UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Mauri Miettinen

ProGame - Event-Based Machine Learning Approach for in-Game Marketing

Master's Thesis
Degree Programme in Computer Science and Engineering
January 2019

Miettinen M. (2019) ProGame: Event-Based Machine Learning Approach for in-Game Marketing. University of Oulu, Degree Programme in Computer Science and Engineering, 63 p.

ABSTRACT

There's been a significant growth in the gaming industry, which has lead to an increased number of collected player and usage data, including game events, player interactions, the connections between players and individual preferences. Such big data has many use cases such as the identification of gaming bottlenecks, detection and prediction of anomalies and suspicious usage patterns for security, and real time offer specification via fine-grained user profiling based on their interest profiles. Offering personalized offer timing could reduce product cannibalization, and ethical methods increase the trust of customers. The goal of this thesis is to predict the value and time of the next in-game purchase in a mobile game. Using data aggregation, event-based purchase data, daily in-game behaviour metrics and session data are combined into a single data table, from which samples of 50 000 data points are taken. The features are analyzed for linear correlation with the labels, and their combinations are used as input for three machine learning algorithms: Random Forest, Support Vector Machine and Multi-Layer Perceptron. Both purchase value and purchase time are correlated with features related to previous purchase behaviour. Multi-Layer Perceptron showed the lowest error in predicting both labels, showing an improvement of 22,0% for value in USD and 20,7% for days until purchase compared to a trivial baseline predictor. For ethical customer behaviour prediction, sharing of research knowledge and customer involvement in the data analysis process is suggested to build awareness.

Keywords: Data analysis, machine learning, market analytics, video games.

Miettinen M. (2019) Progame: tapahtumapohjainen koneoppimisjärjestelmä pelinsisäiseen markkinointiin. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 63 s.

TIIVISTELMÄ

Peliteollisuuden kasvu on johtanut kerättävän pelaaja- ja käyttödatan määrään nousuun, koostuen mm. pelitapahtumista, interaktiodatasta, pelaajien välisistä yhteyksistä ja henkilökohtaisista mieltymyksistä. Tällaisella massadatalla on monia käyttötarkoituksia kuten tietoliikenteen teknisten rajoitusten tunnistaminen pelikäytössä, käyttäjien tavallisuudesta poikkeavan käytöksen tunnistaminen ja ennustaminen tietoturvatarkoituksiin, sekä reaaliaikainen tarjousten määrittäminen hienovaraisella käyttäjien mieltymysten profiloinnilla. Ostotarjousten henkilökohtaistaminen voi vähentää uusien tuotteiden aiheuttamaa vanhojen tuotteiden myynnin laskua, ja eettiset menetelmät parantavat asiakkaiden luottamusta. Tässä työssä ennustetaan asiakkaan seuraavan pelinsisäisen oston arvoa ja aikaa mobiilipelissä. Tapahtumapohjainen ostodata, päivittäiset pelin sisäiset metriikat ja sessiodata yhdistetään yhdeksi datataulukoksi, josta otetaan kerrallaan 50 000:n datarivin näytteitä. Jokaisen selittävän muuttujan lineaarinen korrelaatio ennustettavan muuttujan kanssa analysoidaan, ja niiden yhdistelmiä käytetään syötteenä kolmelle eri koneoppimismallille: satunnainen metsä (Random Forest), tukivektorikone (Support Vector Machine) ja monikerroksinen perseptroniverkko (Multi-Layer Perceptron). Tutkimuksessa havaittiin, että sekä tulevan oston arvo että ajankohta korreloivat aiemman ostokäyttäytymisen kanssa. Monikerroksisella perseptroniverkolla oli pienin virhe molemmille ennustettaville muuttujille, ja verrattuna triviaaliin vertailuennustimeen, se vähensi virhettä 22,0% arvon ennustamisessa ja 20,7% seuraavaan ostoon jäljellä olevien päivien ennustamisessa. Eettisen asiakkaiden käyttäytymisen ennustamisen varmistamiseksi ja tietoisuuden lisäämiseksi ehdotetaan tutkimustiedon jakamista ja asiakkaan ottamista mukaan analyysin tekemiseen.

Avainsanat: Data-analytiikka, koneoppiminen, markkina-analytiikka, pelit.

TABLE OF CONTENTS

Oulu, 15th June, 2020

Mauri Miettinen

1. INTRODUCTION

There's been a significant growth in the gaming industry. Wijman et al. predicted in 2019 that game industry would generate 152.1 billion USD in profit that year which would be an increase of 9.6% over 2018 [1]. There are a total of 2.5 billion gamers across the world. The ways in which they engage with games are constantly changing, leading to more overall engagement and creation of more segments of game enthusiasts. Although the console gaming segment of gaming industry was predicted to be the fastest growing segment, mobile gaming has been and will likely remain the largest segment of global games market. USA and China are the largest video game markets. Wijman et al. predicted that consumer spending on video games will increase to 196.0 billion USD by 2022.

The number of collected player and usage data has increased. The data collected from video games includes user metrics such as revenue metrics and in-game user behaviour, performance metrics such as frames per second and error rate, and process metrics related to performance in the context of game development itself [2]. Such big data has many use cases, which include the identification of gaming bottlenecks such as unstable network connectivity for mobile games [3], detection and prediction of anomalies and suspicious usage patterns for security [4], and real time offer specification via fine-grained user profiling.

Marketing driven by data has been shown to be significantly better at improving conversion and retention rates than previously used "best practice" approaches due to taking into account behavioural patterns, as demonstrated by Sundsoy et al. already in 2014 [5]. The personalization of marketing via machine intelligence helps customers to get more relevant offers, which effectively reduces the amount of perceived "spam". Sundsoy et al. conducted a large-scale analysis for a mobile network operator in Asia to segment customers for text-based marketing. New metrics were created using metadata and social network analysis to identify the customers most likely to convert into mobile internet users. These metrics consisted of three categories; discretionary income, involving customer spending and behaviour; timing, such as whether the customer has changed handset over last month; and social learning, which involves activity among close social graph neighbors. Using this data, a machine learning prediction model was created and used to select a customer group. Conversion rates for machine learning based approaches was found to be far superior to best marketing practices, as results show that conversion rate increased 13-fold compared to the control group using best practice marketing. In addition, 98% of converted customers renew their internet packages after the campaign in the experimental group, compared to 37% of the customers in the control group.

Valero-Fernandez et al. compared a range of algorithms classifying the purchase repetition of online retail customers in 2017 and hence found machine learning approaches feasible for customer analysis. [6]. The accuracy of the classifiers was analysed with linear regression, Lasso and regression trees. The classification of customers was done in accordance to specific marketing focused behaviors using historical transaction data. The models compared were Logistic Regression, Quadratic Discriminant Analysis, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest and Multi-Layer Perceptron. The classification of untrained customers in the UK data set was found to be better than 80%, while prediction accuracy remained better than 60% when using public data sources such as postcode data and UK Land Registry derived demographic data. Classifiers are hence feasible in correctly predicting the customers with the highest probability of purchasing again in the future. In addition, this shows that internal data can be augmented by public data sources to improve profitability and marketing.

However, privacy and ethical concerns arise from using personal data for behaviour prediction. People might not be completely aware that cross-platform information is used to categorize and label customer segments [7]. In addition, automated prediction tools might limit the ability of customers to self-define their social trajectory [8]. There should ideally be a method to predict purchase behaviour without breaching the right to privacy of a customer, such as analytics involving non-demographic data and focusing on in-game information. In addition, making decisions for the user can be unethical, and hence keeping the predictions non-personal and directional keeps data analysis from limiting user agency.

Prediction of customer behaviour has shown promising results in binary prediction of customer purchase [9], prediction of customer lifetime value [10] and prediction of the number of purchases a customer will make [11]. However, there is very little research on predicting the exact monetary value and date of the next purchase of a customer, aside from clustering them into broader customer segments [12]. The ethical evaluation of marketing campaigns is one of the key motivators of this thesis, and hence this thesis focuses on features derived from in-game purchase and usage behaviour and data available from the device itself.

The goal of this study is to predict the next most probable purchase time and value from event-based data using data fusion and machine learning. These two labels were selected due to their novelty, as purchase probability and churn prediction have been studied extensively. Most research on customer purchase behaviour depends on previous purchase behaviour, so to avoid the cold-start problem more features than previous purchase data is needed. The research questions are as follows:

- What factors in in-game behaviour predict purchase time and value?
- How accurately can one predict purchase behaviour by in game behaviour alone in terms of performance?
- What factors need to be considered for ethical and privacy-preserving data analysis?

The answers to these questions can later be used to improve product sales via prioritization of customers and building the customers' trust by using ethical methods for doing so. Offering personalized offer timing could reduce product cannibalization, and the value multiplier can be lower with improved targeting. Ethically considerate usage of data can help future-proof data analytics solutions to changing paradigms in the perception of individual rights concerning their data.

This thesis proposes a new system called ProGame. A new data table is formed using sixty different features aggregated from a set of several separate data tables. The labels to be predicted are the most likely purchase value and time of next purchase realized as the amount of days until purchase from the current data point. A sample of 50 000 data points is analysed to find the most predictive features by calculating the linear correlation of each feature with the two labels to be predicted. Further samples of data points using these features are used as input for three learning algorithms; Random Forest, Support Vector Machine and an Artificial Neural Network. These algorithms are tested both on the whole feature set and on a more limited set including only those variables with high linear correlation.

The structure of this thesis is as follows: Section 2 is provided as an overview consisting of related work in the field of data analysis, specifically focusing on customer data analytics, game analytics and mobile analytics. Section 3 describes the use case in the context of a mobile game, the specifics of data collected and the pre-processing and mapping of variables into features. Section 4 explains the theoretical background of all algorithms used, starting with learning algorithms used in the study consisting of Random Forests, Support Vector Machines and Artificial Neural Networks. This is then followed by descriptions of the evaluation methods used, outlining RMSE, Linear Regression and One-Way ANOVA. Section 5 focuses on the experimental setup, consisting of description of the program pipeline and outlines of all 3 experimental setups; learning algorithm testing with all feature sets, analysis of feature significance and learning algorithm testing with feature sets obtained from the analysis of feature significance. Section 6 describes the results from all three test scenarios and provides a summary. Section 7 considers some ethical aspects related to data analysis and its application to business intelligence. Section 8 provides discussion related to the findings from previous two sections. Section 9 concludes the study and outlines future prospects and research possibilities.

2. RELATED WORK

Advanced analytics are a collection of techniques and tools to use data to discover new facts or business information for data-driven decision making [13]. These techniques include data management, data preprocessing, data analysis and support for decision making. This requires a large volume of detailed data, which often implies the concept of big data that can be defined by three Vs; volume, data of large size; velocity, high speed and possibly real time delivery of data; and variety, possibly unstructured data from a large number of sources. There are many applications of this kind of analysis, including the prediction of churn in an economic recession using historical data combined with application data. This section focuses on analytics in three domains: customer data analytics, game analytics and mobile analytics.

2.1. Customer data analytics

Customer analytics is the usage, aggregation and application of customer data to make informed decisions for marketing. According to a study by Germann in 2014, the extent of deployment of customer data analysis is positively correlated with increased firm performance in all industries, especially so in the retail industry due to the multitude of customer and transaction data to use for this purpose [14]. In addition, Brynjolfsson demonstrated in 2011 that firms that use data driven decision making show an increased performance in terms of productivity and output of up to 6% higher [15]. The positive effect of the usage of data driven decision making also applies to asset utilization, return on equity and market value.

The RFM model (Recency, Frequency, Monetary) by Hughes can be used to model customer profitability [16]. Recency is measured by time since last transaction, Frequency by purchases per time period, and Monetary by the total currency spent on purchases within a time period. RFM model is criticised for its lack of behavioural modelling, but as Tsai and Chiu demonstrated in 2004, RFM model can be used to analyse the profitability of customer clusters after segmentation via a clustering algorithm [17]. Market segmentation is critical, as it enables good marketing and creating or improving customer relationship management programs. Tsai and Chiu developed a novel market segmentation model that relies on product-specific attributes within the transactional purchase history of the customer. This was done at opposition to the idea that all members of a demographic show similar patterns of behaviour. Purchases are represented as a cumulative sum of items bought during a predefined time period, with a quantity assigned to each purchasable item. A genetic algorithm approach is used to group customers with similar purchase patterns into clusters, in combination with a clustering quality function. As the data can be sparse, the similarity measure outlined in the work incorporates co-purchase association. After selecting cluster centers, RFM can be measured for each cluster separately to select the most appropriate customers. This methodology can be used to provide a practical overview for marketing purposes and help marketers revise marketing strategies.

Cumby et al. showed in 2004 that customer shopping lists can be predicted from point-of-sale purchase data [18]. Prediction systems can be used by retailers to provide personalized interaction to customers as they navigate within their retail store. Such system gives retailers tools to interact individually with each customer, enhancing customer experience and loyalty. Using historical transaction data, a separate classifier involving binary predictors for each product category is learned for each customer to predict what items they will buy in each shopping trip. The task required high recall to cover the most items the customer will buy and high precision to not overload the customer with non-relevant items. The features included the following attributes for each product category in the current and four previous transactions:

1. Replenishment interval, meaning the number of days since purchasing the given product category.
2. Frequency of replenishment interval of the given product category, calculated from a histogram.
3. The interval range of the current purchase.
4. Day of week of the current trip.
5. Time of day of the current trip divided into six four-hour blocks.
6. Month of year of the current trip.
7. Quarter of the year of the current trip.

Additionally, four more features were appended for each transaction in local history:

1. Whether the given category was bought in this transaction.
2. The total amount spent in this transaction.
3. The total number of items bought in this transaction.
4. The total discount received in this transaction.

After identification of the customer, the system suggests a whole shopping list at first and only nearby items as the user traverses the store. The shopping list of the customer can be predicted with high levels of accuracy, precision and recall, although 50% of the bought categories were difficult to predict with a reasonable level of precision. A hybrid classifier combining the top 10 products from customer purchase history and a Perceptron-based classifier achieved the best results. The system can increase revenue by up to 11%.

Fader at al. used iso-value curves for customer base value analysis, combining the RFM model with the paradigm of Customer Lifetime Value [19]. Fader et al. argue that these two have been connected previously in a conceptual level, but a formal model for making specific lifetime value predictions for customers using only RFM inputs has not been presented before. Iso-value curves are used to group together individual customers with similar value predictions but different behavioural histories. Iso-value curves are good for summarizing interaction and trade-offs between RFM and CLV values, and a combination of Pareto/NBD framework to capture transaction flow over time and gamma-gamma sub-model for dollars per transaction is used. Iso-value curves are forward looking and feature previously accepted behavioural assumptions, and in addition to prediction the models is resistant to noise. The ease of the input being in the RFM format is also significant benefit. The model is used to estimate a net present value for large customer groups of an online music site, and it reveals significant non-linear associations that could not be revealed by merely observing data. Customers with higher frequency likely have lower future purchase potentials compared to lower purchase rate customers. The monetary value per transaction appear to be independent of recency and frequency. Although monetary value and frequency are independent, they appear to be correlated due to a strong "regression-to-the-mean" pattern forming an illusion. The zero class of customers that made no purchases during the observation period have significant collective profitability.

A users social media profile can be used to predict a users purchase behaviour, as demonstrated in 2013 by Zhang and Pennachiotti [20]. Users often connect to social media accounts from e-commerce websites, so mapping their correlation was of importance. The typical strategic goal for customer engagement using social media is to provide engaging experiences to increase customer retention and adoption or attract younger users. However, technologies for data collection from social media by e-commerce companies had not been fully developed, although it shows potential for purchase behaviour prediction, product recommendation and fixing the so called "cold start" problem where a user has no previous history within the system. Zhang and Pennachiotti conducted an extensive analysis of how a user's Facebook profile information correlates with their eBay purchases, and compared the performance of different algorithms and feature sets to outline a system that uses the user's social media profile to predict user purchase behaviour. A high correlation between purchases and the categories of pages the user liked on Facebook was found, and all categories of products had at least one highly associated Facebook category. There were four types of feature sets used: demographic features (D) consisting of age and gender, Facebook categories (F) representing the amount of likes of the user per page meta-category; user's likes at page level (L); and Facebook n-grams (N) derived from the names of the liked pages. Feature set D showed the smallest improvement over time but was nevertheless found to be valuable in predicting purchases. Feature set F performed the best regardless of having less fine-grained information, while similar performance was found for L and N. This indicates that N and L convey the same information as categories, which is logical as they are generated from the same list. Feature set F provided the best generalization power over all users while having a low computational cost, while L and N were most likely too sparse which reduced their prediction power. The machine learning task was to predict the most likely product category an e-commerce website user will buy in a cold-start situation. For algorithms, Logistic Regression and Support Vector Machine outperformed the baseline system at a significant level.

