6.5 Outliers Treatment

To deal with the presence of outliers on the dataset, the first step was to have a visualization of the data points through a scatter plot, which is represented in Figure 51. Due to privacy concerns, the customers' LTV were normalized to avoid revealing the nominal values. In the top part of Figure 51, it can be seen that there is a significant concentration of users under the red line crossing the 0.05 vertical mark, and only a few exceptions above it. Therefore, digging deeper on the LTV distribution analysis, the second chart shows the same data as the first one, but limiting the target variable to 0.05 and plotting the red line on the 0.01 normalized dollar equivalent.

The total number of customers above the 0.01 LTV mark represent around 8% of the paying users, which clearly indicates a very small portion of data. Thus, those players were removed from the analysis so that the algorithms could have a better overall performance.

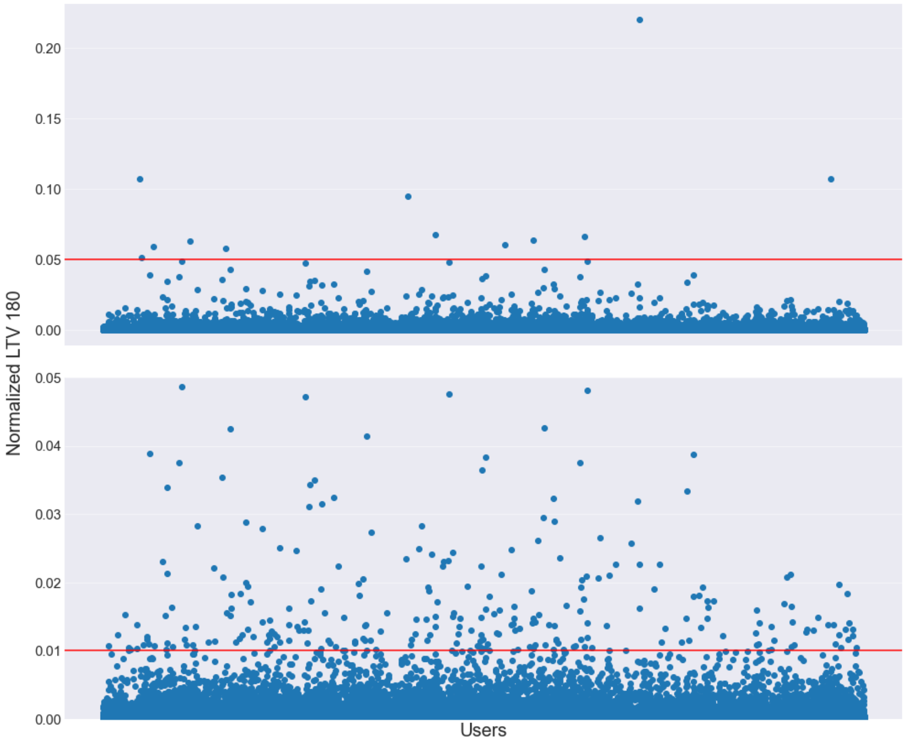


Figure 51 – LTV outliers detection

7 Model Selection

This section will deal with the process used to select the best model to fit the data. To achieve this, the master dataset was split in training set and test set right before any exploration or preprocessing step. It guarantees that final results in the test set are reliable and completely out-of-sample. To do so, the method *train_test_split* from *sklearn* was used and the dataset was randomly divided into training and testing sub-sets, holding a proportion of 80% to 20%, respectively. Only after the training set was properly explored and preprocessing steps were completely defined, the testing sub-set was treated. In this case, the steps Feature Selection and Scaling were performed, based on the same criteria developed for the training set.

Several regressors from the *sklearn* library were applied onto the training and test datasets. The following classes and regressors were used.

Table 14 – Applied Machine Learning Algorithms

Class	Names
Linear Model	Ridge, Lasso
Ensemble	RandomForestRegressor, AdaBoostRegressor, XGBoostRegressor
Neighbors	KNeighborRegressor
Neural Networks	MLPRgressor

These regressors were chosen accordingly to the main works in LTV prediction using Machine Learning, which are better discussed in Section 2. Each model was tuned using the *GridSearch* method, using 5 folds for cross-validation and Root Mean Squared Error (RMSE) as the optimization metric. *GridSearch* allows us to quickly test any combination of hyper-parameters for a single model and the best estimator is easily retrieved. The following table shows the final results and metrics evaluated for each regressor.

