INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 4, PP. 1081-1087
Manuscript received September 13, 2024; revised October, 2024. doi: 10.24425/ijet.2024.152510

Prediction of the mobile game players' payments-related retention from the Big Data perspective

Piotr Bilski, Rafał Łabędzki, and Adrian Bilski

***Abstract*—The paper presents the methodology for evaluating mobile game players retention, which is the basis for generating economic income for its creators. The system for collecting and processing the in-game data (events related to the players' actions) exploiting the Big Data cloud platform is described. The player profile with the crucial features allowing for the retention analysis is introduced. Datasets generated for the My Spa Resort mobile game by CherryPick company are described. The retention prediction approach based on the similarity estimation between the analyzed and already inactive players is presented. Results of the prediction using the k Nearest Neighbors (kNN) classifier are discussed.**

***Keywords*—retention; data science; prediction; mobile game**

I. Introduction

THE computer games industry has become a significant branch of economy in the XXI century. Through the past forty years the domain has significantly changed its business model, from the Software as a Product (i.e. tangible, physical carrier with the program installer) to the Software as a Service (the user pays for the access to the game. Introduction of mobile games, besides the typical approach (where the user pays for the ability of downloading the product, especially practiced in Apple) has pushed it even further, to the freemium (also known as Free-to-Play) model. Here the user is able to run the game for free, but in such a situation the program has the limited functionality, which can be extended after the payment is made. This includes the additional equipment, extended capabilities of the protagonist and so on. Additionally, during the gameplay the user is shown commercials, which is the additional source of revenue for the producer. Therefore every creator of the game is interested in maximizing the number of users playing the game, their gameplay duration, and the number of transactions during the game.

The latter is the most challenging, as the user has to spend the actual money, which is the most difficult from the psychological perspective. Therefore the game producer may be interested in learning the ways to encourage the clients to spend more money on the specific product. Unfortunately, there are no strict rules binding the game itself and the user's economical decisions (which is also a problem in the "traditional" model of the game distribution). Also, these details can be reasoned only based on the interaction with the game itself. This calls for the introduction of data analysis methods for extracting information crucial for the game reception. This includes selection of the game genre, prediction of the player's retention, and the identification users who spend money to play.

The aim of the paper is to present the adaptive module for predicting the players' actions related with spending money on the game based on their past actions (i.e. the gaming style). For the prediction, the Artificial Intelligence (AI) algorithms are used to maximize the prediction accuracy. They are used to predict the money spending behavior of the particular player, especially how many payments can be estimated and how long he/she will stay in the game. For this purpose feature vectors must be extracted from the "raw" data collected in the Big Data form on the server. The prediction itself is performed using the k Nearest Neighbors classifier.

The outline of the paper is as follows. In Section II State of the Art is presented, to show the current advancement of the AI-based approaches to profile the players' behavior from the economic perspective. Section III contains the description of the proposed intelligent module that is implemented in the CherryPick Games company operations. Section IV describes the data set used for training the prediction module and the process of collecting parameters used for analysis. Section V covers the proposed prediction algorithm, based on the machine learning, while in Section VI results are presented. The paper is concluded with the summary containing information about the possible prospects of the presented approach.

II. State of the Art

Currently multiple approaches are implemented in the games industry to track and profile users' actions. Two groups of methods are distinguished here: using psychological or sociological data, or based on their in-game operations. The main object of interest are mobile (Free to Play) and on-line games (either accessible through the web browser, or as the

This work was supported by National Centre of Research and Development, project number POIR.01.01.01-00-0927/17, acronym "Cherrylytics."
Piotr Bilski is with Warsaw University of Technology, Poland (e-mail: piotr.bilski@pw.edu.pl).
Rafał Łabędzki is with SGH Warsaw School of Economics, Poland (e-mail: rlabed@sgh.waw.pl).
Adrian Bilski is with Warsaw University of Life Sciences, Poland (e-mail: adrian_bilski@sggw.edu.pl).

desktop application), such as MMORPG genre. Their impact on the gaming industry is growing, leading to the greater interest of developers and major studios (with Sony's Concord being the most recent example [1]).