Gupta and Pathak showed in 2014 that data analysis can be used to predict customer purchase based on the dynamic pricing of a product on an e-commerce platform [12]. Dynamic pricing is beneficial for online retailers as it increases sales and margins. Gupta and Pathak aim to create a generic framework to enhance right price purchase instead of the cheapest purchase in an e-commerce platform. The data sources for the framework include visit attributes, visitor attributes, purchase history, web data and context understanding, but had to be processed as they were not in a continuous form. The derived variables from this data used as input features were purchase by category (PCT), purchase by quantity (PQT), purchase by company (PCY), purchase by brand (PBD) and purchase by channel (PCN), alongside purchase amount and quantity. Instead of individual buyers the customers are segmented into clusters, and the price range is defined for each cluster using linear regression. In this way the appropriate price range can be selected by finding their corresponding customer class. Finally, given the appropriate customer segment and price range, a binary Logistic Regression system is used to predict if a customer is likely to purchase a given product or not. Using this price prediction framework improved revenue generation with lower prediction errors when compared to the same product offered at a fixed price. The study focused on the context of an inventory-based e-commerce platform but can be applicable to non-inventory based online marketplaces.

Machine learning algorithms are a useful tool for predicting the expected lifetime value of a customer, as Chamberlain et. al demonstrated by using customer embeddings to do so [10]. Predicting Customer Lifetime Value is important for effectively allocating resources for marketing spending, identification of high value customers and mitigating losses. Clamberlain et al. analysed this kind of system comprising a random forest model trained on a set of 132 handcrafted features. Random forest was selected as it performed better on a situation in which CLTV is mostly zero with non-zero cases differing by orders of magnitude, and during experimentation with feature importance the most important feature category was previous purchase history (0.600) followed by app session logs (0.345). The top features by order of importance were:

- Number of orders
- Standard deviation of order dates
- Number of sessions in the last quarter
- Country
- Number of items in the new collection
- Number of items kept
- Net sales
- Days between the first and last sessions
- Number of sessions
- Customer tenure
- Total number of ordered items

- Days since last order
- Days since last session
- Standard deviation of the session dates
- Orders in the last quarter
- Age
- Average date of orders
- Total value of orders
- Number of viewed products
- Days since last order in the previous year
- Average session date
- Number of sessions in the previous quarter

Customer is defined as churned if they do not place an order in a year. Using this previous system as a baseline as defining CLTV as the net of orders placed within a year, two candidate architectures for hybrid systems based on both handcrafted and learned features were proposed. The first model was a feed-forward neural model trained on handcrafted features in a supervised setting. The second model used unsupervised learning to generate a customer level embedding directly from session data consisting of sequences of products viewed by the customer, which then were used to augment random forest features. The usage of these embeddings to predict customer churn showed a significant improvement over a benchmark classifier consisting of just large numbers of handcrafted features.

Aside for customer lifetime value, behaviour such as binary prediction of whether the user will purchase can be valuable, a tool for which was outlined by Martinez et al. in 2018. [9]. Predicting the purchase behaviour of customers supports planning the warehouse and point of sale inventory, defining manufacturing strategy and efficiently directing resources. Martinez et al. developed an advanced analytics tool for predicting whether a customer is going to purchase a product within a given time frame in the future in a non-contractual setting. They proposed a new set of relevant features for customers via times and values of previous purchases divided by month, values of which were updated every month. The features were:

- Number of total purchases
- Mean time between purchases
- Standard deviation of purchase frequency
- Maximal time without purchase
- Time since last purchase

- Thresholds for classification for the number of time units between time purchases based on the mean and standard deviation of time units between purchases, 3 in total
- Classification of purchase frequency in terms of customer risk in terms of "normal", "attrition", "at-risk", and "lost", based on the previous thresholds.
- Moving averages of order 6 and 3 of binned purchase values and their polynomial approximation
- Maximum values of purchase over actual purchases and polynomial fit
- Mean values of purchase for the actual, binned and fitted values
- Median values of purchase for the actual, binned and fitted values
- Time frame variations as the relative change in purchase values as a number value and a categorical variable
- Purchase trend as the relative change of fitted purchase values
- Country of the customer

For all non-categorical variables pairwise products, powers of three and two, and logarithms were added, making a total of 274 features. Three algorithms were tested: logistic Lasso regression, extreme learning machine and gradient tree boosting. Gradient tree boosting method was found to perform the best, and a data set of 200000 purchases and over 10000 corresponding customers were analysed. An accuracy of 89% and an AUC value of 0.95 was obtained in the prediction of purchase in the next month.

2.2. Game analytics

Game analytics is the application of business analysis to the context of games, concerned with all forms of data related to game business and research [2]. Game analytics are valuable due to the highly competitive nature of game industry, as thousands of games compete from the players' time and attention. It provides support for decision making in all levels of game development including but not limited to design, marketing and user research, and it's directed to both games as a product (user experience) and a project (development). Business intelligence in general gains data from various sources including but not limited to market sources such as benchmark reports, white papers and business reports; company sources such as QA reports, production updates, budgets and business plans; and sources such as test reports, research and customer support analysis.

Medler et al. developed a visual game analytic tool called Data Cracker in 2011 for the analysis of online gameplay behaviour alongside the development of Dead Space 2 [21]. Similar game analytic tools help designers to support design intuitions with data, which increases the speed of iterations in the development cycle. Data Cracker enabled the access of player data to the whole team instead of a small subset of "superusers", which was intended to increase the data literacy of the team and hence have an increased interest in data analysis. Building the tool during the early game development phase of Dead Space 2 increased the interest of team members in the tool. With this interest, Medler et al. encourage creation of live teams that continue the data analysis process for a period after the release of a game, which was considered an alternative for at the time common practice of disbanding the game team after release.

Similar ideas were explored by Hullett et al. in 2011, where they outlined how data collected from a released game can inform subsequent development especially in regard to resource allocation [22]. Data analysis can improve software productivity, quality, reliability and performance, and Hullett et al. outlined qualitative and quantitative sources for three types of broad categories. Internal testing includes developer testing and quality assurance, while external testing consists of usability testing, beta testing and long-term play data. Subjective data can be collected via surveys, reviews, from online communities or from "post-mortems" of games, which are development summaries of successes and failures published after a game's release. Data collected by such analysis can inform development, such as in the case of resource allocation and unused content. Many game modes, event types and vehicles had low appeal and were hence unnecessary. Knowing such things in advance would reduce costs and development time in asset creation, which is a significant expense.

In-game behaviour can be used to predict purchase decisions, as demonstrated by Sifa et al. in 2015 [11]. Only a small fraction of players make any purchases, and hence predicting who will purchase from a user profile enables optimization and tailoring of marketing efforts. Two models were generated on a 100 000 player data set for the distinction of players: a classification model for binary prediction whether the user will purchase or not, and a regression model for prediction of the number of purchases the user will make. The features used were aggregated from several data tables, and from each session:

- Country
- Device
- Move Count
- Active Opponents
- Logins and game rounds
- Skill level
- Reached goals in the game
- World number
- Number of interactions
- Number of purchases
- Amount spent

- Playtime
- Last inter-session time
- Last inter-login time
- Distribution of inter-login time
- Distribution of inter-session time
- Correlation on time
- Mean and deviation on time
- Country segments

The error was reduced by 13% for a descriptive model compared to a baseline of using the mean purchase amount from 7-day observations. Predictors for future purchase in the classification task included intensive interaction with a game, large amount of total playtime rather than mere session length, and the player's progression within the game. The regression task shows similar results, as the importance of the country of the customer encourages localization and optimization to local markets. The RFM model is argued to be driver in customer purchase due to the good correlation of purchase with total purchases and amount spent.

CLTV analysis has been applied for video games as well, and the early detection of high value players is vital in free-to-play games where up to 50% of the revenue is provided by 2% of the players [23]. As players may stay in the game for years, the result is a rich data set for prediction. Chen et al. found that convolutional neural networks are more efficient for the prediction of CLTV for individual players compared to several parametric models including Pareto/NBD and its extensions. Neural network models were better suited for describing the purchasing behaviour of high-value customers, as their errors values were half as large as the parametric algorithms. Deep Neural Network and Convolutional Neural Network produced similar results, which was largely explained by a high overlap of features between the two. In addition to superior accuracy they scale to big data and have low computational times due to their capability of working with raw sequential data and not requiring feature engineering. They are suitable in identifying so called "whales", which allows game developers to develop methods to retain high-value customers and hence increase revenue.

Baolo Burelli provided an overview of CLTV in different fields and presented challenges that are specific to free-to-play-games, where the lack of pay wall and erratic spending behaviour makes revenue spending difficult [24]. Finding the correct methods is especially important since in free-to-play market the competition in user acquisition is intense. The four main groups of prediction are average-based, Pareto/NBD and derivatives, Markov Chain Models and Supervised Learning Models. Using supervised learning algorithms for CLV prediction is considered an emerging trend in the current industry where Pareto/NBD and average-based methods are dominant. Unlike their methods, classical statistical methods make assumptions about the distributions of input data and do not allow multiple co-variates. The future directions for research suggested were advanced deep learning techniques such as auto-encoders and deep convolutional neural networks, sequential modelling with time series regression and classification, transfer learning for multi-game management and lifelong learning.

2.3. Mobile Analytics

As defined by Zaslavsky in 2013, mobile data analytics comprise of tools to process, analyse and visualise data that originates from mobile devices [25]. There has been a significant increase in research interest in mobile analytics due to the increasing amount of data generated by mobile sources. Mobile devices have turned into a rich source of data due to their server connections and their powerful but limited processing capabilities. These data streams can be valuable for urban modelling, transportation and mobile crowd sensing applicable to citizen journalism. Zaslavsky outlined a method for mobile stream mining to both save bandwidth and energy and address previously observed scalability issues in runtime processing and data collection.

It is possible to predict human behavior using smart phones as sensing devices, as demonstrated by Do and Perez in 2014 when they predicted where users will go and what app they will use in the next ten minutes [26]. This was made possible by the rich contextual information that smart phones can provide. The goal of the study was to define which sensor data types are important for the classification and extracting generic behavioural patterns and study whether they can be used to improve performance of personalized models. The experiment was conducted on the Lausanne Data Collection Campaign data set consisting of longitudinal data from smart phones collected over a period of 17 months from 71 users. The nearby Bluetooth devices, visited named locations and time are important in predicting location, demonstrating the dependency between human mobility and social interaction. The current app, visited named locations and nearby Bluetooth devices are important to predicting the next app used, which show a potential to infer application usage in the future as little work has been done on such a task in the past. Transforming user specific contexts and variables into generic concepts is a plausible approach to combine generic models with a personalized model for making accurate predictions in situations where the amount of single user data is small.

He et al. argued in 2016 that big data analytics can improve performance of mobile cellular networks and maximize the revenue of their operators [27]. He et al. introduced a unified data model and a framework for applying big data analytics in mobile analytics. Examples on signalling data, traffic data, location data, radio data and heterogeneous data were described. The open research challenges included privacy, filtering out non-useful data, automatic generation of correct metadata. Additionally, in the future data should be located, identified, understood and cited automatically to reduce the labor of the data analysis process.

Using mobile analytics, Peltonen et al. demonstrated in 2018 that geographic, demographic and cultural factors affect mobile phone usage [28]. A large-scale analysis was conducted on 25,323 Android users from 44 countries and 55 app categories. The difference in the usage of these app categories reflect geographic boundaries of countries the most, but there are also geographic and socio-economic subgroups. Language has a strong impact, as English-speaking groups use all categories of apps in a more diverse fashion compared to group from non-English speaking countries. Educated professionals form strong clusters across countries, while younger users (under 25 years) are dissimilar between countries. App usage also correlates with cultural values, as countries with collectivist and feminine values show higher usage of family-related applications, while countries with shallow hierarchy or high value placed on individualism prefer leisure-related apps. These findings can be used to improve the targeting and personalization of mobile apps for users across countries.

3. DATA DESCRIPTION

The use case of this study is the collection of data for each customer in a context of a casual mobile free-to-play racing game with a game loop involving several sub-activities. The sub-activities included are playing and finishing races, selecting game modes, participating in events, customization of cars and player characters and in-app purchases. The game is based on physics and involves only two controls; brake, which also rotates the car clockwise when in the air; and gas pedal, which increases speed and rotates the car counter-clockwise when airborne or driving downhill. The game supports both online and offline play with multiple game modes, including but not limited to adventure mode and cups.

The most important sub-activity is playing and finishing races. The player selects a car from several options with different handling characteristics. After selecting a game mode, the player is put into one of available stage types possibly involving procedural generation. Fuel is a limited resource and one can find fuel cans scattered around a stage to refill the fuel meter. Performing tricks such as wheelies and flips earns bonuses, however the player immediately loses the race if the car lands upside down or the driver collides with the scenery in some other way. Finishing a race earns the player coins and other rewards, but the levels or environments of the game presented as different racing "stages" also contain coins the player can collect in addition to this.

The game has several available game modes for racing for which the goal varies. The first main type is an endurance type mode in which the goal is to drive as high of a distance as possible without running out of fuel or damaging the player. The second main type involves defeating AI drivers or other players in fixed-length cups. The last type are events, which are limited time gameplay that provide extra rewards. These challenges include daily races, weekly races, public events and race challenges from other players.

The coins that the player acquires through playing the game can be spent on upgrades for the vehicle in a customization menu. These upgrades take the form of car parts that improve the handling characteristics of the car, including but not limited to engine and suspension improvements and specialized tires. Other customizations are cosmetic and change the appearance of the player via clothing and head options or the paint job of the selected car.

The game is free to play but includes optional in-app purchases. Real money can be used to purchase coins and so called "gems", which are a currency type specific to the in-game store. Customizations and upgrades can be purchased with varying amounts of coins or gems, the latter of which can additionally be used to purchase loot boxes containing random rewards. Using the in-game purchases enables faster acquisition of all in-game resources, reducing the amount of time spent earning them via racing in cups, and disables the showing of ads.

According to Google Analytics, the player demographics are varied. Figure 1 shows that 75,3% of players are male and 24,7% female. The largest age groups per gender are 18-24 years for males (>25% of total players) and 25-34 years for females (10% of total players). Players of age 65 and over make up less than 2% of total players. Figure 2 shows that the largest amount of players (15% of total players) are from the United States, with Russia, India and Germany following in order.

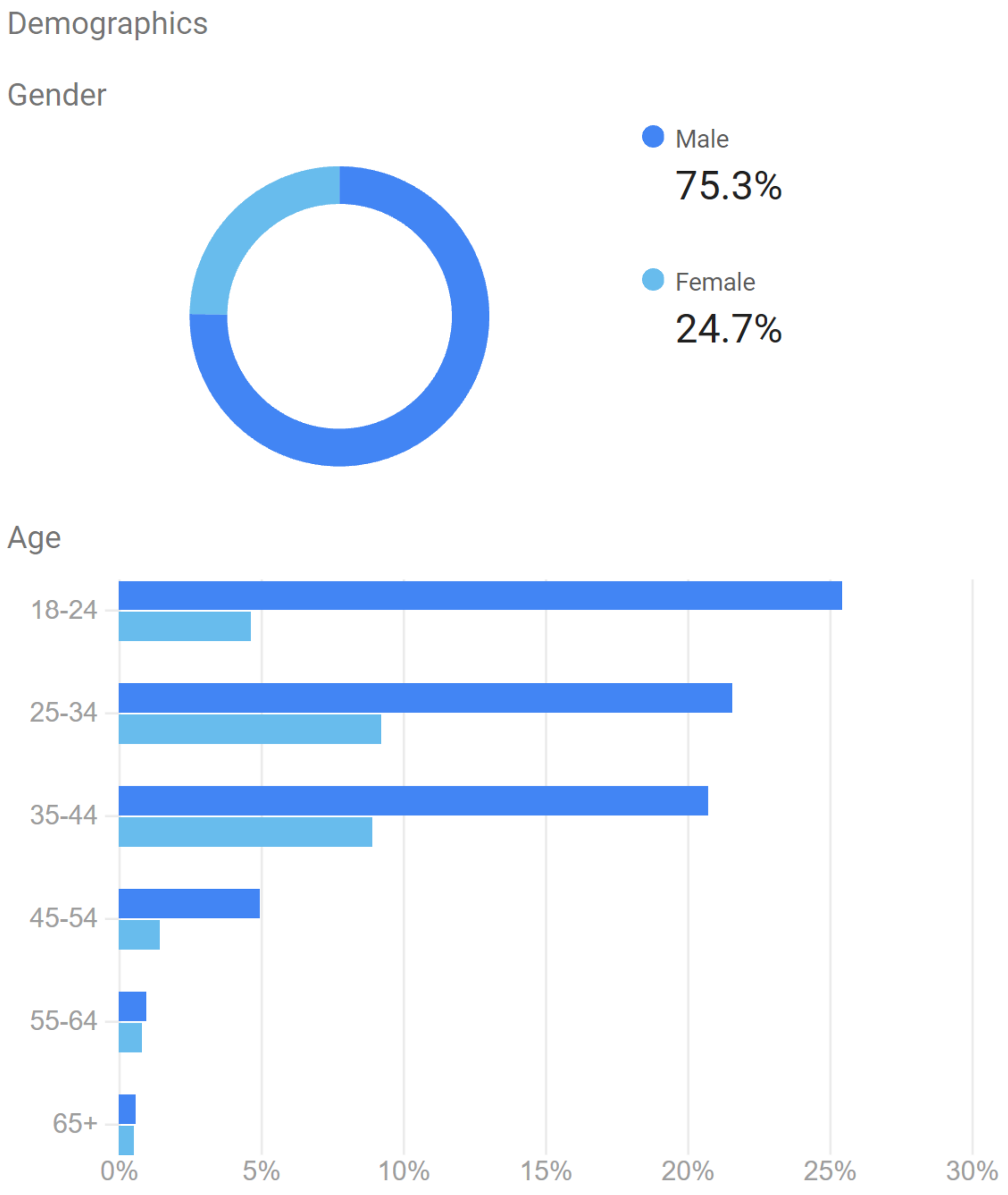

Figure 1. Demographics of the game by gender and age.