Table 15 – Final results

Regressor	CV Score	RMSE	MAE	R^2	Hit Rate Recall	Hit Rate F1 Score	RMSE paying	MAE paying	R^2 paying
Lasso	11.81	14.53	3.51	0.44	0.82	0.10	129.64	19.67	0.47
Ridge	11.67	14.57	2.88	0.44	0.84	0.11	131.31	20.60	0.45
MLP	11.51	15.51	2.46	0.37	0.78	0.12	140.96	21.99	0.37
RF	11.54	18.76	2.65	0.07	0.84	0.10	171.09	25.60	0.07
XGBoost	11.51	18.79	2.77	0.07	0.82	0.10	171.50	25.74	0.07
KNN	12.48	18.88	1.25	0.06	0.71	0.15	171.85	26.56	0.06
AdaBoost	13.27	20.35	7.71	-0.09	1.00	0.02	171.78	30.80	0.06

First, the cross-validation score is evaluated and it is the only metric in this table that involves a training score. All other metrics are calculated comparing the predicted values for test dataset and the true test values. CV score is evaluated taking the mean

RMSE across the 5 folds in the cross-validation. MLP and XGBoost slightly outperform the other models.

The second is the test RMSE in which Lasso and Ridge outperform the other models. The Mean Absolute Error shows the average absolute difference between test and prediction values, and KNN outperforms the other models. R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. In this case, this metric tell us how well LTV 180 is explained by the input variables. Interestingly, Lasso and Ridge are the best models regarding relationship between input and output variables, although they are really simple models. It worth mentioning how RF, XGBoost, KNN and AdaBoost fail to find strong relationships.

The following two columns shows the Hit Rate metrics, which are adapted classification metrics to evaluate how well models are predicting paying and non-paying users. To do so, the prediction values were transformed in binary values using a small threshold, due high data imbalance as explained in Section 5. The Hit Rate Recall shows how well model are avoiding False Negatives, which means that models rarely assign a non-paying "flag" when user is actually a paying user. On the other hand, models fail to avoid False Positives, which is shown by the low Hit Rate F1 score for all models.

The last 3 columns show how well models predict LTV 180 within paying users. At a first glance, we see that models perform poorly in this task. RMSE and MAE increase significantly comparing to the overall test set. However, R^2 is slightly higher for Lasso and Ridge, which means that those models better explain a paying user than a non-paying user.

Finally, it is easy to see that although RF, XGBoost , AdaBoost and KNN provide error metrics that are in line (but still high) with other models, their predictions are based in an uncorrelation between input and output variables, according to R^2. Lasso, Ridge and MLP are the Top 3 models in almost all metrics, but Lasso is slightly better in all paying metrics. For this reason Lasso is selected as the best model in this works' analysis. Lasso's fine tuned parameters and coefficients are shown in Tables 16 and 17.

Table 16 – Lasso parameters

Parameter	Value
alpha	0.05
copy_X	True
fit_intercept	True
max_iter	1000
normalize	False
positive	False
precompute	False
random_state	None
selection	cyclic
tol	0.0001
warm_start	False

Table 17 – Lasso coefficients

Feature	β_i Coefficients
net_revenue_acc_ds7	116.702114
platform_iOS	3.533894
country_US	2.559655
source_channel_10	1.237592
source_channel_4	0.768841
source_channel_1	-1.013580
country_IN	-1.895729

Apart from simplicity and better performance, Lasso also provides easy and understandable linear coefficients that show how relevant each feature is for the prediction task, which will be analyzed in Section 8.2.

Part IV - Results

8 Results

8.1 Feature Insights

After conducting the data analysis presented in Section 5, several observations regarding the dataset can be done. Initially, characteristics regarding the general distribution of data points and the inter-relationships between features were mapped. Then, the parallel of those data points with our variables of interest, User Lifetime Value (LTV), was determined, completing the observations.