The first approach is to identify the game preferred by the user, which is the guide for the creators about which genre might be the most interesting (and therefore economically rewarding) [2]. Here the Support Vector Machines (SVM) and Random Forest (RF) were used to process socio-demographic data. In [3] the factor analysis was used to classify the Turkish players according to the Bartle's taxonomy. Similarly, in [4] personal traits of players are investigated to adjust games' mechanics to them. In [5] the generic framework is proposed to combine demographic and behavioral features of players to group them into clusters. In all these attempts no information about the practical applications of the extracted knowledge is given. Only the general prospects for the game developers and creators are described.

The second problem to be solved is to identify and predict the players retention, i.e. their influx and in-game duration. Such solutions are already implemented in practice [6], which proves the problem is already mature. In [7] the Far Cry 4 data are used to build the model combining the survival analysis and weapons usage with the time spend on the gameplay. The online network-oriented games (based on the World of Tanks example) were analyzed in [8]. Here activity level, gaming performance (but also social relations between players) were considered to build the machine learning based retention predictor. In [9] the player's retention is maximized by applying the adaptive artificial opponent in the Scrabble-type on-line game. It was determined that adjusting the difficulty level is crucial for delaying the user quitting the game. The interesting approach for the retention-oriented on-line game development framework is discussed in [10]. Here the reinforcement learning (i.e. Q-learning) is applied to optimize the design of the game levels on the example of World of Warcraft and EVE. The more general approach is applied in [11], where five mobile games are considered to identify three types of retention.

The problem of in-game purchases is considered in [12]. Here the Mixed Effects Cox Regression is used to predict the churn of the Leaguer of Legends players. In [13] the mobile gaming retention is considered (analyzed data are extracted from the Jelly Splash game). Daily activity is monitored and provided to the input of the Logistic Regression (LR), SVM and RF classifiers (also, operating as the ensemble). In [14] the in-game session durations, gained skills and time between the sessions are statistically processed to predict the purchase decisions of mobile players of the puzzle game.

The presented works show the importance of the analysis of users of the on-line game for from the developers' standpoint. Modern AI-based classifiers are used in most cases, especially RF, SVM and ANN, operating on the datasets with the predefined features. The deficiency of such applied approaches is the relatively low level of details regarding the information about the players (disregarding their actual behavior during the game). Also, the practical aspects of the periodical updates of knowledge are not considered. This justifies constructing the complete system for providing the thorough data acquisition and analysis of gamers' behavior, as presented in the following paper.

III. Data Collection and Analysis System

To approach the problem of predicting the retention, the data collection and analysis system must be developed. Contrary to [3]-[14], the presented solution is complex in terms it considers recording "raw" data from the game, extracting information to form feature vectors and use them for the predictions. This allows for selecting and constructing unique features to find the best one for the selected task.

This section presents the architecture of the system, focusing on the Big Data paradigm and defines the prediction tasks that can be resolved using the presented approach.

A. System structure

The architecture presented in Fig. 1 is generic, i.e. on the high level of abstraction it can be used to analyze any on-line game where the player's advancement is divided into levels of experience. The detailed implementation requires selecting the genre-specific features and analyze their usefulness to the prediction task.

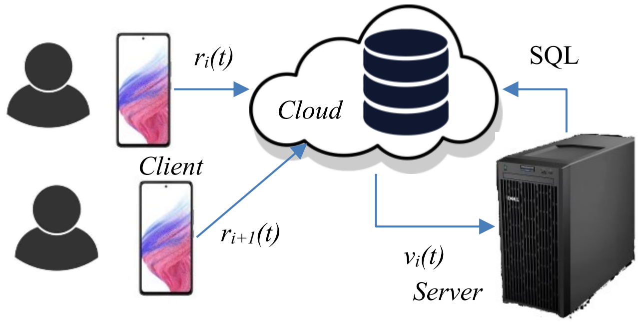

Fig. 1. Structure of the data collection and analysis system

The system is decomposed into three parts. The first one is the game engine (run on the *Client*), which fragment is aimed at sending the "raw" features related with the user's actions (forming the set $\boldsymbol{R}$) to the cloud-located storage (*Cloud*). The latter is a large database, storing information from all active players. Its internal structure must allow for keeping data of the TB or EB size. It is constantly updated to have the most recent information about the players community. The third element is the computational *Server*, which queries the database through the periodic SQL requests for the features usable by the AI-based predictor. The dataset for training $\boldsymbol{V}$ is constructed from all available data, which means that its size will be increasing with time. The process of the knowledge extraction (through machine learning) is represented by the following process:

$$K = f(\boldsymbol{V}) = f\big(g(\boldsymbol{R})\big) \quad (1)$$

where *g()* is the SQL-based function extracting features from the set $\boldsymbol{R}$ (stored in the cloud) and producing the feature set $\boldsymbol{V}$. The latter is further processed by the function *f()*, responsible for producing knowledge *K*. The structure of the latter depends on the applied AI-based methodology.

The system was constructed based on the Google BigQuery engine, accessible through both API and web-based GUI (the former is essential for embedding the data acquisition into the processing application on the *Server*). The analytic module was created using Python 3 language and sci-kit learn package for implementing the prediction algorithm. The target for analysis was the My Spa Resort mobile game (Fig. 1), developed by the CherryPick company. Results presented in Section VI are then specific for the particular genre. Using the system for another game type will require creating the individual features.

The game is a relatively simple economic-strategy game that that integrates farming, building, and resort management mechanics. Users are supposed to expand on their facility, by harvesting in-game resources (virtual currency), processing them into products required for the spa operation, and managing various treatments for guests. The game allows for customization of spa facilities, interaction with staff and clients, and social features like visiting and trading with other players.. Through these activities, players build and maintain their resorts, aiming to optimize resources and client satisfaction while fostering social connections with other players. The game is available for Android and iOS system, available for most of the contemporary smartphones, though it can also be run on the emulators (such as Bluestack [15], which was used for the tests).

Fig. 2. Screenshot from the My Spa Resort game

Actions performed by the user (such as building the specific structure or making a selected purchase) are recorded in the BigQuery database and cab be further processed.

B. Big Data feature extraction

The purpose of the data storage is to deliver all available information from the game engine for processing. As the number of players changes dynamically, it is essential to use all available data, which requires waiting some time after the game release before starting the analytics. The presented framework at this stage is aimed at producing the relevant features describing behavior of the player through SQL queries. The Google BigQuery delivers the ANSI-compliant dialect of this language, allowing for the DQL-based data extraction. The data collection for the processing module (performed on *Server* in Fig. 1) must be done periodically, considering the fact that the online game engine is updated, identified in the database through the version number (varying game versions offer different functionalities, which may influence the collected data). The features extracted from the cloud base are saved in the csv files on the server. The core events, considered by the SQL statements to construct the training data are in Table I.

The database stores information about each event that was generated by the game engine (such as making purchase using physical currency or building the particular structure). The SQL query allows for extracting them by aggregations regarding the user_id, user_level and app_version fields, i.e. combining the

TABLE I
DATA EXTRACTED FROM THE BIGQUERY ENGINE

Field	Interpretation
user_id	Unique identifier of the player
user_level	Level, at which the event was recorded
app_version	Version of the game
purchase_commenced	Purchase made by the player using physical currency
building_constructed	Construction of the specific building in the game
event_timestamp	Time indicator of the event
experience_points	Number of possessed experience points (related with the experience level)

events into a single record, representing operations and actions for the particular user on the specific game level. This allows for building the player profile, for which the prediction can be made (see Section III.C). The idea of the players profiling is based on the statistical measures, leading to the simple representation of their actions.

C. Prediction tasks definition

Retention is the core concept in the economy of the gaming industry seen as the main source of income for the producer of the "Free to Play" game. Having access to the detailed information about the player's behavior, it is important to correctly define the space of input features, being relevant for the prediction task, and the actual problem to solve. The *i-th* player profile $\boldsymbol{P}_i=\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \ldots, \boldsymbol{p}_n\}$ is represented by records $\boldsymbol{p}_j$ describing his/her achievements at the particular levels of advancement (Fig. 3). Each record is a vector of the following features:

$$\boldsymbol{p}_j = \{l, n_e, n_p, n_b, x_p, t_l\} \tag{2}$$

where l is the advancement level, for which the statistics are extracted, n_e is the number of events generated by the player at the level l (integer number from the set Lk, which cardinality depends on the particular game), n_p is the number of purchases (using Real-World currency), n_b is the number of buildings constructed, x_p is the amount of the experience points obtained by the player at the level l, and t_j is the in-game duration of the level (the difference in the timestamps – see table I – between the first and the last event fort he player at the level l).