3.1. Data collection and data processing

Event data is collected continuously at all points in the game loop and saved into Google Cloud as data tables, which can then be queried with SQL commands using BigQuery[1]. The events are data points with a time stamp and relevant

[1] https://cloud.google.com/bigquery

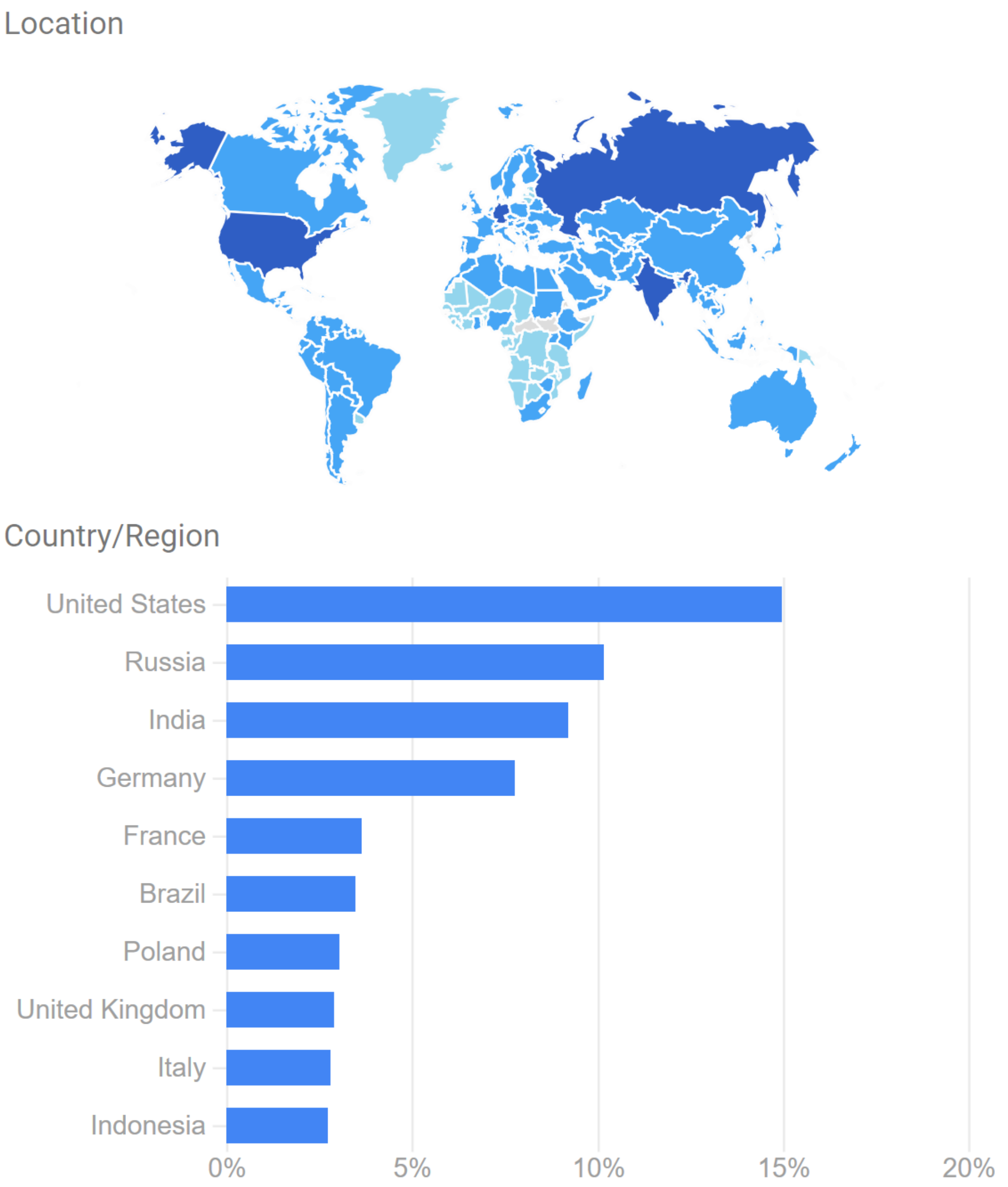

Figure 2. Demographics of the game by country.

attributes and consist of user purchases and in-game behaviour such as finishing games, gaining resources and changing customizations recorded several times a second as they happen. The event stream is collected into a daily raw data table and further aggregated into sub-tables. The attributes used in this study are collected from these pre-materialized sub-tables which include event-based purchase data, fixed time sampled daily recorded user values and event-based game session data.

The terms of use in the form of EULA and privacy policy are given in the download description on Google Play, and the user approves to the processing of data by the mentioned policy by downloading and using their services. In addition to this, during the second day of playing the user is shown a popup regarding EULA. The developer prevents unauthorized access and improper use of personal data with encryption and limited access to data. The data collected is used to ensure function of services, improve player interaction, show personalized advertisements, ensure safety and fairness and in data processing such as analysis, segmentation and profiling. The data is shared to certain service providers, development partners, public authorities, advertising and social media partners. The user's rights consist of opting-out of targeted advertising and access, modification, deletion and access control of the personal data collected from the them.

3.2. Pre-processing and attributes

In order to generate suitable features and labels for the experiment, the data needed to be processed into a single data set and further divided into feature sets to create test cases. The first step was to generate an aggregated data set from multiple sub-tables as shown in Figure 3. Using BigQuery to measure correlation between the time stamps of the last and current purchases, a large correlation was found but as the amount of lifetime purchases per customer was found to be very small (75% of customers have bought 4 items or less in total) and the average time between purchases was found to be over a month. Based on this result, the time scale accuracy was designated to be a single calendar day. Hence, a subset of daily user data was selected as the starting point of data aggregation, as it provides in-game features for each day and some demographic information.

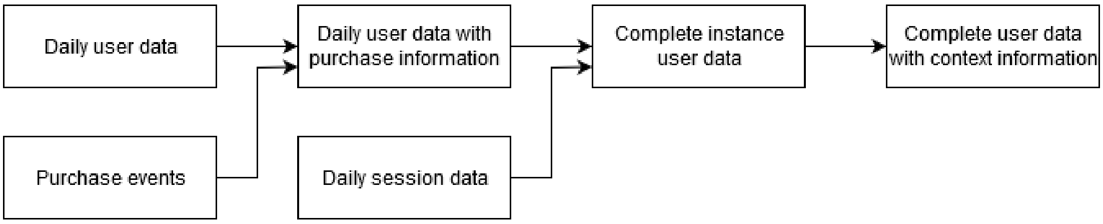
Daily user data
Daily user data with purchase information
Complete instance user data
Complete user data with context information
Purchase events
Daily session data

Figure 3. SQL processing represented as joint subtables.

As some of the aggregated data tables are only updated once a day, the timestamps of all the recorded purchase events and daily session data used for aggregation were processed to represent the calendar date at the time of the event. To combine irregularly sampled purchase event data and daily user values for each user, the lifetime purchases of the user were effectively grouped into sets of two sequential purchase. Let d_i be the date at timestamp i. Two purchases at timestamps t and $t-1$ will be associated with a daily record from timestamp r as the future and last purchase, respectively, when $d_r < d_t \wedge d_i >= d_{t-1}$, forming a data set of daily user data with purchase information. Due to previous usage in research [11], daily session data was added by grouping all sessions by the calendar date and taking the amount of sessions and total sum of their session lengths for each calendar day. These were then added to the combined daily user data to form the complete instance user data, having a data point for each user active at each calendar date. Of note is that only the data points with a known future purchase are selected. The selected variables include among others purchase data due to previous success in their application [17][10][9], and changes in in-game rank and in-game currencies intended to model player performance and purchase of in-game improvements.

The completed instance user data was appended with a subset of its non-static features from the last 14 data points to model sequential behaviour as context data. A sliding window -style approach was used, meaning if the user had less than 14 previous data points in their behaviour log that amount was used. The amount of data points was selected based on internal experience that many in-app purchases are done before the 14th day of being a customer. Event based data can be represented as a cumulative list [17], and aggregation of events into equal sized time windows with the time span included [29] was used to incorporate the total sum of a subset of the frequently changing non-static features from past 14 or less valid data points ("_sum" suffix). The time span was defined as the time in days between the current and first data point in the window. In addition, the maximum ("_max" suffix) and average values ("_avg" suffix) from a set of these 14 data points for each feature were included in the final feature set consisting of complete daily user data with context information.

The aggregated features and labels were divided into feature categories for easier creation and performance analysis of combinations of feature sets. Data is represented such that every calendar day has a single data point for each of the users. The labels to be predicted are the days until next purchase (days_until_purchase) and the value of the future purchase in USD (value_in_usd), which are both of float data type. General static or steadily increasing variables were designated as the "General" feature set, while frequently changing variables and their change compared to previous day were designated as the "Daily" feature set. RFM variables were analysed from lifetime purchase data and designated as the "RFM" feature set. Of note is that day of week, day of month, month and day of year are mapped into categorical variables for training purposes. Finally, the context data from the last fourteen data points was designated as the "14-Point" data set. After processing, there were a total of 16 million data points. The full list of features are shown in Appendices in Table 9, Table 10, Table 11 and Table 12.

4. THEORETICAL BACKGROUND

The following sections will outline the theoretical background of the algorithms used in this experiment. The three algorithms used for model training are Random Forest, Support Vector Machine, and an artificial neural network using Multi-Layer Perceptron. The performance of the three algorithms are evaluated with Root Mean Squared Error. The linear correlation of features and labels are evaluated with Linear Regression and One-Way ANOVA.

4.1. Learning algorithms

Machine learning is the development of methods that can automatically detect patterns in data and use those patterns to predict future data or generate other outcomes [30]. Machine learning tools use approaches and methods from probability theory, and they have been applied for multiple domains of research such as molecular biology, text processing, computer vision and robotics. Machine learning is used for tasks that are too complex to adequately explain and hence program by humans, tasks that are beyond human capabilities such as analysis of very large and complex data sets and tasks that require adaptability to a changing environment [31].

Machine learning can be divided into subsets as a form of taxonomy [31], from which the most relevant type of learning for this study is passive supervised statistical batch learning. Supervised learning concerns situations where the correct labels to be predicted are included in the learnable data and the goal is to predict labels for data where such labels are missing. Unsupervised learning however involves meaningful summarization and compression of data without labels. Batch learning is machine learning in situations which there are large amounts of training data to output conclusions from without time specifications, which is contrasted by online learning in which decisions are made throughout data stream at time intervals. Passive learning involves observation of information provided without modification, in contrast to an active learner which interacts with the environment by posing queries and performing experiments.

The formal model for each of the machine learning algorithms includes input, output, data generation model and measures of success [31]. The input consists of the domain set X described by features, a label set Y which represents the "correct" output for each data point, and training data S as a set of pairs $(x_1, y_1, \ldots, x_n, y_n)$ from $X \times Y$ which is a sample of the domain set that is available for learning for the algorithm. The training set is created via a data generation model, meaning each data point x_i is sampled according to the distribution of the domain set D and labeled by f, which is the assumed "correct" labeling function. After training the learner outputs a prediction rule $h : X \to Y$. Finally, the measure of success is the error or loss of the classifier which can be defined as the probability that the classifier does not predict the correct label, or $h(x_i) \neq f(x_i)$. Machine learning algorithms learn by empirical risk minimization, which is minimizing the error over each sample by pre-selecting a set of classifiers h_i from a hypothesis class H and outputting the classifier that has the lowest error in the training sample, also known as training error or training loss $L_S(h)$. Overfitting is when the analysis or model corresponds too closely to a set of data and may for that reason fail to reliably predict future data [32].

4.1.1. Random Forests

Random forests are classifiers or predictors consisting of ensembles of "trees", meaning tree structured classifiers or predictors [33]. Each tree is grown using a random vector sampled independently from the same distribution, the dimensionality and nature of which depends on how it is used in the construction of the trees. The generalization error converges asymptotically to a limit when the tree increases in size, and it depends on the strength of the individual trees and the correlation between them. Random forests do not overfit as more trees are added. When the decision to split the node is done on a random selection of features, compared to AdaBoost [34] it provides increased resistance to noise while offering comparable performance. The algorithm is insensitive to the number of features used for each split, and in most cases selecting one or two features gives nearly optimal results. Also, random forest does not change the training set like adaptive bagging and arcing, which reduces bias. Using random features and inputs produces good results for classification, while regression shows lower performance. One additional benefit is that Random Forest provides an estimate of variable importance as internal estimates which show response to increasing feature numbers. Out-of-bag estimates are used to measure generalization error, strength and correlation.

Following the definitions of Breiman [33], a typical tree classifier or predictor $h(\boldsymbol{x}, \Theta_k)$ is grown influenced by values of an independently sampled random vector Θ_k from the same distribution, and each tree as output casts a unit vote on the most popular class for input $\boldsymbol{x}$, or in the case of regression a predicted value. In Random Forest, bagging is used with random feature selection, meaning a bootstrap subset T_k of training set T is taken and a classifier $h(\boldsymbol{x}, T_k)$ trained using random feature selection. The so called out-of-bag estimate is used to estimate generalization error: For each $\boldsymbol{x}, Y$ in the training set aggregate only the votes of out-of-bag classifiers, meaning those classifiers in which T_k does not include $\boldsymbol{x}, Y$. Out-of-bag estimation is also used for strength, correlation and variable importance. Each individual tree in a random forest consists of a sequence of decision nodes. The tree is grown as follows [35]:

1. Sample a set of N cases from training data with replacement to be used as the training set for the individual tree.

2. For M input variables select $m << M$ variables for candidate split points. For each of these m, produce a split and select from these variables based on their respective performance according to a split criterion such as Gini impurity or information gain, metrics typical for decision tree training.

3. Grow each tree as large as possible without pruning, stopping when a split no longer provides improvement or all samples in the node have the same class/value.

This study uses the Scikit-Learn implementation, in which the probabilistic predictions of the trees are averaged instead of individual voting, and Mean Squared Error used as the default split criterion[1].

Random Forest can still be modified further for less perturbations to the model but at a slight cost to "optimality" by using Extremely Randomized Trees for regression and classification [36]. They involve randomizing the choice of both the set of attributes and their cut points fully independent of the target variable for the splitting of a tree node. The model can be tuned for the specifics of a given problem via three parameters: main parameter K, the strength of randomization; secondary parameter n_{min}, degree of smoothing; and M, number of trees generated. In extreme case, Extra-Trees build completely randomized trees by picking a single attribute and its cut point at random regardless of the value of the target variable at each node. The default choice for this parameter was evaluated for robustness and what values were optimal in different situations. Extra-Trees algorithm was found to be computationally efficient and accurate. A bias-variance analysis showed that the algorithm decreases variance while increasing bias.

Random forests are the most popular ensemble method due to it's desirable properties such as a built-in measure of importance, out-of-bag error and proximities [37]. This is in addition to the previously mentioned resistance to outliers and noise and ease of parallelization. Since their inception by Breiman in 2001 [33], they have been used in various fields of study such as medicine, agriculture and astronomy. For example, Random Forests have been found effective for creating network intrusion detection systems [38]. The main purpose of an IDS system is to use network traffic data to classify user activity as normal or anomalous. This problem is non-linear and complex, but a Random Forest based model trained on NSL-KDD data set using Symmetrical Uncertainty (SU) as pre-processing feature selection shows high efficiency. The pre-processing also included replacing missing values with the mean and mode from training data and the discretization of numbers with unsupervised 10 bin discretization. The model has a high detection rate and a low false alarm rate, reaching an accuracy of 99.67% for detecting all four types of attacks in the data set including DoS attacks, Probe, R2L and U2R.

4.1.2. Support Vector Machines

Support vector machines, also called support vector networks, are a type of learning algorithm based on linear or non-linear mapping of features to a higher dimension feature plane [39]. Linear decision planes are then generated on these mappings using training data for reference to separate the

[1] RandomForestRegressor: `https://scikit-learn.org/stable/modules/ensemble.html#id5`

samples for clustering and regression. The decision surface has special properties that enable them to be very generalizable, and can also be used for non-separable data.