For instance, regarding the players' distribution across countries, the country with the majority of the user base is India. However, India does not show a significant user in-game expenditure, indicating that the revenue per customer is quite low in that country. An alternative in this case would be to increase average user spending by providing a different pricing strategy in countries that show this behaviour, providing even lower prices and more promotions than usual. This same pattern is also noted in Brazil, Russia and Mexico. On the other hand, the United States is a country that has a significant part of the user base and is present in the Top 20 customer average LTV ranking, showing the country's potential for future marketing strategies and expansion efforts.

Regarding the software platform used to play the game, the vast majority of players is concentrated on Android, which is quite intuitive given the worldwide distribution of customers in this platform. However, when looking at users' LTV across platforms, iOS users have a clear advantage in that aspect, showing higher LTV for all time periods analyzed. This demonstrates that iOS players are quite significant to Wildlife, considering their average spending. Therefore, actions can be taken to add in-game features that are specific for iOS users and that focus on maintaining their relationship with the game, such as exploring specific software tool that are not present on Android devices. Similarly, over 70% of iOS users play the game on a mobile phone, which also demonstrates the care that Wildlife must have in optimizing the game for that specific device and to try to increase their share on iPad gamers.

Also, when it comes to the game download sources, the bulk of downloads from the analyzed dataset come organically, which means that they usually do not download the game from paid sources. However, with the presented correlation matrix on Section 5, it can be seen the source-related features have a very low correlation with customers' LTV.

The second part of the EDA consisted of examining the Purchase-related features, which consisted of the number of purchases, net revenue and net revenue per purchase. Interestingly, it was noticed that users made more purchases on the initial days after installation, indicating a higher interaction of users when they start playing the game. Also, when contrasting the average number of purchases per day with the LTVs, it was perceived that the correlation decreases as the purchasing day goes far from the installation day, which states the importance of increasing players' engagement in the first couple of days. Wildlife can implement that by deploying tutorials on how to make store purchases, providing bundle promotions or offering special discounts right after installation. Similarly, the company could put videos with top players from the game showing the importance of store items to better perform in battles, for example.

After mapping the number of purchases, the next step was to understand the behaviour of customers' net revenue during the first seven days since installation. Different from the previous analysis, the average revenue per day does not have a clear pattern in distribution, changing a lot from day to day, which is counter intuitive, since one would imagine that players would become more comfortable with in-game spending as time passes. Furthermore, when comparing the accumulated net revenue with its daily counterpart for the first week of playing, the first presented a stronger correlation with the LTV_180. This indicates that users could be mapped daily with the objective of increasing its accumulated expenditure over a week, and not only daily. Also, the highest correlated days with respect to the net revenue are days 2 and 4, which means that user spending should be highly encouraged on day 2 to push the response two days after.

Another important feature was the average ticket, given by the net revenue divided by the number of purchases. The correlation map indicated in Section 5 shows that this feature presented a very high correlation with LTVs, surpassing the number of purchases and net revenue in some cases. Thus, Wildlife could stimulate an increase in average ticket by displaying higher price in-game products with more relevance, for example.

Finally, the third part of the exploratory data analysis focused on the Event-related features, which are in-game actions and attributes that the users can interact with. In general, these attributes had an extremely low correlation with our variable of interest, LTV_180, and therefore did not have a significant contribution in our regression model. This is a consequence of the fact that event related to tournaments, buying and upgrading cards, piggy bank promotions, etc. usually do not happen before 7 days of user-activity. Nevertheless, some interesting observations can be cited. For instance, the number of users taking the six different parts of the game Tutorial does not change significantly, indicating that there is no urgent need to make the tutorial more appealing to maintain its consistency. Lastly, the degree of interaction with the game shop, despite its low correlation with LTV, is more significant in the first couple of days since installation, indicating the slight potential of stimulating and teaching users how to better use this feature.

8.2 Features Importance Analysis

Random Forests are often used for feature selection in a data science workflow. The reason is because the tree-based strategy used by random forests naturally ranks features by how well they improve the purity of a node. The following analysis shows how the best Random Forest Regressor classified features according to their importance on the prediction task.