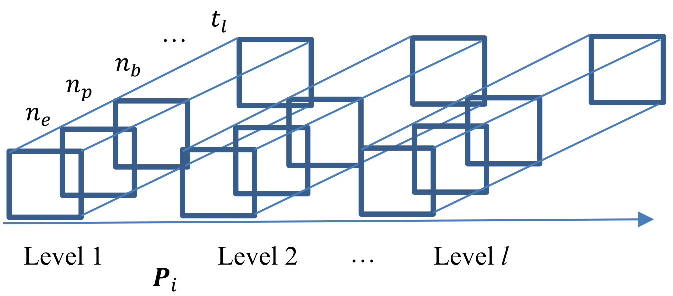

Fig. 3. Representation of the player $\boldsymbol{P}_i$.as the set of aggregated feature vectors

The key concept here is the “active player”, i.e. the user that still participates in the game, as he/she may be the source of income (either by spending actual money or being the target for commercials). To define the active user, timestamps stored for him/her in the *Cloud* _module must be confronted against the current date/time t_{rt} (i.e. the timestamp of the query execution). The player is labeled as “active” (its activity function a is given the value of 1) if the time of its last interaction within the game is not older than the predefined threshold θ (here defined as two weeks):

$$a(\boldsymbol{P}_i) = \begin{cases} 1 & \left| t_{rt} - \max t_l \right| < \theta \\ 0 & otherwise \end{cases} \qquad (3)$$

The second important concept is the “spender”, i.e. the player $\boldsymbol{P}_i$ who at least once spent actual money, for example to buy virtual resources. The requirement to assign the player to the set of spenders $\boldsymbol{S}$ is as follows:

$$\boldsymbol{P}_i \in \boldsymbol{S} \Leftrightarrow \exists l \in L{:}\, n_p(l) > 0 \qquad (4)$$

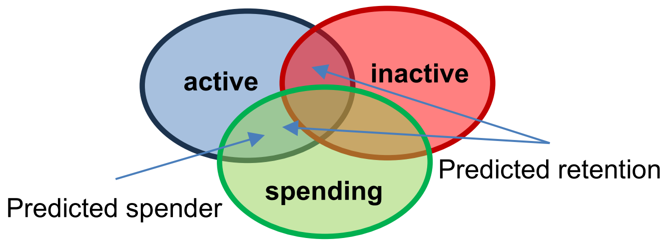

Fig. 4. Illustration of the retention search problem

The retention is then explored on the intersection of these groups of players (treated separately), as presented in Fig. 4. The general idea is the analyze the current player and find the most similar inactive ones (with $a(\boldsymbol{P}_i)=0$) that have the specific profiles (the particular game duration or the number of payments). This way it is possible to predict the future behavior of the selected active players. The similarity is calculated by the comparison between the feature vectors of the corresponding users belonging to two categories: “active” and “inactive” (3) Two tasks are defined as follows:

(1) find the most similar inactive players and define the most probable maximal level to be reached

(2) find the most similar inactive players and define the most probable number of payments

The second task in practice is executed on much smaller subset (as the number of spenders is usually small compared to the whole population). It is possible to define subtasks here, for instance distinguishing between the “one-time spenders” and “frequent spenders”. Because of the limited amount of data, this paper focuses only on the first task, i.e. prediction of the most probable retention of the active player based on the similarity evaluation (up to its current level) with all inactive players (Fig .5). Due to the large number of players’ they can be clustered into the homogenous groups (like in [16]).