There is a library for using Support Vector Machines called LIBSVM [40]. One of the algorithms supported by the library is called the Epsilon-Support Vector Regression, which is based on the 1998 work Statistical Learning Theory by Vapnik [41] and will be the focus of this research. Let there be a set of training points $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ where $\mathbf{x}_i \in \mathbb{R}^n$ is a feature vector and $y_i \in \mathbb{R}^1$ the target output or the label. The dual problem to be solved is the following

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T Q (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} z_i (\alpha_i + \alpha_i^*)$$

subject to

$$\mathbf{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0$$

$$0 \leq \alpha, \alpha^* \leq C, i = 1, \ldots, l$$

where $Q_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ is the kernel function, $\phi(\mathbf{x}_i)$ maps input $\mathbf{x}_i$ into a higher dimension, ϵ is the main parameter, $\mathbf{e}$ is a vector of all ones, $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$ are parameter vectors for which values are below the regularization parameter C. After solving the dual problem, the approximation of the decision function is

$$\sum_{i=1}^{l} (-\alpha + \alpha^*) K(\mathbf{x}_i, \mathbf{x}) + b$$

Scikit-learn implementation uses rbf function as the kernel[2].

Support vector machines show improvement over k-nearest neighbour and random forest models for land cover classification from imagery [42]. Random Forests, k-Nearest Neighbor and Support Vector Machines are reported to produce high accuracy models. Land use and cover classification data set of a 30 x 30 km^2 area in the Red River Delta in Vietnam from a Sentinel-2 Multispectral Imager was used in the study. The overall accuracy of all algorithms was between 90% and 95%, however SVM had the highest followed by Random Forest, while kNN had the lowest. If the training sample was large enough and encompassed approximately 0.25% of the total area all classifiers showed a similar and high overall accuracy regardless of the extent of balance and imbalance in the sets.

4.1.3. Artificial Neural Networks

Neural networks construct complex internal representations from tasks using neuron-like processing units [43]. Connectionist learning procedures such as neural networks resemble the functioning of brains more closely than conventional computers as they consist of a system of interconnected layered units, involving at least input units and output units. Due to this structure they can utilize parallel computation well. The units of this network interact using weighted connections, and the location and weight of these connections determine the

[2] SVR: `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html`

long-term knowledge in this network, meaning learning involves changing the connections or their weights. Every unit or neuron involves a "state" or "activity level" determined by weighted linear combination of inputs from other neurons in the system. The total input x_j of the neuron j is the linear combination of the activity levels of all neurons providing input to the neuron:

$$x_j = \sum_t y_t w_{ji} - \theta_j$$

where y_j is the state of unit i, w_{ji} is the weight of the connection between i and j, and θ_j is the threshold of neuron j. The state of the neuron is a non-linear function of this input and defined as the activation function. A typical function for continuous units is the logistic function defined as $y_j = \frac{1}{1+e^{-x_j}}$.

Supervised learning procedures involve training a network for a pre-defined ”desired” output. The measure on how the network is doing compared to these is defined as

$$E = \frac{1}{2} \sum_{j,c} (y_{j,c} - d_{j,c})^2$$

where $y_{j,c}$ is the state of output unit j in input-output case c while $d_{j,c}$ is the desired state. The error can be minimized by iteratively changing each weight w by

$$\Delta w_{ji} = -\epsilon \frac{\delta E}{\delta w_{ji}}, \quad \frac{\delta E}{\delta w_{ji}} = \sum_{cases} (y_j - d_j) \frac{dy_i}{dx_j} y_i$$

where ϵ is a configurable value tending to zero.

In networks with hidden units between the input and output units, the weight values are updated via back-propagation. In back-propagation, it is possible to calculate $\delta E / \delta w_{ji}$ for all nodes with modifiable income weights given $\delta E / \delta y_j$. The procedure involves two steps, the first of which involves calculation of the activity levels of every unit in the system starting from the input nodes in the "forward pass". Afterwards, $\delta E / \delta y_j$ is calculated for all hidden units starting from the output units in the "backward pass". Given a hidden unit j in layer J, the effects of which are seen on the units k in the next layer K, $\delta E / \delta y_j$ is calculated as

$$\frac{\delta E}{\delta y_j} = \sum_k \frac{\delta E}{\delta y_k} \cdot \frac{dy_k}{dx_k} \cdot w_{kj}$$

where the index c is omitted.

One of the key components of multi-layer networks is the activation function, of which Rectified Linear Unit has shown success in improving convergence and performance compared to sigmoid units [44]. Parametric ReLU (PReLU) is a similar but more generalized activation function defined as

$$f(y_i) = \begin{cases} y_i, & : y_i > 0 \\ a_i y_i, & : y_i \leq 0 \end{cases}$$

for which ReLU can be interpreted as a special case where $a_i = 0$, meaning ReLU can be defined as $f(y_i) = max(0, y_i)$.

Deep multi-layer neural networks have been shown to be an improvement over non-deep approaches, although previous attempts with standard gradient descent from random initialization has shown poor results and new initialisation schemes needed to be created in order to ensure faster convergence [45]. Logistic Sigmoid activation is not suitable for deep neural networks with random initialization, because the mean value can drive the top hidden layer into saturation. Saturated units are capable of slowly moving out of saturation by themselves, which results in "plateaus" in neural network training. Soft sign networks have little non-linearity, which means they are robust to initialization. Normalized initialization combined with second order information shows good performance.

One of the possible weight optimizers for neural network architectures is the Adam stochastic optimizer [46]. Adam, from "adaptive moment estimation", is suited for problems with large data sets and parameters as it is straightforward, computationally efficient and has low memory requirements due to it using only first-order gradients. The magnitudes of its parameter updates are invariant to diagonal re-scaling of gradients. It is also suitable for very noisy or sparse gradients and non-stationary objectives, and its hyper parameters require little tuning. The focus of Adam is on "the optimization of stochastic objectives in high-dimensional parameters spaces". The scikit-learn implementation of multi-layered Perceptrons uses this weight optimization strategy by default[3].

Artificial neural networks show marginal improvement over random forests for modelling the energy consumption of buildings [47], meaning ANNs are a worthy contender. The task of energy prediction includes multi-dimensional complex data. Energy prediction models help facility managers to evaluate performance and hence improve energy efficiency, and data driven approaches are suitable due to their lack of need for detailed simulation and fast response time. The performance of a feed-forward back-propagation artificial network was compared with random forest hourly HVAC energy consumption of a hotel in Madrid, Spain. While the models offer comparative prediction power, ANN performed marginally better in terms of root mean squared error than RF. However, ensemble models like RF offer the advantages of easier tuning and categorical variable modelling [48].

4.2. Evaluation methodology

Evaluation is the process of "ascertaining the decision areas of concern", which involves the selection, collection and analysis of appropriate information in the form of a data summary that decision makers can use to select among alternatives [49]. The evaluation procedure depends on the decision to be made, and the information gathered should be presented to the decision-maker in a helpful and effective manner. In this study, two kinds of evaluation are used: error measures and statistical tests. Error measures can be used to compare accuracy of forecasting methods for a given series of data [50]. Error measures are important in calibrating and refining forecast or prediction models, as the effect of changing

[3] MLP: `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html`

different parameters can be visualized. Single error measures are desirable due to difficulty in interpreting multiple measures. Root Mean Squared Error was selected for this study due to its sensitivity to large errors. Statistical tests are used to draw conclusions on series of observational data [51]. Tests are not designed to prove or disprove anything but to reject a null hypothesis within a certain probability region also known as the critical region. There are several test procedures depending on the type of variables and what is needed, from which the most relevant to this study are the measurement of association between two measurement variables and the comparison of variance between groups of data [52]. The selected methods for these tasks were Linear Regression and One-Way Anova, respectively.

4.2.1. RMSE

Root Mean Squared Error (RMSE) is a scale-dependent measure and is useful for evaluating error in situations in which different methods are compared on the same set of data [53], meaning it is useful for comparing different learning algorithms on the same testing set. Scale-dependent accuracy measures such as RMSE are evaluation tools whose scale is dependent on the scale of the data, meaning they represent the error in the context of the data well. They are however sensitive to outliers, and hence in this work RMSE is calculated for percentile subsets in addition to the whole testing data set to provide a more representative visualization of the actual behaviour of the algorithms. The formula of RMSE is the following

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

where T is the number of items in the test set, y_t is the value of the dependent variable at item t, and $\hat{y}_t$ is the predicted value of the dependent variable for item t.

4.2.2. Linear Regression

Linear regression is useful in situations where one needs to find the association between two measurement variables, providing values for the strength of association and creation of functions to predict values of unknown variables [52]. Linear regression returns the slope b and intercept a of the proposed linear function, strength of association $r \in \{-1, 1\}$, p-value and the standard error of the estimated gradient. This is the case also in the Scipy implementation used in this study[4]. Linear regression assumes normality of data distribution, linearity of data, and independence of data points, however linear regression and classification have both been found robust to non-linearity. A common way to do linear regression is fitting an optimal line between data

[4] Linregress: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

points using least square regression. Least square regression minimizes the square distance of the linear function involving the independent variable x to all the data points, or in other words minimizes $(y - \hat{y})^2$ where y is the value of the dependent variable and $\hat{y}$ the value predicted by the linear function $\hat{y} = a + bx$.

Linear regression with least square method calculates values of a and b with the following equation [54]:

$$a = \bar{y} - b\bar{x}, \quad b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

where $\bar{x}, \bar{y}$ are the mean values of x and y, respectively. The p-value can be derived by comparing the standard errors of the intercept s_a and gradient s_b to the t-distribution on $n - 2$ degrees of freedom. The standard errors can be calculated as

$$s_a = s\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}, \quad s_b = \frac{s}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}},$$

where

$$s = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2 - b\sum_{i=1}^{n}(x_i - \bar{x})^2}{(n - 2)}}.$$

4.2.3. One-way ANOVA

One-way ANOVA (analysis of variance) can be used for situations where there is one nominal variable and one measurement variable, and it works by comparing the distribution of the measurement variable within groups defined by the nominal variable [52]. After calculating the mean, the variance of the measurement variable values for each group is used to form the test statistic F_s which is the ratio of the variance of the means of the group to variance within the group. The probability of achieving a certain F_s under a null hypothesis can be calculated due to its known distribution. This depends on the among-group degrees of freedom as number of groups minus one in the numerator and within-group degrees of freedom as number of observations minus number of groups in the denominator. Given n_g is the number of observations in group g and x_gi is the ith value observation in g, the algorithm for ANOVA is as follows [55]:

1. For all k groups together as set T, calculate the sum of squared deviates as $SS_T = \Sigma x_i^2 - \frac{(\Sigma x_i)^2}{n_T}$.

2. For each group $g \in k$, calculate the sum of squared deviates as $SS_g = \Sigma x_{gi}^2 - \frac{(\Sigma x_{gi})^2}{n_g}$.

3. Take the sum of all SS_g in k as $SS_{wg} = \sum_{g=0}^{k} SS_g$

4. Calculate $SS_{bg} = SS_T - SS_{wg}$.

5. Calculate the degrees of freedom $df_T = n_T - 1$, $df_{bg} = k - 1$, and $df_{wg} = n_T - k$.

6. Calculate the mean-square values $MS_{bg} = \frac{SS_{bg}}{df_{bg}}$ and $MS_{wg} = \frac{SS_{wg}}{df_{wg}}$.

7. Calculate $F_s = \frac{MS_{bg}}{MS_{wg}}$.

F_s can then be compared to its distribution table using the acquired value and degrees of freedom to acquire a measure of significance.

5. EXPERIMENTAL SETUP

The aim of this experiment was to use data fusion and machine learning to determine the next most probable purchase time and value from event-based data. The experiment consisted of three parts, first of which was to explore preliminary performance by testing three algorithms on the full set of selected data: Random Forest, Support Vector Machine and Artificial Neural Network in the form of a Multi-Layer Perceptron. The second part of the experiment was determine what factors predict in game purchase behaviour by analyzing the correlation of all features to future purchase value and days until purchase. This is done using Linear Regression on numeric variables and One-Way ANOVA for categorical variables. Finally, the most correlated features were formed into feature sets and the three previous algorithms were trained on these feature sets to determine the most suitable algorithm for predicting these labels. Using RMSE, model performance was evaluated compared to a trivial baseline predictor of returning the average for the given data set.

5.1. Program Pipeline

The program is in the form of a Jupyter Notebook[1] file with sets of functions meant to be run sequentially represented as a process in Figure 4. Pre-processed game data is taken either partially or completely depending on the test case from its respective BigQuery database and turned into a Pandas DataFrame[2] for pre-processing. Data Pre-Processing involves data format conversions, removal of data rows with missing values and dividing features into categorical and numeric features for correlation testing. The feature "paid_user" was mapped from Boolean (True, False) into a string ("Yes", "No") for easier handling by SKLearn model fitting interfaces. The features were divided into categorical and numeric features originally based on their data type, mapping string type features as categorical and any other as numeric. However, the day of week, day of month, activity month and day of year were categorized as categorical variables due to their limited selection of possible values. In addition, one-hot encoding for these features was seen to provide better value to learning, as enabling the assignment of different weights for each day can uncover periodic behaviour.

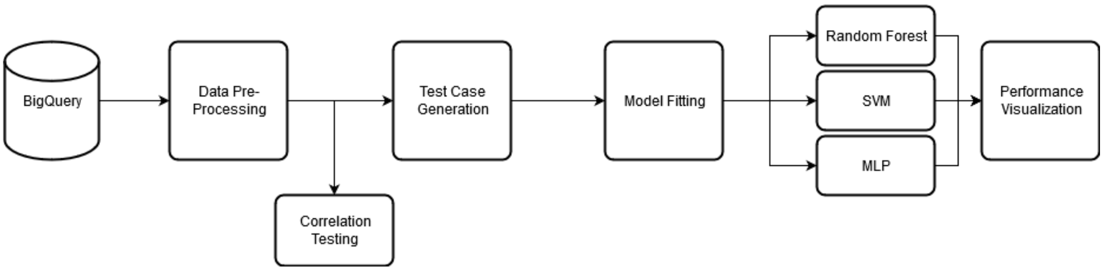

Figure 4. Program pipeline of ProGame data analytics.

[1] https://jupyter.org/

[2] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

Correlation testing is an alternate process to Test Case Generation where the data is analysed for linear correlation, using One-Way ANOVA for categorical features and linear regression for numeric features. This process enables selection of candidate features by providing measures of correlation to each of the two labels. It generates four tables consisting of rows of variable names and the respective correlation parameters for each algorithm, meaning the f-value and p-value for One-Way ANOVA and slope, intercept, R-value, p-value and standard error for linear regression. These four tables correspond to the correlation of both categorical features and numeric features to each of the labels.

Test Case Generation forms a recursive JSON-style dictionary representing a set of test cases. This test case dictionary consists of a name and it's associated two lists, a list of feature columns to use from the DataFrame and a list of labels to be predicted. To create appropriate feature combinations for testing the whole data set, it is possible to use a class called TestGenerator to form a recursive dictionary. TestGenerator internally takes dictionaries of named label lists and named feature set lists and forms named combinations for all the feature sets and labels to form the needed test scenarios.

Model Fitting provides a function for training instances of any model supported by scikit-learn when provided with data and test cases. When provided with a test case dictionary, the original data frame and a function that initializes the class of the needed model, it first splits the values into categorical and numeric values as in data pre-processing. The function applies one-hot encoding for categorical features. Due to the logarithmic distributions of the numeric features, for numeric features it applies either a logarithmic function if the minimum value of the variable is larger than -1 or normalizes the value otherwise. The input is split into a training set and test set, with a ratio of 75% to 25% respectively. The model is initiated and trained on the training set, after which the model is made to predict the labels from the features of the testing set. When these predictions and the correct labels are combined, this results in a dictionary of the test cases with a list of values of the label to be predicted and a list of their algorithm-predicted equivalent based on the same data point for each test case.

Finally, Performance Visualization uses the value and predicted value pairs to calculate RMSE for both the whole test case and plot the error per percentile for all the test cases. The overall RMSE for the whole testing set is calculated for each test case and inserted into a table with columns for the test case name and the RMSE values for both labels to be predicted. To model the performance of the algorithms on different value ranges, for both of the labels the value range was divided into sets of percentile ranges of size 5%, and the RMSE value for all data points with label within that percentile range was calculated. These errors were then plotted as a line graph of percentile range to error.

5.2. Testing with all feature sets

Testing on this whole feature set is used for initial experimentation on learning model performance, and further used to compare against a more limited feature set optimized to each label with linear correlation. The RMSE for both target variables (days until purchase and value of the purchase in USD) is evaluated as overall performance and per value percentile performance for selected combinations of feature sets. Four different combinations of feature sets are used: "Generic", "Generic + RFM", "Generic + 14-Point" and "All" which included all previous feature sets. The "Generic" feature set is a combination of the "General" and "Daily" feature sets. Predictions with these are tested separately for days before purchase and purchase value in USD, making a total of eight test cases per algorithm. Tests are run for the Random Forest, Support Vector Machine and Multi-Layer Perceptron algorithms with a data point sample of size 50 000. The Multi-Layer Perceptron is run with a single hidden layer of 100 neurons. All parts of the pipeline except correlation testing are used in this test scenario, and the feature-label combinations were created with TestGenerator.