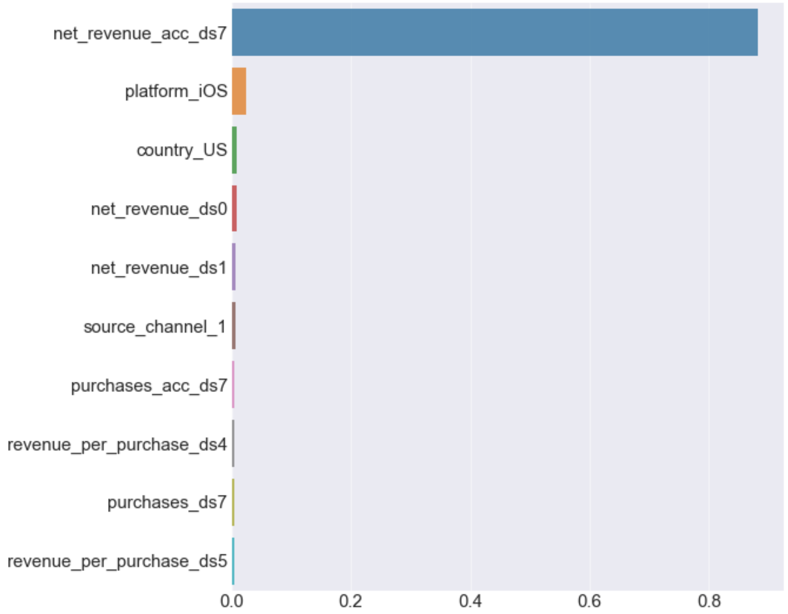


Figure 52 – Features importance- top 10

As expected, the feature importance ranking has a lot of purchase and net revenue-related features, with accumulated net revenue until day 7 (LTV 7) being the most important one. However, interestingly, having an iOS operating system and being from US are the two following most relevant features for LTV prediction according to this analysis using the Random Forest. In order to better visualize relative importance between the other top features, 53 zooms in on the following features after net_revenue_acc_ds7.

Comparing those results to the Table 17, we see that, excluding the top 3, Lasso and Random Forest disagree on which are the real important features on LTV 180 prediction. This result explains the poor performance of RF on R^2 against Lasso, since this metric shows how input explains output variables. However this analysis also shows that purchase-related features are too correlated to be included as separate features, which indicates that a lower maximum correlation threshold could be set on the last feature selection step (see Section 6.5).

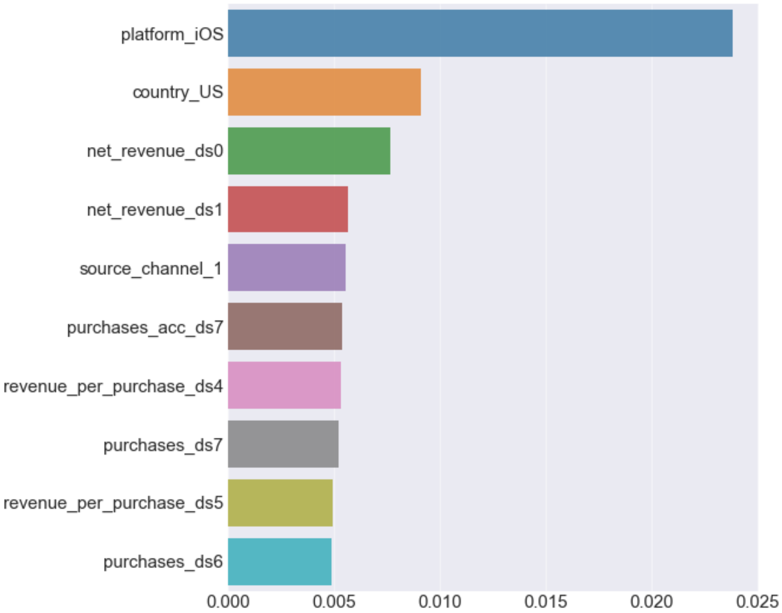


Figure 53 – Features importance zoom in

9 Conclusions

The exploratory analysis of Castle Crush's data revealed several interesting insights regarding the Lifetime Value of players. Firstly, the features that showed the highest correlation with respect to LTV are related to in-game purchases and expenditure. Other created features, such as accumulated net revenue and average ticket per purchase also represented a positive correlation with our variable of interest, demonstrating its relevance to an analysis of this kind.