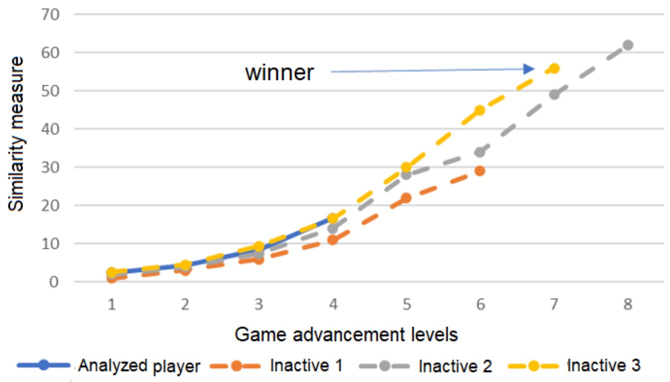

Fig. 5. Result of the prediction task

IV. Training Data Analysis

This section presents the data collected from the My Spa Resort game forming players profiles as in Section III.C. This analysis may be used by the game developers to change the game structure or the interaction model with the player (for instance, by decreasing the difficulty level at some point, as this may be the main cause of users quitting). Two main aspects of the players profiles are considered here: the maximum advancement level reached (correlated with the gameplay duration) and the number of actual currency payments.

The players were divided into two categories based on their activity (3), while money spending feature (4) is treated as one of the characteristics at the particular advancement level. Inactive players are used for predictions, while the active ones are the currently analyzed ones. The dataset collected from BigQuery taken for experiments includes a total of 69101 players. For 11313 of them the function $a(\boldsymbol{P}_i)=1$ (active), while for the remaining 57788, $a(\boldsymbol{P}_i)=1$. The searched information is the maximum level at which the player will stop plying. It is assumed that the player operations, defined by the number of events, purchases, and completed buildings at each level of gameplay, remain similar for players who quit the game at the same level. Statistical distribution of events at the particular levels is in Fig. 6 for players clustered into four groups, depending on the maximal level reached.

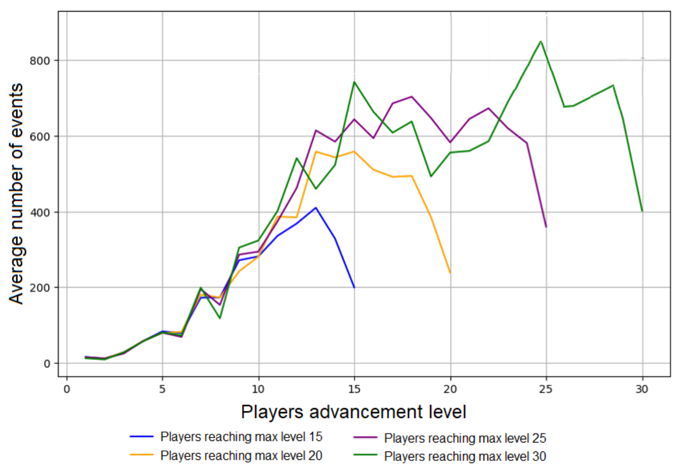

Fig. 6. Illustration of the events at the particular level

The numbers of events completed by players belonging to four groups follows a similar rising trend up to the level No. 10. However, from level 10 onwards, these groups start to differentiate significantly. Players who eventually reach level No. 30 show a continued increase in the event generation, peaking around level 20 and maintaining higher activity ratios compared to other groups. In contrast, players in the level 15 group begin to diverge sharply, showing a clear drop in event participation after level 15, while the level 20 and level 25 groups fluctuate around mid-game but show no sustained increase, peaking around their respective max levels before eventually decreasing.

The purchasing behavior of players remains relatively similar across all groups until level 10, where differentiation begins to emerge. Players finally reaching level 30 demonstrate a sharp increase in purchases starting around level 15, showing a clear divergence from other groups. By contrast, the level 15 and level 20 groups show minimal purchasing activity throughout, with small variation. The level 25 group shows moderate buying/spending activity, but it still lags behind the more substantial purchasing behavior of the level 30 group (the most attractive for the game creators). The sharp distinction in purchasing behavior between the level 30 group and the others becomes most pronounced from level 15 onwards.