5.3. Significant feature selection

For both target variables, all the aggregated features are analysed with statistical tests to find the features with the most significance for each of the variables. A sample of 50 000 data points is randomly selected from the total data set. Using the correlation testing component numeric features are analysed with linear regression and one-way ANOVA is used for categorical features. For categorical features the criterion of acceptance is defined as whether the f-value of the feature was at the level of hundred or more, while for numeric features the criterion of acceptance is whether the absolute value of the R-value was higher than 0.2 for a medium effect.

5.4. Designated feature testing without hyperparameter optimization

To create a lightweight but effective model, the best performing features based on the significant feature selection are tested on all of the algorithms as in the "all features" test case, evaluating RMSE for both overall performance and per value percentile performance. The experiment is first run by using both the features found to correlate with value in USD and for days until purchase the prediction of both labels. Afterwards, the experiment is repeated with "optimized" features, meaning for both labels only the features most significant to that label are used.

6. RESULTS

6.1. Results from the whole feature set

Table 1 shows that compared to the baseline predictor using the average of training data, several algorithm and data set combinations showed improvement in the effectiveness of predicting both labels. The RMSE for the baseline was 10,08 for value in USD and 62,17 for days until purchase. Random Forest had the lowest RMSE for both value in USD (RMSE=7,82) and for days until purchase (RMSE=48,35) when using all features. All algorithms demonstrated efficient feature selection, as their performance with the "Generic + RFM" feature set was similar to their performance on the "All" feature set, although Multi-Layer Perceptron had a slightly larger error for days until purchase in the "All" data set (RMSE=51,21) than "Generic + RFM" data set (RMSE=50,81). MLP had the largest RMSE for the "Generic" feature set (f=10,41), while Support Vector Machine showed the largest error for days until purchase for the "Generic" feature set (f=60,35). For value in USD, the variance of the RMSE for RF was 0.52, for SVM 0.08 and for MLP 0.34. For days until purchase, the variance of the RMSE for RF was 1.55, for SVM 0.98 and for MLP 0.86. For both features, RF had the highest variance, while SVM had the lowest variance.

Algorithm + feature set	**RMSE, Value in USD**	**RMSE, Days until purchase**
Baseline	10,08	62,17
RF: Generic	9,35	51,20
RF: Generic + RFM	7,88	48,42
RF: Generic + 14-Point	9,22	50,43
RF: All	7,82	48,35
SVM: Generic	9,77	60,35
SVM: Generic + RFM	9,22	58,22
SVM: Generic + 14-Point	9,77	59,74
SVM: All	9,21	58,00
MLP: Generic	10,41	52,39
MLP: Generic + RFM	9,12	50,81
MLP: Generic + 14-Point	9,86	53,15
MLP: All	8,95	51,24

Table 1. RMSE per label for each algorithm and feature set combination

When comparing RMSE performance per percentile for RF, Figure 5 shows that the "All" feature set had the best predictive power for value in USD in the lowest percentiles (<25) and highest percentiles (>90). However, it shows anomalous behaviour between the 55th and 90th percentile, as it shows highest RMSE in those areas. The performance of the rest of the algorithms were nearly indistinguishable. For predicting days until purchase, the RMSE values of all test cases were nearly identical. For SVM, Figure 6 demonstrates that the different feature sets had nearly identical performance per percentile for both labels, with the "All" feature set showing slight improvement in the higher percentiles. For MLP, it can be sseen in Figure 7 that the performance of the different feature sets show clearer differentiation. The "All" feature set shows the best performance for predicting Value in USD until the 50th percentile, after which the "Generic + 14-Point" show the lowest RMSE value until the 90th percentile. The "Generic" feature set had the lowest RMSE for predicting days until purchase until the 80th percentile, after which the "All" data set had the lowest RMSE. Overall, the RMSE values of errors are stable at start and begin increasing exponentially after the 60th percentile.

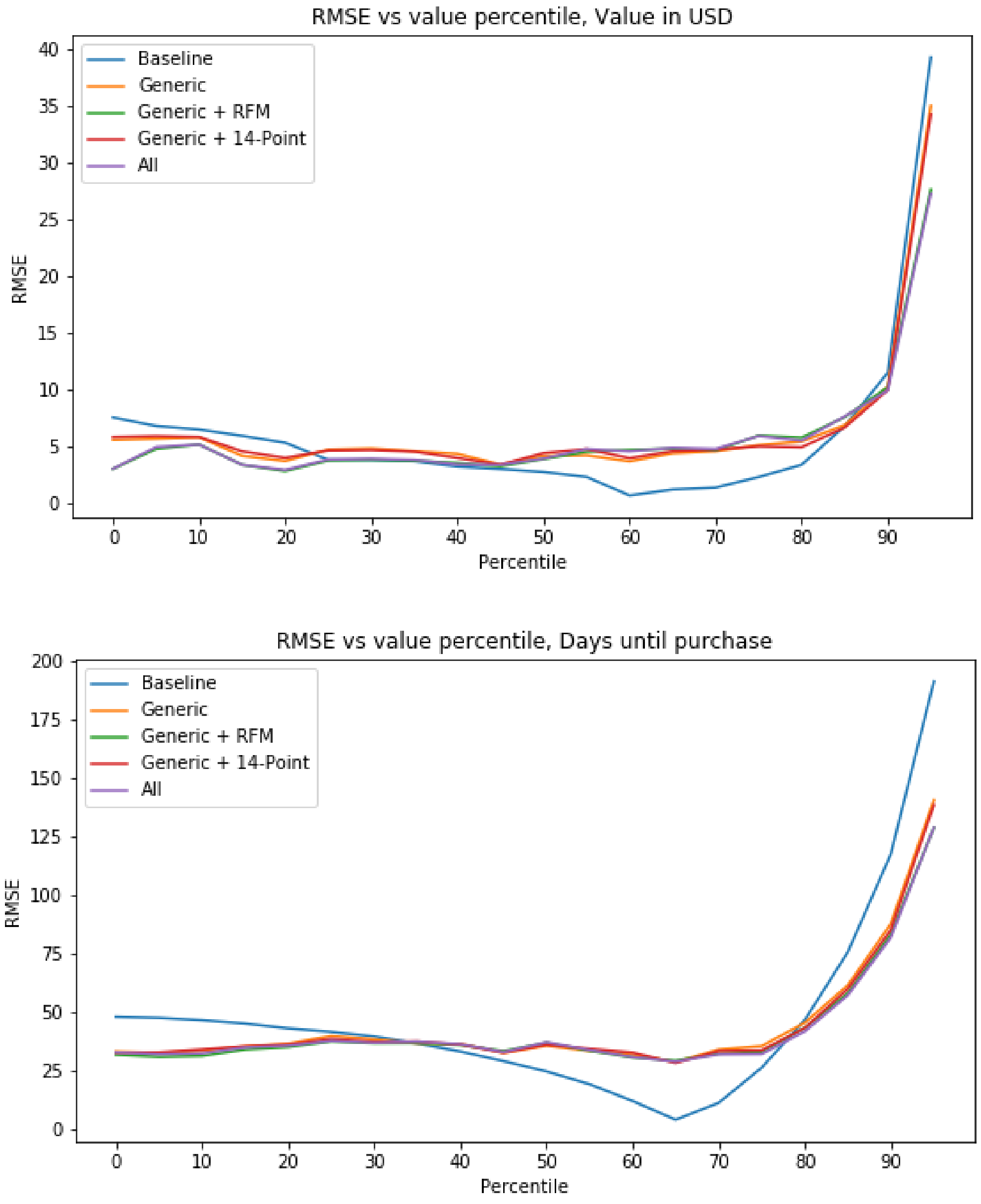

Figure 5. RMSE per percentile for Random Forest.

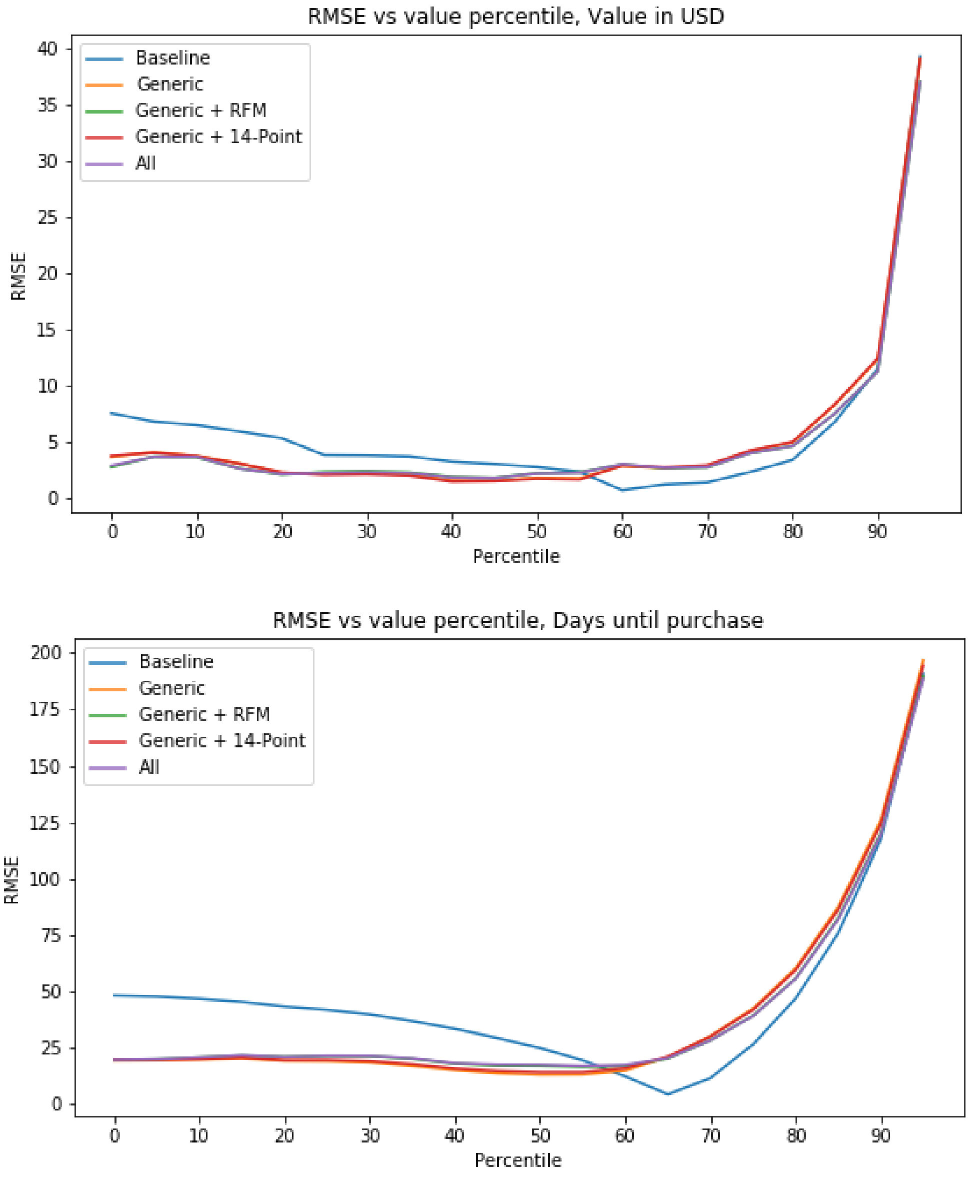

Figure 6. RMSE per percentile for Support Vector Machine.

6.2. Linear analysis of feature significance

Table 2 shows that the categorical feature with the largest correlation with purchase value was the "platform" feature representing the operating system of the phone ($f = 777,99$). Of note is that the "paid_user" variable, representing whether the user had been attracted to the game via an advertisement campaign or not, had the least correlation with purchase value ($f = 0,49$).

Table 3 shows that the categorical features with the most significant correlation with days until purchase were the "activity_month" representing the activity month ($f = 727,99$) and whether the user was converted via an advertisement campaign ($f = 727,80$). The feature with the least correlation with days until

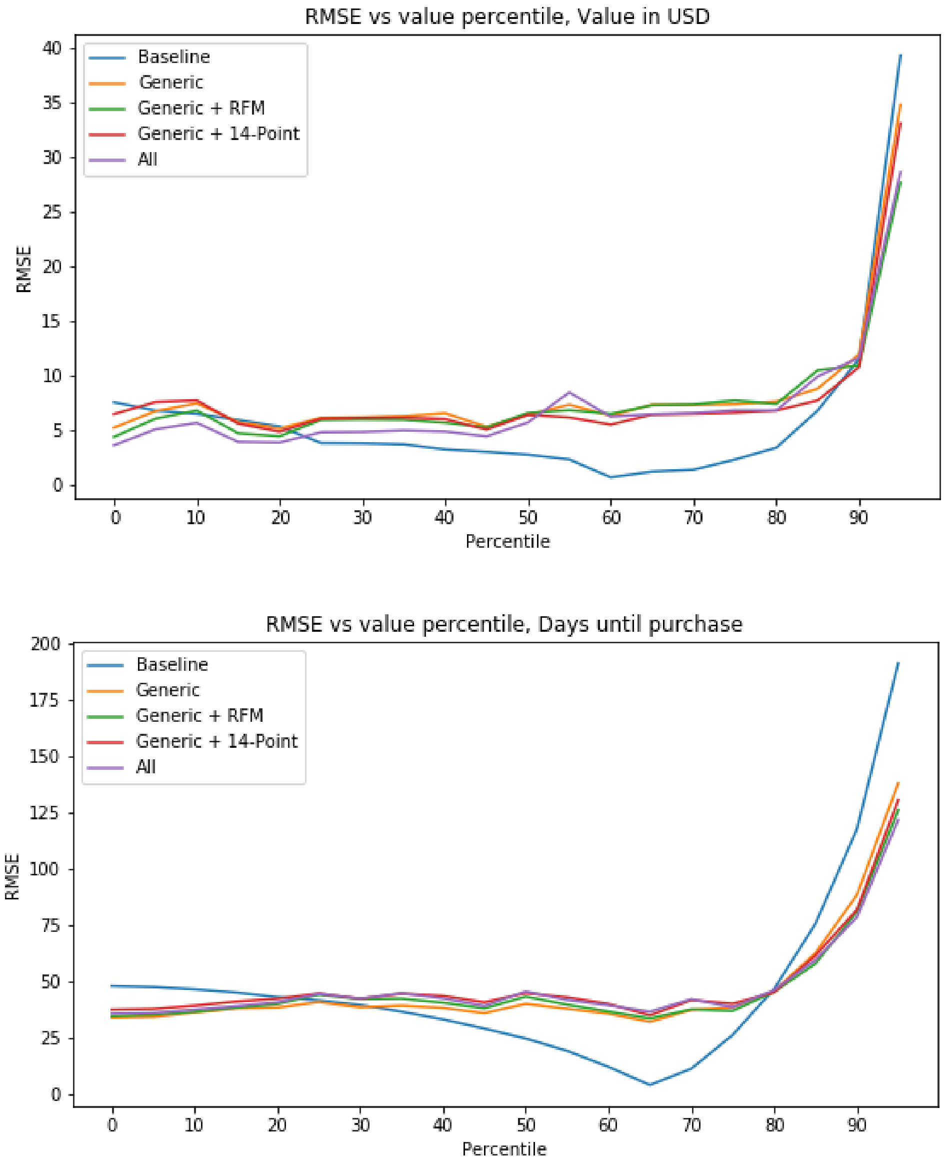

Figure 7. RMSE per percentile for Multi-Layer Perceptron.

Feature	**f-value**	**p-value**
platform	777,99	0,00
country	24,95	0,00
paid_user	0,49	0,49
activity_day_of_week	0,78	0,58
activity_day_of_month	1,05	0,40
activity_month	6,51	0,00
activity_day_of_year	1,19	0,01

Table 2. Correlations with value in USD for categorical variables.

purchase is "activity_day_of_week" representing the day of the week of the activity ($f = 0,53$).

Feature	f-value	p-value
platform	49,06	0,00
country	3,32	0,00
paid_user	780,57	0,00
activity_day_of_week	0,53	0,78
activity_day_of_month	1,20	0,21
activity_month	727,80	0,00
activity_day_of_year	25,58	0,00

Table 3. Correlations with days until purchase for categorical variables.

Table 4 shows that the numeric features most significantly correlated with purchase value were "cum_purchases" representing the number of cumulative purchases ($r = 0,25$), "lead_value_in_usd" representing the value of the previous purchase in USD ($r = 0,49$), "max_purchase_value" representing the value of the maximum purchase of the customer ($r = 0,45$), and "avg_purchase_value" representing the average of the purchase value of the customer ($r = 0,50$). Of note is that the three features with the highest r-value are all related to the value of previous purchases. The features with the lowest correlation with purchase value were the features "daily_delta_season_rank", "delta_min_season_rank", "delta_min_season_rank_sum" and "delta_avg_rank_avg" ($r = 0,00$) which are all related to the change of various rankings of the player over time.