Furthermore, event-related features showed really low correlation with LTV, which is in line with the domain expert expectations. This is because, usually, event-related features start to be relevant after several weeks or months after download day, when players already understand the game, interacted with the shop, upgraded their cards and, maybe, started a tournament. For this reason, these features are more useful to evaluate game engagement and new features performance over time and not for LTV prediction with a short period of time after installation.

Regarding the LTV prediction models applied to the problem, Lasso was selected as the best choice for the LTV prediction problem for two main reasons. First, it showed the best error rates, specially when looking at paying users, which is quite an important factor. Second, the Lasso regressor also provides a very understandable interpretation of its linear coefficients and it is way simpler than other models tested, such as XGBoost and MLP.

It is interesting to point out, though, that all the regressor models tested still showed higher error rates than expected. Also, the models perform better when taking into account all the users in the dataset, showing higher error rates when taking into account only paying users. These results can be partially justified by the fact that the dataset used corresponds to real world data, which includes lots of discrepancies and irregularities. As a next step to minimize those problems, a even more robust data preprocessing, feature engineering and balancing can be done to try to improve variables correlation with the target variable and model's performance. As an example, time-related features like inter-session time, average time between battles, total playtime, etc., were avoided in this work due to its extraction complexity from Wildlife's database, but they could be a good choice in the feature engineering step, according to literature.

Finally, there are a few things to keep in mind when using the impurity based ranking applied in the Feature Importance step of this work. Firstly, feature selection based on impurity reduction is biased towards preferring variables with more categories. Secondly, when the dataset has two (or more) correlated features, then, from the point of view of the model, any of these correlated features can be used as the predictor, with no concrete preference of one over the others. But, once one of them is used, the importance of others is significantly reduced since, effectively, the impurity they can remove is already eliminated by the first feature. As a consequence, they will have a lower reported importance. This is not an issue when we want to use feature selection to reduce overfitting, since it makes sense to remove features that are mostly duplicated by other features. But when interpreting the data, it can lead to the incorrect conclusion that one of the variables is a strong predictor while the others in the same group are unimportant, while actually they are very close in terms of their relationship with the response variable.

Bibliography

AGARWAL, C.; SHARMA, A. Image understanding using decision tree based machine learning. In: IEEE. *ICIMU 2011: Proceedings of the 5th international Conference on Information Technology & Multimedia.* [S.l.], 2011. p. 1–8. Citado na página 35.

BEHRENS, J. T. Principles and procedures of exploratory data analysis. *Psychological Methods*, American Psychological Association, v. 2, n. 2, p. 131, 1997. Citado na página 46.

BERGER, P. D.; NASR, N. I. Customer lifetime value: Marketing models and applications. *Journal of interactive marketing*, Wiley Online Library, v. 12, n. 1, p. 17–30, 1998. Citado na página 29.

BURELLI, P. Predicting customer lifetime value in free-to-play games. *Data Analytics Applications in Gaming and Entertainment*, CRC Press, p. 11–79, 2019. Citado 3 vezes nas páginas 29, 30, and 38.

CANNON, H. M.; CANNON, J. N.; SCHWAIGER, M. Incorporating customer lifetime value into marketing simulation games. *Simulation & Gaming*, SAGE Publications Sage CA: Los Angeles, CA, v. 41, n. 3, p. 341–359, 2010. Citado na página 26.

CHAMBERLAIN, B. P. et al. Customer lifetime value prediction using embeddings. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* [S.l.: s.n.], 2017. p. 1753–1762. Citado na página 36.

CHEN, P. P. et al. Customer lifetime value in video games using deep learning and parametric models. In: IEEE. *2018 IEEE International Conference on Big Data (Big Data).* [S.l.], 2018. p. 2134–2140. Citado na página 42.

DRACHEN, A. et al. 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.* [S.l.: s.n.], 2018. p. 1–10. Citado 4 vezes nas páginas 36, 40, 41, and 80.

DWYER, F. R. Customer lifetime valuation to support marketing decision making. *Journal of Direct Marketing*, Wiley Online Library, v. 11, n. 4, p. 6–13, 1997. Citado na página 30.