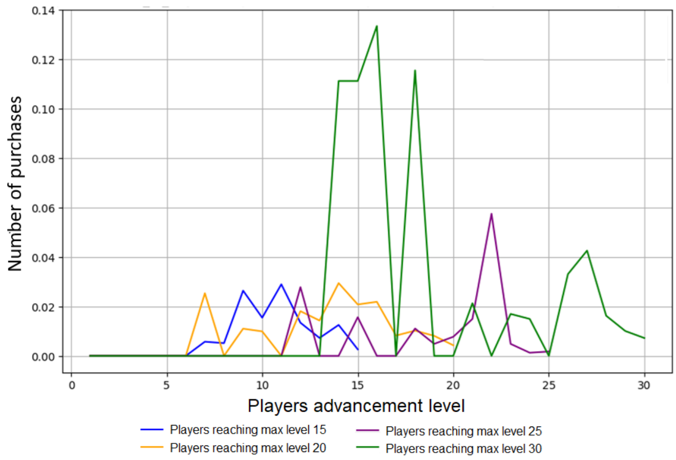

Fig. 7. Comparison of Number of Purchases (Buys)

Regarding the building completions, players exhibit similar behavior across all groups until about level 5, at which point the level 30 and level 25 groups begin to increase the activity in terms of building completions. The groups further differentiate from the level 10 onwards, while the level 30 group consistently completes more buildings than all other players. The level 15 and level 20 groups show declining building activity after their reaching one of the earlier level, while the level 30 group maintains a more consistent pace in completing buildings across the later stages of the game. This suggests that building completion becomes a more sustained activity for higher-level players, while lower-level players drop off earlier in their engagement with this aspect of the game.

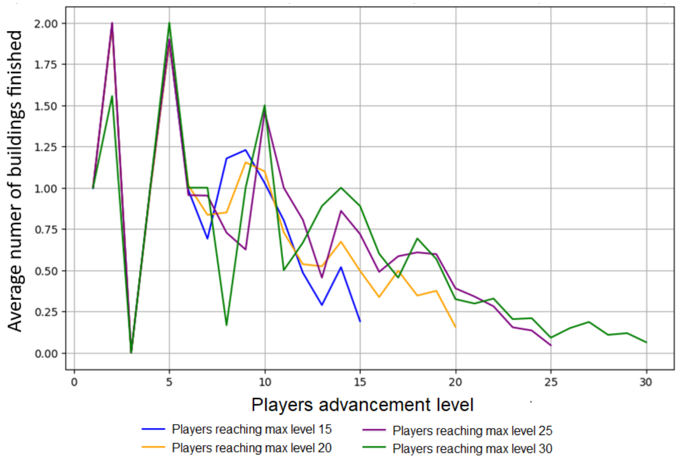

Fig. 8. Comparison of Number of Buildings Finished

This analysis confirms the hypothesis that the player journey—measured by the number of events, purchases, and buildings completed at each level of gameplay—shows consistent patterns for players who stop playing at the same level. Fig. 6-8 reveal that players behavior regarding the two crucial aspects (constructing buildings and spending money) begin to differentiate as players approach their respective maximum levels, with clear distinctions in engagement emerging from levels 10 to 15. Based on the analysis it is suggested that the player's behavior can be reliably predicted based on his/her actions. The next step is the prediction algorithm to forecast player retention and identify potential churn points in future active players.

V. Prediction Algorithm

This section presents the prediction approach to identify the most probable retention of the player (evaluated as the last level l reached before quitting the game). The approach distinguishes between the active and inactive players, attempting to identify the retention of the former based on the past information about the latter.

The players are represented by their profiles (i.e. feature vectors (2)) with the maximum level reached being the crucial information used for the comparison. To emulate the "inactivity", simultaneously maintaining the proper amount of data to process, the players reaching level 15 were analyzed.

A. Data preprocessing

The available dataset (constructed through the subsequent, periodical SQL queries) was first filtered regarding the advancement level of interest. Players below level 10 reflect a very similar behavior, as the gameplay is in the early stages of advancement, and reaching these levels takes relatively short time and requires small number of experience points to collect.

For each player the feature vectors were created (with n_b and n_p being the most prominent ones). Additionally, the following statistical features were calculated:

- Rate of Change (r_{oc}), i.e. the intensity of events (users actions) occurrences (real-world money purchases and buildings constructions) between the subsequent levels (i.e. the difference between levels 1 and 2, 2 and 3 and so on).