Table 5 shows the numeric features with the most significant correlation with days until purchase were "days_since_last_purchase" representing days since last purchase ($r = 0,31$), "purchases_sum" representing the sum of purchases in the last 14 days ($r = -0,22$), "purchases_avg" representing the average of purchases made in the last 14 days ($r = -0,20$), and "purchases_max" representing the maximum number of purchases in the last 14 data points ($r = -0,25$). Of note is that the days since last purchase had the largest effect. The features with the lowest correlation with days until purchase were the features "daily_min_season_rank" and "delta_min_season_rank" related to the smallest in-game seasonal rank recorded and the features "daily_gem_state", "delta_coin_state_sum", "delta_gem_state_sum", "delta_coin_state_avg", "delta_gem_state_avg" and "delta_gem_state_max" related to changes in in-game currency ($r = -0,00$).

6.3. Results with designated feature sets

Multi-Layer Perceptron had the lowest RMSE when using the whole feature set in Table 6 for predicting the value in USD (RMSE=7,61), with an improvement of 21.2% over the baseline predicting the average of value in USD (RMSE=9,66). Random forest and Multi-Layer Perceptron both had the overall lowest RMSE for predicting value of the next purchase when using "optimized" features

(RMSE=7,53) as shown in Table 7, with an improvement of 22.1% over the baseline (RMSE=9,66). Of note is that the RMSE of Random Forest with "optimized" features was lower than that of Multi-Layer Perceptron with the whole feature set, implying that some less-significant variables hinder the performance of the latter. Of the tested algorithms, Support Vector Machine had the highest RMSE for predicting value in USD both when using the selected features (RMSE=8,11) and optimized features (RMSE=8,12).

Multi-Layer Perceptron had the lowest RMSE for predicting days until purchase for both selected features (RMSE=48,77) and optimized features (RMSE=51,04), with an improvement of 20.7% over the baseline with selected features (RMSE=61,52) and an improvement of 17.0% over the baseline with optimized features (RMSE=61,52). This indicates that when using "optimized" features there is not enough dimensions to effectively model behaviour using neural networks. However, Multi-Layer Perceptron did not fully converge for either variable case with the default hyper-parameters, meaning further adjustment can provide increased accuracy in the future. Of the tested algorithms when predicting days until purchase, Support Vector Machine had the highest RMSE when using selected features (RMSE=53,84) while Random Forest had the highest RMSE when using optimized features (RMSE=54,92).

When comparing the values of labels as percentile ranges compared to their respective RMSE as shown in Figure 8 and Figure 9, Support Vector Machine showed the best overall accuracy in the first 80 percentiles, although the performance for the highest percentiles was slightly lower compared to the other two algorithms. Multi-layer Perceptron showed the second-best performance, providing a small improvement over Random forest in both cases.

6.4. Summary of results

Random Forest had the best performance when looking at the RMSE on the whole feature set. Based on the RMSE of percentiles, using all features in the feature set showed best prediction accuracy. Random forest showed the largest variance in RMSE between feature sets, while SVM showed the lowest. The improvement does not spread evenly across percentiles. Baseline method showed the best performance in the 50 to 70 percentiles, which is expected. The "All" test case was the best at predicting values in the lower and higher percentiles of Multi-Layer Perceptron, while "Generic + 14-point" was a middle-of-the-road performer. For the rest of the algorithms, the differences in RMSE between test cases were less clearly visible.

Based on linear analysis, the best predictors for the value of the next purchase were the operating system of the phone, number of cumulative purchases, the value of the previous purchase in USD, the value of the maximum purchase of the customer and the average purchase value of the customer. The best predictors for days until purchase were the activity month, whether the user was converted via an advertisement campaign or not, days since last purchase, the sum of purchases in the last 14 days, the average value of purchases made in the last 14 days, and

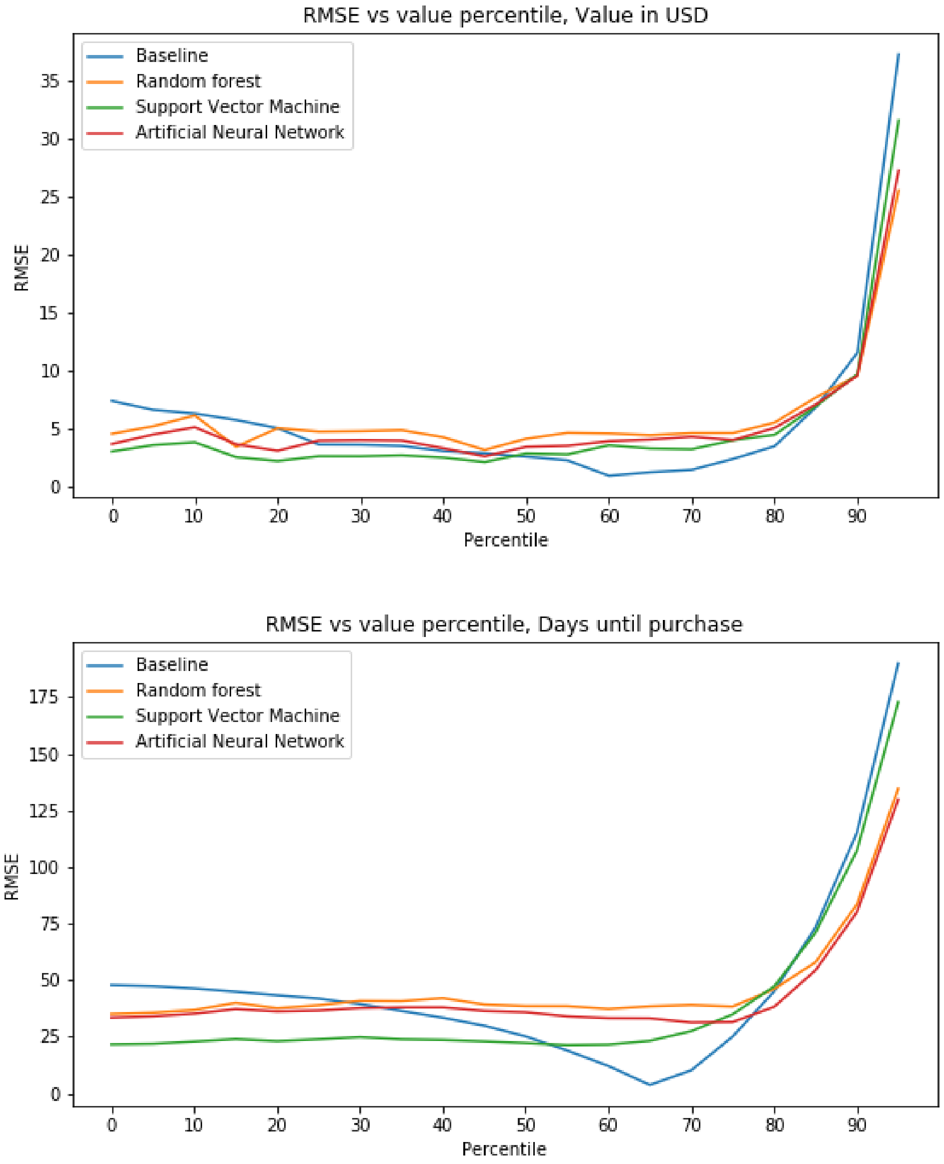

Figure 8. RMSE values per percentile for selected features.

the maximum number of purchases in the last 14 data points. Changes within in-game variables were found to have little correlation with either variable.

Artificial neural networks had the lowest overall RMSE in both predicting value in USD and days until purchase when using the set of features found by this linear analysis, although using the most limited feature set in the amount of features showed lower performance. Support vector machines had the lowest error for optimized features before the 70th percentile for both labels when comparing RMSE per percentile. The models generated in the study show marginal improvement over a trivial baseline estimator using the mean of the global value of the respective labels, with the lowest RMSE error for next purchase value being 7,53 USD and days until purchase being 48,77 days. Compared to the baseline values, the improvements were 22,0% and 20,7% respectively.

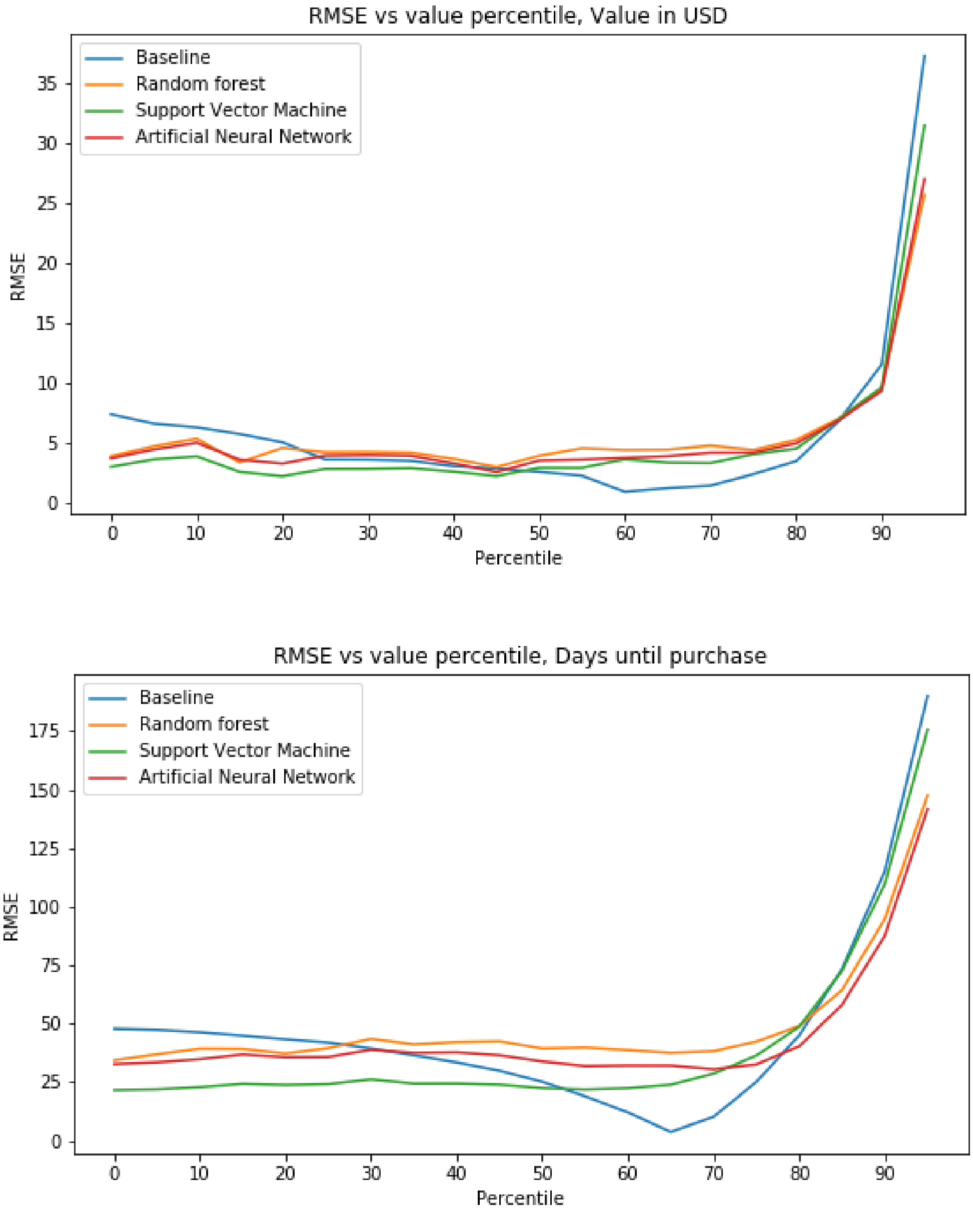

Figure 9. RMSE values per percentile for optimized features.

Feature	Slope	Intercept	r-value	p-value	Standard Error
retained_days	0,00	8,19	0,06	0,00	0,00
purchases	1,32	8,63	0,04	0,00	0,16
net_revenue	0,45	8,56	0,13	0,00	0,02
int_ads_watched	−0,21	9,01	−0,10	0,00	0,01
rew_ads_watched	−0,23	9,19	−0,07	0,00	0,01
daily_min_rank	0,00	8,18	0,10	0,00	0,00
daily_min_season_rank	0,01	8,45	0,05	0,00	0,00
daily_coin_state	0,00	8,68	0,02	0,00	0,00
daily_gem_state	0,00	8,69	0,01	0,04	0,00
daily_delta_rank	0,00	8,59	0,03	0,00	0,00
daily_delta_season_rank	0,00	8,68	0,00	0,49	0,00
cum_purchases	0,13	7,63	0,25	0,00	0,00
lead_value_in_usd	0,47	5,72	0,49	0,00	0,00
max_purchase_value	0,27	6,21	0,45	0,00	0,00
avg_purchase_value	0,52	5,57	0,50	0,00	0,00
purchase_frequency	3,02	8,37	0,12	0,00	0,11
delta_min_rank	0,00	8,67	0,01	0,02	0,00
delta_min_season_rank	0,00	8,69	0,00	0,45	0,00
delta_coin_state	0,00	8,69	0,02	0,00	0,00
delta_gem_state	0,00	8,69	−0,01	0,00	0,00
sess_length_seconds	0,00	8,44	0,03	0,00	0,00
total_sessions	0,07	8,16	0,05	0,00	0,01
days_since_last_purchase	−0,00	9,05	−0,06	0,00	0,00
purchases_sum	0,76	8,25	0,12	0,00	0,03
int_ads_watched_sum	−0,03	9,27	−0,13	0,00	0,00
rew_ads_watched_sum	−0,03	9,58	−0,09	0,00	0,00
delta_min_rank_sum	0,00	8,67	0,01	0,27	0,00
delta_min_season_rank_sum	−0,00	8,69	−0,00	0,31	0,00
delta_coin_state_sum	0,00	8,68	0,04	0,00	0,00
delta_gem_state_sum	0,00	8,69	0,03	0,00	0,00
total_sessions_sum	0,01	7,96	0,07	0,00	0,00
sess_length_seconds_sum	0,00	8,30	0,05	0,00	0,00
purchases_avg	6,48	8,39	0,09	0,00	0,32
int_ads_watched_avg	−0,35	9,29	−0,13	0,00	0,01
rew_ads_watched_avg	−0,49	9,78	−0,11	0,00	0,02
delta_avg_rank_avg	0,00	8,69	−0,00	0,78	0,00
delta_avg_season_rank_avg	−0,02	8,69	−0,01	0,31	0,02
delta_coin_state_avg	0,00	8,68	0,04	0,00	0,00
delta_gem_state_avg	0,00	8,69	0,03	0,00	0,00
total_sessions_avg	0,09	8,01	0,06	0,00	0,01
sess_length_seconds_avg	0,00	8,38	0,04	0,00	0,00
purchases_max	1,62	8,14	0,12	0,00	0,06
int_ads_watched_max	−0,13	9,35	−0,14	0,00	0,00
rew_ads_watched_max	−0,25	10,02	−0,11	0,00	0,01
delta_min_rank_max	0,00	8,60	0,05	0,00	0,00
delta_min_season_rank_max	0,03	8,56	0,04	0,00	0,00
delta_coin_state_max	0,00	8,69	0,01	0,30	0,00
delta_gem_state_max	0,00	8,69	0,04	0,00	0,00
total_sessions_max	0,06	7,98	0,05	0,00	0,01
sess_length_seconds_max	0,00	8,36	0,03	0,00	0,00
size_of_window	−0,01	8,77	−0,01	0,01	0,00

Table 4. Correlations with value in USD for numeric variables.