FADER, P. S.; HARDIE, B. G.; LEE, K. L. "counting your customers" the easy way: An alternative to the pareto/nbd model. *Marketing science*, INFORMS, v. 24, n. 2, p. 275–284, 2005. Citado na página 33.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, ACM New York, NY, USA, v. 39, n. 11, p. 27–34, 1996. Citado 2 vezes nas páginas 45 and 47.

FEIJOO, C. et al. Mobile gaming: Industry challenges and policy implications. *Telecommunications Policy*, Elsevier, v. 36, n. 3, p. 212–221, 2012. Citado na página 23.

GLADY, N.; BAESENS, B.; CROUX, C. A modified pareto/nbd approach for predicting customer lifetime value. *Expert Systems with Applications*, Elsevier, v. 36, n. 2, p. 2062–2071, 2009. Citado na página 32.

GONZÁLEZ-PIÑERO, M. Redefining the value chain of the video games industry. *memory*, v. 36, n. 4, p. 75–90, 2017. Citado na página 25.

GUPTA, S. et al. Modeling customer lifetime value. *Journal of service research*, Sage Publications Sage CA: Thousand Oaks, CA, v. 9, n. 2, p. 139–155, 2006. Citado na página 31.

HANNER, N.; ZARNEKOW, R. Purchasing behavior in free to play games: Concepts and empirical validation. In: IEEE. *2015 48th Hawaii International Conference on System Sciences*. [S.l.], 2015. p. 3326–3335. Citado 2 vezes nas páginas 25 and 26.

HUGHES, A. M. *Strategic database marketing: the masterplan for starting and managing a profitable, customer-based marketing program*. [S.l.]: McGraw-Hill New York, NY, 2000. v. 12. Citado na página 30.

JASEK, P. et al. Comparative analysis of selected probabilistic customer lifetime value models in online shopping. *Journal of Business Economics and Management*, v. 20, n. 3, p. 398–423, 2019. Citado na página 33.

KELLY, C.; MISHRA, B.; JEQUINTO, J. The pulse of gaming: gaming disruption. *Accenture report*, Accenture, 2014. Citado na página 27.

MARCHAND, A.; HENNIG-THURAU, T. Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities. *Journal of Interactive Marketing*, Elsevier, v. 27, n. 3, p. 141–157, 2013. Citado na página 25.

MONEREO, I. Insights for evaluating lifetime value for game developers. *Google Play Developer Communications*, 2005. Citado 2 vezes nas páginas 27 and 29.

SCHMITTLEIN, D. C.; MORRISON, D. G.; COLOMBO, R. Counting your customers: Who-are they and what will they do next? *Management science*, INFORMS, v. 33, n. 1, p. 1–24, 1987. Citado na página 31.

SHIH, Y.-Y.; LIU, C.-Y. A method for customer lifetime value ranking—combining the analytic hierarchy process and clustering analysis. *Journal of Database Marketing & Customer Strategy Management*, Springer, v. 11, n. 2, p. 159–172, 2003. Citado na página 30.

SIFA, R. et al. Predicting purchase decisions in mobile free-to-play games. In: *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*. [S.l.: s.n.], 2015. Citado 6 vezes nas páginas 24, 25, 27, 40, 41, and 80.

SIFA, R. et al. 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*. [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 43 and 80.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 38.

TSAI, C.-F. et al. A comparative study of hybrid machine learning techniques for customer lifetime value prediction. *Kybernetes*, Emerald Group Publishing Limited, 2013. Citado 2 vezes nas páginas 38 and 39.

TUKEY, J. W. *Exploratory data analysis*. [S.l.]: Reading, MA, 1977. v. 2. Citado na página 46.

VAFEIADIS, T. et al. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, Elsevier, v. 55, p. 1–9, 2015. Citado 2 vezes nas páginas 36 and 37.

VOIGT, S.; HINZ, O. Making digital freemium business models a success: Predicting customers' lifetime value via initial purchase information. *Business & Information Systems Engineering*, Springer, v. 58, n. 2, p. 107–118, 2016. Citado 2 vezes nas páginas 25 and 26.