- variance *var* and standard deviation σ, representing variability across player metrics was calculated to capture the consistency of a player's behavior.
- Interaction between features, allowing for determining the relationships between the selected features (for instance the number of purchases normalized to the overall number of all events generated for the player).

B. Retention prediction

The classifier used for prediction was the k Nearest Neighbors (kNN), selected for its ability to find similarities between feature vectors represented as points in the *n*-dimensional space [17]. The produced category is the level, at which the player should finish the game. Experiments considered inactive players reaching at least the level l_{max} identical to the current level of the active player. To evaluate the accuracy of the approach, part of the inactive players were treated as the active ones and predictions generated for them were confronted against their actual retention (knowing, at which point they have quit the game). The data set was partitioned in the ratio 9:1 to maintain the cross-validation requirements.

The hyperparameter of the kNN classifier is the number of neighbors voting for the predicted final (maximal) level. This value must be selected according to the number of categories produced. In the presented case the number of levels to produce is 15, so the number of neighbors should reflect that. In the experiments the value of k was between 5 and 30. The evaluation of the classifier consisted in calculating the similarity between the predicted and actual maximum number of levels reached by the player. The Mean Absolute Error (MAE), R-squared (R^2), and Root Mean Squared Error (RMSE) [18] metrics were calculated for the whole testing set, obtaining the average accuracy for all considered players.

VI. Prediction Evaluation Results

The total number of inactive players who reached at least level 15 was 11367, of which 10231 were used for training and 1136 for testing. Fig. 9 shows the accuracy for the subsequent number of neighbors.

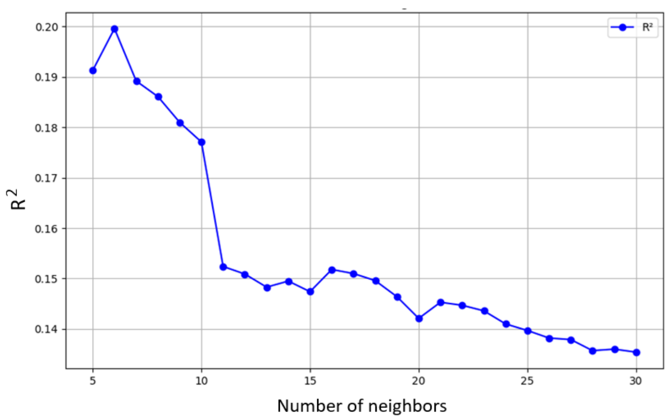

Fig. 9. Prediction accuracy depending on the number of neighbors

The best results were obtained for $k=6$, reaching the values of $R^2=0.1996$, MAE=3.2878, and RMSE=7.3423.. The R^2 values for the range of tested neighbors ranged from 0.1354 to 0.1996, indicating that the model explains 13-20% of the variance in the maximum level reached by players. The average mistake made by the classifier is 3 levels, which was considered by the game developers as a satisfactory result , which may be further analyzed to propose in-game actions to extend the players retention.

Conclusion

The paper presented the methodology for predicting the My Spa Resort mobile game players retention. This is the crucial aspect of the Free to Play games, as their developers do not benefit from installing and downloading the game, but rather from the in-game activity of the users. The presented complete framework is aimed at collecting the game-related data in the cloud-based database and using them to predict the maximal level of the selected players based on their behavior so far. The key concept here is the "activity", which allows for distinguishing users who are still in the game from the ones that have already quit. The kNN classifier was used for the prediction task. The optimal number of neighbors (considering selected similarity measures) has been evaluated.

Future research will be aimed at employing additional classifiers for prediction. Also, the analysis should be conducted regarding the most informative features that can be extracted from the raw data stored in the database.

Acknowledgements

This text summarizes a part of research within a EU co-funded project POIR.01.01.01-00-092 7/17 - Cherrylytics.

References

[1] "An important update on Concord," [Online] https://blog.playstation.com/2024/09/03/an-important-update-on-concord/.

[2] M. Gosztonyi, "Who are the gamers? Profiling adult gamers using machine learning approaches," Telematics and Informatics Reports, Vol. 11, 2023, https://doi.org/10.1016/j.teler.2023.100074.