Feature	**Slope**	**Intercept**	**r-value**	**p-value**	**Standard Error**
retained_days	0,04	41,70	0,14	0,00	0,00
purchases	−24,33	49,88	−0,10	0,00	1,08
net_revenue	−1,66	49,30	−0,07	0,00	0,11
int_ads_watched	0,95	47,36	0,07	0,00	0,06
rew_ads_watched	0,67	47,35	0,03	0,00	0,10
daily_min_rank	0,00	46,74	0,06	0,00	0,00
daily_min_season_rank	0,00	48,79	0,00	0,85	0,01
daily_coin_state	0,00	48,76	0,01	0,00	0,00
daily_gem_state	0,00	48,82	−0,00	0,44	0,00
daily_delta_rank	−0,00	49,58	−0,04	0,00	0,00
daily_delta_season_rank	0,06	48,71	0,01	0,04	0,03
cum_purchases	−0,61	53,79	−0,17	0,00	0,02
lead_value_in_usd	−0,92	54,66	−0,14	0,00	0,03
max_purchase_value	−0,71	55,36	−0,18	0,00	0,02
avg_purchase_value	−1,19	55,97	−0,17	0,00	0,03
purchase_frequency	−22,65	51,19	−0,13	0,00	0,75
delta_min_rank	0,00	49,05	−0,02	0,00	0,00
delta_min_season_rank	−0,01	48,82	−0,00	0,76	0,03
delta_coin_state	0,00	48,82	−0,01	0,23	0,00
delta_gem_state	0,00	48,81	−0,01	0,23	0,00
sess_length_seconds	−0,00	53,71	−0,10	0,00	0,00
total_sessions	−0,57	52,95	−0,06	0,00	0,04
days_since_last_purchase	0,12	36,99	0,31	0,00	0,00
purchases_sum	−9,33	54,18	−0,22	0,00	0,19
int_ads_watched_sum	0,15	45,73	0,10	0,00	0,01
rew_ads_watched_sum	0,15	44,41	0,07	0,00	0,01
delta_min_rank_sum	0,00	49,16	−0,02	0,00	0,00
delta_min_season_rank_sum	−0,02	48,83	−0,01	0,04	0,01
delta_coin_state_sum	0,00	48,82	−0,00	0,57	0,00
delta_gem_state_sum	0,00	48,82	−0,00	0,78	0,00
total_sessions_sum	−0,03	51,81	−0,04	0,00	0,00
sess_length_seconds_sum	0,00	53,41	−0,08	0,00	0,00
purchases_avg	−97,40	53,24	−0,20	0,00	2,11
int_ads_watched_avg	1,20	46,77	0,07	0,00	0,08
rew_ads_watched_avg	1,36	45,79	0,04	0,00	0,14
delta_avg_rank_avg	0,00	49,43	−0,03	0,00	0,00
delta_avg_season_rank_avg	−0,25	48,84	−0,01	0,03	0,11
delta_coin_state_avg	0,00	48,82	−0,00	0,37	0,00
delta_gem_state_avg	0,00	48,82	−0,00	0,77	0,00
total_sessions_avg	−0,69	53,80	−0,06	0,00	0,05
sess_length_seconds_avg	−0,00	55,09	−0,11	0,00	0,00
purchases_max	−21,87	56,18	−0,25	0,00	0,39
int_ads_watched_max	0,42	46,69	0,07	0,00	0,03
rew_ads_watched_max	0,46	46,34	0,03	0,00	0,07
delta_min_rank_max	0,00	49,04	−0,02	0,00	0,00
delta_min_season_rank_max	−0,04	49,02	−0,01	0,05	0,02
delta_coin_state_max	0,00	48,81	0,01	0,30	0,00
delta_gem_state_max	0,00	48,82	−0,00	0,76	0,00
total_sessions_max	−0,45	54,43	−0,06	0,00	0,03
sess_length_seconds_max	−0,00	56,50	−0,11	0,00	0,00
size_of_window	0,16	46,63	0,04	0,00	0,02

Table 5. Correlations with days until purchase for numeric variables.

Algorithm	**RMSE, value in USD**	**RMSE, days until purchase**
Baseline	9,66	61,52
Random forest	7,70	52,34
Support Vector Machine	8,11	53,84
Artificial Neural Network	7,61	48,77

Table 6. RMSE values for selected features.

Algorithm	**RMSE, value in USD**	**RMSE, days until purchase**
Baseline	9,66	61,52
Random forest	7,53	55,98
Support Vector Machine	8,12	54,92
Artificial Neural Network	7,53	51,04

Table 7. RMSE values for optimized features.

7. ETHICAL ASPECTS

As outlined by Zwitter in 2014, the speed of development of Big Data and its associated phenomena has "surpassed the capacity of the average consumer to understand his or her actions and their knock-on effects" [56]. The development of ethics in the context of big data focuses away from knowable outcomes of individual decisions, and steers towards accounting for actions by several users who unknowingly take actions with unintended consequences. As a supplement to digital literacy, children, adolescents and adults need to be educated about the possible consequences of leaving a digital footprint. This educational gap needs to be considered in social science research to draw conclusions about the ethical aspects of using anonymous social data, which can be used to draw conclusions about groups. Zwitter outlines three possible developments in law and politics. Political investigators will use specialized data science methods to investigate new kinds of digital manipulation of public opinion. Social and legal services such as law enforcement will re-conceptualize individual guilt. Finally, states will increasingly develop global strategies using global data and algorithms instead of regional experts. Considering the amount of data collected and available in this study, transparency is vital for ethical research. Providing customers access to examples of usage of their data, such as a summary of this study and its results, could help them understand how their data is used and to what purposes better than a list of access rights.

Mittelstadt and Floridi argued in 2016 that the understanding of ethical implications of Big Data lag behind the state of the art technology [57]. Mittelstadt and Floridi conducted a literature meta-analysis to summarize current knowledge and hypotheses on the ethical risks of the then-emerging phenomenon, Big Data. The five key areas of concern identified from related research were informed consent, privacy, ownership, epistemology and objectivity, and the divide in Big Data between those who have resources to analyse large data sets and those who lack it. Six additional novel areas of concern were suggested: dangers of ignoring group-level ethical harm, the importance of epistemology in Big Data ethics assessment, highly data saturated fiduciary relationships, the need to distinguish between "academic" and "commercial" usage of Big Data in terms of harm to subjects, future problems with intellectual property ownership regarding analysis of aggregated data, and the difficulty of providing access rights to data subjects that lack necessary resources. The usage of data in this study is for both academic and commercial purposes, and hence would fall somewhere in between the two given purposes. The ownership of data is a more complicated issue, as customers have a right to privacy but permission to collect, store and share the data has been given by the customer and data holders currently have rights to define to which purpose their data is used. Informed consent is vital, and legislation needs to be developed for defining the rights and responsibilities of both customer and data collector. These areas of concern are expanded upon in the following paragraphs.

Relying on separate automated tools for building awareness of privacy can prove difficult, as Schaub et al. found in 2016 by measuring the impact of tracking management browser extensions on privacy awareness and concern [7]. The impact of three popular extensions (Ghostery, DoNotTrackMe and Disconnect) on user attitudes was examined in a laboratory study involving 24 participants. Before using the tools, the participants assumed that web tracking occurs but were not aware of the specifics. All three extensions provided some increase of awareness, but ultimately the insight on tracking was limited. The users remained unaware of many aspects of tracking, such as who the tracking companies are, what data is collected and for what purpose the data is collected. The usage of these extensions increased privacy concern due to increased awareness of tracking, but the feeling of being protected by the extension reduced this concern. Some participants distrusted the extensions themselves or accused them of tracking the user. The study by Schaub et al. comes with several design implications in the context of tracker extensions, but which can also be applied for the data collection phase in research similar to this experiment in the future. Privacy risks associated with types of trackers should be emphasized over just showing the number of trackers. Alert bubbles should be used sparingly only in exceptional situations due to habituation. Very few users accessed any added information links in the main panel due to expectations of complexity and low necessity, and hence relevant explanations should be integrated in the main panel. The terminology should be also kept sensible in the context by avoiding the usage of confusing jargon. Clear setup materials ensure the users understanding of the correct model of functionality and reinforce the trustfulness of the extension. There is need to inform users about why trackers are present, what they are collecting and sharing and how they use the collected information. These insights can improve how privacy risks and implications are communicated to users for increased credibility, and hence in the context of purchase behaviour prediction should be incorporated into future systems built on customer data. Players of the game are given access to the developer's privacy policy meaning measures for informed consent are given but considering results by Schaub et al. they might be unlikely to be fully understood or even completely read. Future experiments can have a simplified in-game policy form with clear text with further explanations included in the form itself, and the link to privacy policy is only for further reading and details.

An article by Baruh and Popescu from 2017 looks at how big data analytics preempts individuals' ability to self-define and closes off opportunities to challenge or resist such inferences [8]. They argue that privacy protection based on "notice and choice" self-management frameworks fail to protect individual rights and undermine the concept of privacy itself. The two possible individual strategies, "assimilation" meaning reliance on market-provided privacy protection and "avoidance" which is the withdrawal from the market may result in less privacy options available. Companies claim that users expect real-time hyper-personalization, and that exchanging privacy information for these services is the default option, making privacy a form of payment instead of a right and emphasizing market efficiency over individual privacy concerns. Algorithms slice populations into abstract social categories that can be very different from those the user deems appropriate, and they reduce the right of the individual to define themselves. Big data analytics can shape important decisions for an individual while being ubiquitous and unintelligible for said individual and hence defining a person in a fleeting and unchallengeable manner. Baruh and Popescu further argue that "notice and choice" systems rationalize market withdrawal and create a power imbalance for the users that rely on market solutions. This withdrawal then legitimizes the argument of laxer privacy expectations for "digital natives". Overall, the automation of privacy management creates "discrimination" as the personal data of less digitally literate are exploited more extensively. Hence to restore user agency the collective aspect of privacy as collective value and phenomenon needs to be recognized, and privacy regulation needs to move away from assumptions of individual literacy and self-management. This research considers the prediction of purchase behaviour of individuals using aggregated data, and although this experiment did not involve clustering customers into groups based on their data, future derivatives should take self-definition into account. When using tools derived from this study, customers should be given the right to be involved in the prediction process and possibly change erroneous interpretations. In addition, building awareness on how data collection methods work requires public awareness campaigns.

Digital data is increasingly being called to be treated as a public good due to its value in emergencies and scarce national data augmentation, which Taylor examined in 2016 by evaluating how it fits with the corporate reality of big data [58]. Guidelines and frameworks for ethical principles for data sharing have been discussed but they have not gained traction within those with the highest value data, such as mobile network operators. The power dynamics implied by using data as a public good were examined as well as different incentives to adopt an ethical position on the topic. Taylor uses the idea of corporate data as an ecosystem of conflicting rights, duties and claims in contrast to the sharing imperative based on humanitarian value. Corporations are not only censoring data for risk aversion purposes related to misuse, but they have incentives to remain in control of the data to focus positive customer perception mainly on their firm. It is also in their best interests to keep data as a scarce and hence valuable resource for business insights and customer preferences. Data sharing is done via single-use constructs negotiated by two parties, which also contributes to date being very contextual. Projects have been created for partial querying of proprietary data without sharing the whole data set, but they do not define who is responsible for the ethical use of data and puts pressure on public certification bodies. However, looking for a single set of guidelines is most likely a wrong approach, while the lack of an overarching framework stimulates the development of data ethics, which is an emergent field. Partial information on the data collection procedure is given in this research, but access to it is limited due to above interests. However, this research gives insight on said data, and sharing this insight within a scientific context with permissive access rights enables knowledge transfer and future expansion and development in the field of data analysis. This can initiate a shift in the perception of data ownership towards common ownership.

Releasing organizational data to the public might produce a security risk however, as anonymization might prove ineffective due to presence of de-anonymization procedures such as the geolocated data attack adapted by Gambs et al. already in 2014 [59]. As GPS-equipped devices are common, massive amount of location data is available for privacy breaches, although arguments are presented that location data is anonymous. To prove this wrong, in a de-anonymization attack an adversary uses a set of ability traces to infer the identity of an individual, and geological data sets are especially vulnerable. A novel Mobility Markov Chain tool was implemented by Gambs et al. to build user models from their respective mobility traces. Distance metrics were designed to allow the de-anonymizers to re-identify users based on the closeness or similarities between two MMCs. Using these two tools, as long as users have been observed in movement traces used during the training phase as background information, it is possible to successfully identify them using the new movement traces in the testing phase. With the experiments on real data sets the accuracy of the attack was confirmed, as the success rate of these tools was up to 45%. It was also found to be resilient to sanitation mechanisms. This further encourages sharing of insight over raw data in the context of customer behaviour prediction such as in this study.

A summary of the most relevant ethical considerations are given in Table 8. The proposed solutions to data ethics considerations share two common themes: public sharing of knowledge related to customer prediction and data analysis, and the involvement of the customer in the process. Of note is that the proposed solutions for the high incentives for restricted access rights and the security risks of data sharing can be somewhat contradictory, since more available data means more attack vectors. The implications of this contradiction need to be explored in further research.

Ethical issue	**Proposed solution**
Low digital literacy of customers regarding the extent and purpose of data collection.	Provide results and insight for data analysis tools for customer viewing. Easily understood and clear statement of terms, conditions and rights for the customer preferably within the app where data collection tools are used.
Ownership of data and defining the right of both the user and data collector, and the power balance.	Informed consent and legislation involving clear roles for the user and data collector in terms of rights and responsibilities.
Methods of data collection are unintelligible to the users.	Build awareness with public awareness campaigns.
Automated tools can limit the ability of individuals to self-define.	Involve the customer in the prediction process and provide tools for correction of errors.
High incentives to keep data as a sparse and hence valuable resource.	Public sharing of insights on the results of data analysis experiments to encourage a shift in perception towards common ownership.
Security risks associated with public sharing of data.	Avoid sharing raw data and focus on results and related insights.

Table 8. Summary of ethical considerations.

8. DISCUSSION

The variables most correlated with the value of the next purchase and time until next purchase were related to previous purchases, and modelling of behaviour over the last fourteen days likewise only had an effect on variables concerned with purchase behaviour. This is consistent with the good performance of RFM-derived models on prediction of customer behaviour [17][10][9]. Unfortunately, this means that since both labels are dependent on having records of previous purchase behaviour, cold-start problem stays a relevant issue. Behaviour of friends, neighbours and other in-group members can be a large influence on the behaviour of a given customer [5], so expanding a system with social graph data can provide improved results and help solve the cold-start problem, but considering the ethical aspects discussed previously might not be the most sustainable solution.

Changes in in-game static values such as rankings and currencies have very little or no correlation with the predicted time of the next purchase or the predicted value of the next purchase, which is a notable result and confirms internal knowledge that in-game behaviour does not correlate with purchase behaviour. Alternatively, this can imply that the changes in in-game static values cannot be used to accurately model in-game behaviour and more detailed data is needed. As games can have very different gameplay and goals, game metrics are different to each game type. Drachen et al. argue that many games are a combination of these base types due to the high amount of innovation present [2]. Hence, there's not one single definition of "correct" behaviour metrics and hence the question remains open. For racing games, Drachen et al. suggest track choice, vehicle choice, vehicle performance, win/loss ratio per track or vehicle, completion times and their ratio per track or player, possible upgrades, possible color scheme, hits, and average speed in different types of tracks/track shapes.

The better success of non-linear and otherwise complex predictors demonstrates the non-linearity of the determinants of purchase behaviour. The improved performance of the learning models with optimized features over the whole data set highlights the importance of feature selection methods. However, finding the balance between having sufficient data and interference from non-correlated features can be difficult, as ANN performed better on a mixed set of features linearly correlated with both labels compared to using only the features specific to the label to be predicted. Exploring non-linear correlation between features can be experimented with in the future over using black-box systems such as unsupervised feature embedding systems [10]. The improvements of 22,0% and 20,7% in purchase value and time respectively are similar to the 13% improvement shown in the prediction of number of purchases in the purchase behaviour study by Sifa et al [11].

The more novel approach of predicting the time of the next purchase is similar to predicting its value, as it depends on the frequency and other variables of previous purchases. This is an important result, as there are very little studies on prediction of purchase time. Methods for predicting customer lifetime value and churn, and the respective features correlated with them, can be applied for purchase time prediction in the future. As not all customers will purchase, training a separate model for binary prediction of purchase can be used to select the most potentially valuable customers [11][9]. Future experiments can include further statistical processing of data, such as distributions of session and data [11] and purchase data.

The amount of available processing power limited the amount of available data points to 50 000. The tools in internal use had less algorithms, so the training of models needed to be done locally. Cloud based solutions can be promising. The results of this study need to be validated in practice, as the true performance can only be measured when deployed. This can be achieved by comparing the performance of two equivalent groups in an advertisement campaign for which one uses ProGame and the other is used as a control group. The pass metric is the increase in amount of purchases and/or total value of sales, while the fail condition would be that there is no difference in performance compared to random selection or mass approach. A longitudinal study or validation with another data set can also show success.

9. CONCLUSION

The base goal of this study was to predict the next most probable purchase time as the amount of days until purchase and the value of the next purchase from event-based data using data fusion and machine learning. A data table was aggregated from a set of several separate data tables, from which a sample of 50 000 data points was analysed to find the most predictive features using their linear correlation with the labels, and further samples were used as input for three learning algorithms; Random Forest, Support Vector Machine and an Artificial Neural Network. Their performance was analyzed with RMSE for evaluation purposes. The research goals consisted of discovery of features with the most correlation with the given labels and determination of the most suitable algorithm and it's performance in predicting the labels.

The factors that correlate with purchase value and purchase time are related to previous purchase behaviour. The best predictors for the value of the next purchase were the operating system of the phone, number of cumulative purchases, the value of the previous purchase in USD, the value of the maximum purchase of the customer and the average purchase value of the customer. The best predictors for days until purchase were the activity month, whether the user was converted via an advertisement campaign or not, days since last purchase, the sum of purchases in the last 14 days, the average value of purchases made in the last 14 days, and the maximum number of purchases in the last 14 data points.

Neural networks show the best performance in predicting both labels both with and without feature engineering. Using feature engineering, Multi-Layer Perceptron shows an improvement of 22,0% for value in USD and 20,7% for days until purchase compared to a trivial baseline predictor that returns the average of the values within the data set. However, the lowered performance of MLP with more limited feature sets shows need for effective feature correlation analysis.