[3] F. Erümit, S., Şılbır, L., Erümit, A. K., & Karal, H. "Determination of Player Types according to Digital Game Playing Preferences: Scale Development and Validation Study," International Journal of Human–Computer Interaction, Vol. 37, No. 11, pp. 991–1002, 2020, https://doi.org/10.1080/10447318.2020.1861765.

[4] L. S. Ferro, S. P. Walz, and S. Greuter. "Towards personalised, gamified systems: an investigation into game design, personality and player typologies." Proc. of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death (IE '13), ACM, New York, NY, USA, https://doi.org/10.1145/2513002.2513024.

[5] J. Williams, R. Bendell, S. Fiore, and F. Jentsch, "Towards a Conceptual Framework of Comprehensive Video Game Player Profiles: Player Models, Mental Models, and Behavior Models," Proc. Human Factors and Ergonomics Society Annual Meeting,. 65, pp. 807-811. https://doi.org/10.1177/1071181321651343.

[6] "How Game Studios Can Prioritize Long-Term Player Retention in 2024", [Online] https://solsten.io/blog/player-retention-strategies.

[7] T. Allart, G. Levieux, M. Pierfitte, A. Guilloux and S. Natkin, "Design influence on player retention: A method based on time varying survival analysis," 2016 IEEE Conference on Computational Intelligence and Games (CIG), Santorini, Greece, 2016, pp. 1-8, doi: https://doi.org/10.1109/CIG.2016.7860421.

[8] L. Gu and A. L. Jia, "Player Activity and Popularity in Online Social Games and their Implications for Player Retention," 2018 16th Annual Workshop on Network and Systems Support for Games (NetGames), Amsterdam, Netherlands, pp. 1-6, 2018, https://doi.org/10.1109/NetGames.2018.8463415.

[9] B. Harrison and D. L. Roberts, "Analytics-driven dynamic game adaption for player retention in Scrabble," 2013 IEEE Conference on Computational Inteligence in Games (CIG), Niagara Falls, ON, Canada, 2013, pp. 1-8, https://doi.org/10.1109/CIG.2013.6633632.

[10] L. Pang, Z. Hu and Y. Liu, "How to Retain Players through Dynamic Quality Adjustment in Video Games," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2021, pp. 154-160, https://doi.org/10.1109/IAEAC50856.2021.9390827.

[11] Markus Viljanen, Antti Airola, Tapio Pahikkala, and Jukka Heikkonen, "Modelling user retention in mobile games,". Proc.2016 IEEE Conference on Computational Intelligence and Games (CIG). IEEE Press, 2016, pp. 1–8. https://doi.org/10.1109/CIG.2016.7860393.

[12] S. Demediuk, A. Murrin, D. Bulger, M. Hitchens, A. Drachen, W. L. Raffe, and M. Tamassia. "Player retention in league of legends: a study using survival analysis," Proc. of the Australasian Computer Science Week Multiconference (ACSW '18). Association for Computing Machinery, New York, NY, USA, pp. 1–9. https://doi.org/10.1145/3167918.3167937.

[13] A. Drachen, "Rapid Prediction of Player Retention in Free-to-Play Mobile Games," [Online:] https://arxiv.org/pdf/1607.03202v1.

[14] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, C. Bauckhage, "Predicting Purchase Decisions in Mobile Free-to-Play Games," Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference, Vol. 11, No. 1, 2015, https://doi.org/10.1609/aiide.v11i1.12788.

[15] https://www.bluestacks.com/pl/bluestacks-5.html

[16] A.T. Tekin, and T. Kaya, "Retention Prediction in the Gaming Industry: Fuzzy Machine Learning Approach", Industrial Engineering in the Age of Business Intelligence, 2023, pp. 102-117 https://doi.org/10.1007/978-3-031-08782-0_9.

[17] J.S. Baheri, "Predicting the retention of customers of sport gym using the K-nearest neighbor algorithm," Sport Marketing Studies, 2024, Vol. 5 No. 1, pp. 69-86, https://doi.org/10.22034/SMS.2024.140657.1302

[18] T.O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not," Geoscientific Model Development, 2022, Vol. 15, pp. 5481-5487, https://doi.org/10.5194/gmd-15-5481-2022.