For ethical usage of purchase behaviour prediction and other forms of customer analytics, public sharing of results and insights from data analysis experiments and the involvement of the customer in the process is vital. These help alleviate the issues of low digital literacy of customers, debating ownership of data and the right of the customer to self-define, and offsetting the power imbalance between people with access to data analysis tools and those without. However, the contradicting goals between public ownership of data and minimizing the amount of available data for security attacks need to be explored further.

Microsegmentation has been shown to be more effective than a one-to-one or a larger segmentation system, especially when compared to aggregation systems that have the whole customer base as the unit of analysis [60]. One-to-one marketing approaches clearly dominate segmentation and aggregation approaches, but segmentation levels taken to the best granularity level are superior to them in modeling customers with little to no purchase transactions and in performance. When training a model for individual personification, the cold-start problem could be avoided by using predicted labels from a model trained on all customers as input features into a personal model. As ProGame requires a purchase for designating labels, the generic model can be utilized as the only predictor until a purchase is made and as a supporting predictor afterwards.

Future studies can experiment with clustering customers into groups, and hence create more easily understood demographics based on purchase date and amount. Joint learning of clusterizer and a corresponding classifier that assigns customers to classes has been found to outperform baseline approaches [61]. Clusters can be evaluated and prioritized with a suitable model, e.g. RFM or value ranges. Adding the predicted classification probabilities from previous time window provides past context and can also provide marginal improvement, while introducing time decay for the values of each of the data points in the last 14 data points reduces the influence of activity that is spread far apart from the current time. [29].

Future studies on predicting purchase time can involve time series feature extraction tools such as *tsfresh* developed and validated by Christ et al. in 2018 [62]. Time series analysis is the mapping of sequences of observations over time into a feature vector of specific dimensionality M [63]. For pattern recognition[64], it is efficient to map the time series as a feature vector that represents the distribution of data points, their correlation, stationarity, entropy and non-linear time series analysis [65].

10. REFERENCES

[1] Wijman T., Meehan O. & de Heij B. (2019), Global games market report.

[2] Drachen A., El-Nasr M.S. & Canossa A. (2013) Game analytics–the basics. In: Game analytics, Springer, pp. 13–40.

[3] Cai W., Leung V.C. & Hu L. (2014) A cloudlet-assisted multiplayer cloud gaming system. Mobile Networks and Applications 19, pp. 144–152.

[4] Habeeb R.A.A., Nasaruddin F., Gani A., Hashem I.A.T., Ahmed E. & Imran M. (2019) Real-time big data processing for anomaly detection: A survey. International Journal of Information Management 45, pp. 289–307.

[5] Sundsøy P., Bjelland J., Iqbal A.M., de Montjoye Y.A. et al. (2014) Big data-driven marketing: how machine learning outperforms marketers' gut-feeling. In: International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, Springer, pp. 367–374.

[6] Valero-Fernandez R., Collins D.J., Lam K., Rigby C. & Bailey J. (2017) Towards accurate predictions of customer purchasing patterns. In: 2017 IEEE International Conference on Computer and Information Technology (CIT), IEEE, pp. 157–161.

[7] Schaub F., Marella A., Kalvani P., Ur B., Pan C., Forney E. & Cranor L.F. (2016) Watching them watching me: Browser extensions' impact on user privacy awareness and concern. In: NDSS workshop on usable security.

[8] Baruh L. & Popescu M. (2017) Big data analytics and the limits of privacy self-management. New media & society 19, pp. 579–596.

[9] Martínez A., Schmuck C., Pereverzyev Jr S., Pirker C. & Haltmeier M. (2018) A machine learning framework for customer purchase prediction in the non-contractual setting. European Journal of Operational Research .

[10] Chamberlain B.P., Cardoso A., Liu C.H., Pagliari R. & Deisenroth M.P. (2017) Customer lifetime value prediction using embeddings. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1753–1762.

[11] Sifa R., Hadiji F., Runge J., Drachen A., Kersting K. & Bauckhage C. (2015) Predicting purchase decisions in mobile free-to-play games. In: Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference.

[12] Gupta R. & Pathak C. (2014) A machine learning framework for predicting purchase by online customers based on dynamic pricing. Procedia Computer Science 36, pp. 599–605.

[13] Russom P. et al. (2011) Big data analytics. TDWI best practices report, fourth quarter 19, pp. 1–34.

[14] Germann F., Lilien G.L., Fiedler L. & Kraus M. (2014) Do retailers benefit from deploying customer analytics? Journal of Retailing 90, pp. 587–593.

[15] Brynjolfsson E., Hitt L.M. & Kim H.H. (2011) Strength in numbers: How does data-driven decisionmaking affect firm performance? Available at SSRN 1819486 .

[16] Hughes A.M. (1996) Boosting response with RFM. Marketing Tools , pp. 4–8.

[17] Tsai C.Y. & Chiu C.C. (2004) A purchase-based market segmentation methodology. Expert Systems with Applications 27, pp. 265–276.

[18] Cumby C., Fano A., Ghani R. & Krema M. (2004) Predicting customer shopping lists from point-of-sale purchase data. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 402–409.

[19] Fader P.S., Hardie B.G. & Lee K.L. (2005) Rfm and clv: Using iso-value curves for customer base analysis. Journal of marketing research 42, pp. 415–430.

[20] Zhang Y. & Pennacchiotti M. (2013) Predicting purchase behaviors from social media. In: Proceedings of the 22nd international conference on World Wide Web, ACM, pp. 1521–1532.

[21] Medler B., John M. & Lane J. (2011) Data cracker: developing a visual game analytic tool for analyzing online gameplay. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp. 2365–2374.

[22] Hullett K., Nagappan N., Schuh E. & Hopson J. (2011) Data analytics for game development: Nier track. In: 2011 33rd International Conference on Software Engineering (ICSE), IEEE, pp. 940–943.

[23] Chen P.P., Guitart A., del Río A.F. & Periáñez Á. (2018) Customer lifetime value in video games using deep learning and parametric models. In: 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp. 2134–2140.

[24] Burelli P. (2019) Predicting customer lifetime value in free-to-play games. Data Analytics Applications in Gaming and Entertainment , p. 79.

[25] Zaslavsky A., Jayaraman P.P. & Krishnaswamy S. (2013) Sharelikescrowd: Mobile analytics for participatory sensing and crowd-sourcing applications. In: 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), IEEE, pp. 128–135.

[26] Do T.M.T. & Gatica-Perez D. (2014) Where and what: Using smartphones to predict next locations and applications in daily life. Pervasive and Mobile Computing 12, pp. 79–91.

[27] He Y., Yu F.R., Zhao N., Yin H., Yao H. & Qiu R.C. (2016) Big data analytics in mobile cellular networks. IEEE access 4, pp. 1985–1996.

[28] Peltonen E., Lagerspetz E., Hamberg J., Mehrotra A., Musolesi M., Nurmi P. & Tarkoma S. (2018) The hidden image of mobile apps: Geographic, demographic, and cultural factors in mobile usage. In: Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 1–12.

[29] Krishnan N.C. & Cook D.J. (2014) Activity recognition on streaming sensor data. Pervasive and mobile computing 10, pp. 138–154.

[30] Murphy K.P. (2012) Machine learning: a probabilistic perspective. MIT press.

[31] Shalev-Shwartz S. & Ben-David S. (2014) Understanding machine learning: From theory to algorithms. Cambridge university press.

[32] Lexico.com (2020), Meaning of overfitting in english. `https://www.lexico.com/definition/overfitting`. Accessed: 2020-03-10.

[33] Breiman L. (2001) Random forests. Machine learning 45, pp. 5–32.

[34] Freund Y., Schapire R.E. et al. (1996) Experiments with a new boosting algorithm. In: icml, vol. 96, Citeseer, vol. 96, pp. 148–156.

[35] Breiman L. & Cutler A. (2013), Random forests. `https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`. Accessed: 2020-02-06.

[36] Geurts P., Ernst D. & Wehenkel L. (2006) Extremely randomized trees. Machine learning 63, pp. 3–42.

[37] Goel E., Abhilasha E., Goel E. & Abhilasha E. (2017) Random forest: A review. International Journal of Advanced Research in Computer Science and Software Engineering 7.

[38] Farnaaz N. & Jabbar M. (2016) Random forest modeling for network intrusion detection system. Procedia Computer Science 89, pp. 213–217.

[39] Cortes C. & Vapnik V. (1995) Support-vector networks. Machine learning 20, pp. 273–297.

[40] Chang C.C. & Lin C.J. (2011) Libsvm: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2, p. 27.

[41] Vapnik V. & Vapnik V. (1998) Statistical learning theory wiley. New York , pp. 156–160.

[42] Thanh Noi P. & Kappas M. (2018) Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. Sensors 18, p. 18.

[43] Hinton G.E. (1990) Connectionist learning procedures. In: Machine learning, Elsevier, pp. 555–610.

[44] He K., Zhang X., Ren S. & Sun J. (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp. 1026–1034.

[45] Glorot X. & Bengio Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256.

[46] Kingma D.P. & Ba J. (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

[47] Ahmad M.W., Mourshed M. & Rezgui Y. (2017) Trees vs neurons: Comparison between random forest and ann for high-resolution prediction of building energy consumption. Energy and Buildings 147, pp. 77–89.

[48] Siroky D.S. et al. (2009) Navigating random forests and related advances in algorithmic modeling. Statistics Surveys 3, pp. 147–163.

[49] Alkin M.C., Christie C.A. & Vo A.T. (2012) Evaluation theory. Evaluation Roots: A Wider Perspective of Theorists' Views and Influences , p. 386.

[50] Armstrong J.S. & Collopy F. (1992) Error measures for generalizing about forecasting methods: Empirical comparisons. International journal of forecasting 8, pp. 69–80.

[51] Kanji G.K. (2006) 100 statistical tests. Sage.

[52] McDonald J.H. (2014), Handbook of biological statistics. `http://www.biostathandbook.com/index.html`. Accessed: 2020-02-11.

[53] Hyndman R.J. & Koehler A.B. (2006) Another look at measures of forecast accuracy. International journal of forecasting 22, pp. 679–688.

[54] Bewick V., Cheek L. & Ball J. (2003) Statistics review 7: Correlation and regression. Critical care 7, p. 451.

[55] Lowry R. (2014) Concepts and applications of inferential statistics .

[56] Zwitter A. (2014) Big data ethics. Big Data & Society 1, p. 2053951714559253.

[57] Mittelstadt B.D. & Floridi L. (2016) The ethics of big data: current and foreseeable issues in biomedical contexts. Science and engineering ethics 22, pp. 303–341.

[58] Taylor L. (2016) The ethics of big data as a public good: which public? whose good? Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374, p. 20160126.

[59] Gambs S., Killijian M.O. & del Prado Cortez M.N. (2014) De-anonymization attack on geolocated data. Journal of Computer and System Sciences 80, pp. 1597–1614.

[60] Jiang T. & Tuzhilin A. (2006) Segmenting customers from population to individuals: Does 1-to-1 keep your customers forever? IEEE Transactions on knowledge and data engineering 18, pp. 1297–1311.

[61] Haider P., Chiarandini L. & Brefeld U. (2012) Discriminative clustering for market segmentation. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 417–425.

[62] Christ M., Braun N., Neuffer J. & Kempa-Liehr A.W. (2018) Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). Neurocomputing 307, pp. 72–77.

[63] Box G.E., Jenkins G.M., Reinsel G.C. & Ljung G.M. (2015) Time series analysis: forecasting and control. John Wiley & Sons.

[64] Bishop C.M. (2006) Pattern recognition and machine learning. springer.

[65] Fulcher B.D. (2018) Feature-based time-series analysis. In: Feature engineering for machine learning and data analytics, CRC Press, pp. 87–116.

11. APPENDICES

Appendix 1 Feature Tables

General		
Feature name	**Feature type**	**Description**
platform	string	Which operating system the player is using (iOS or Android)
country	string	Current location as country the player is currently
retained_days	integer	Amount of days since first downloading the game
purchases	integer	Amount of purchases during the current day
paid_user	Boolean	Whether the user was converted with an advertisement campaign (true) or found the game by some other manner (false)
cum_purchases	integer	Total amount of lifetime purchases by the user before the current day calculated as $purchase\_number - 1$, where the purchase number is derived from the row number of the future purchase in the list of all purchases ordered by time stamp
activity_day_of_week	integer	Day of the week of the current day, represented as a number from one to seven
activity_day_of_month	integer	Day of the month of the current day, represented as a number from one to 31
activity_month	integer	Month of the year of the current day, represented as a number from one to 12
activity_day_of_year	integer	Day of the year of the current day, represented as a number from one to 365

Table 9. Features in the General set.

Daily		
Feature name	**Feature type**	**Description**
purchases	integer	Amount of purchases during the current day
int_ads_watched	integer	Number of ads watched during the current day
rew_ads_watched	integer	Number of ads that give in-game rewards watched during the current day
daily_min_rank	integer	Minimum in-game rank during current day
daily_min_season_rank	integer	Minimum in-game seasonal rank during current day
net_revenue	float	Total revenue for the current day
daily_coin_state	integer	Amount of coins the user has in the current day
daily_gem_state	integer	Amount of gems the user has in the current day
daily_delta_rank	integer	Change of rank during the current day, calculated from the difference of daily maximum rank and daily minimum rank
daily_delta_season_rank	integer	Change of seasonal rank during the current day, calculated from the difference of daily maximum seasonal rank and daily minimum seasonal rank
delta_min_rank	integer	Change of minimum rank compared to the previous day
delta_min_season_rank	integer	Change of minimum seasonal rank compared to the previous day
delta_coin_state	integer	Change in the amount of in-game coins compared to the previous day
delta_gem_state	integer	Change in the number of in-game gems compared to the previous day
total_sessions	integer	Number of usage sessions in the current day
sess_length_seconds	float	Total sum length in seconds of the sessions during the current day

Table 10. Features in the Daily set.

RFM		
Feature name	**Feature type**	**Description**
days_since_last_purchase	integer	Number of days since the last registered purchase: if there are no previous purchases, the value of days_retained is used
lead_value_in_usd	float	Value of the previous purchase: If there are no previous purchases, the value is filled with 0
purchase_frequency	float	Lifetime purchases per day, calculated from cum_purchases/retained_days: if there are no previous purchases, the value is filled with 0
max_purchase_value	float	Maximum lifetime purchase value of the customer in USD: if there are no previous purchases, the value is filled with 0
average_purchase_value	float	Average of the lifetime purchase value in USD for the customer: if there are no previous purchases, the value is filled with 0

Table 11. Features in the RFM set.

14-Point		
Feature name	**Feature type**	**Description**
purchases_valid_sum	integer	Sum of the number of purchases within the last 14 data points
int_ads_sum	integer	Sum of the number of ads watched within the last 14 data points
rew_ads_watched_sum	integer	Sum of the number of ads that give in-game rewards watched within the last 14 data points
delta_min_rank_sum	integer	Sum of the change of minimum rank between days within the last 14 data points
delta_min_season_rank_sum	integer	Sum of the daily change of minimum seasonal rank between days within the last 14 data points
delta_coin_state_sum	integer	Sum of the change in the amount of in-game coins between days within the last 14 data points
delta_gem_state_sum	integer	Sum of the change in the number of in-game gems between days within the last 14 data points
total_sessions_sum	integer	Sum of the number of usage sessions within the last 14 data points
sess_length_seconds_sum	float	Sum of the length of usage sessions last 14 data points
purchases_valid_avg	integer	Average number of purchases in a day within the last 14 data points
int_ads_avg	integer	Average number of ads watched in a day within the last 14 data points
rew_ads_watched_avg	integer	Average number of ads that give in-game rewards watched in a day within the last 14 data points
delta_min_rank_avg	integer	Average change of minimum rank between days within the last 14 data points
delta_min_season_rank_avg	integer	Average change of minimum seasonal rank between days within the last 14 data points
delta_coin_state_avg	integer	Average change in the amount of in-game coins between days within the last 14 data points
delta_gem_state_avg	integer	Average change in the number of in-game gems between days within the last 14 data points
total_sessions_avg	integer	Average number of usage sessions in a day within the last 14 data points

sess_length_seconds_avg	float	Average total length of usage sessions in a day within the last 14 data points
purchases_valid_max	integer	The largest number of purchases in a day within the last 14 data points
int_ads_max	integer	The largest number of ads watched in a day within the last 14 data points
rew_ads_watched_max	integer	The largest number of ads that give in-game rewards watched in a day within the last 14 data points
delta_min_rank_max	integer	The largest change of minimum rank between days within the last 14 data points
delta_min_season_rank_max	integer	The largest change of minimum seasonal rank between days within the last 14 data points
delta_coin_state_max	integer	The largest change in the amount of in-game coins between days within the last 14 data points
delta_gem_state_max	integer	The largest change in the number of in-game gems between days within the last 14 data points
total_sessions_max	integer	The largest number of usage sessions in a day within the last 14 data points
sess_length_seconds_max	float	The largest total length of usage sessions in a day within the last 14 data points

Table 12. Features in the 14-Point set.