Action Research Literate
Vol. 8, No. 4, April 2024
ISSN: 2808-6988

PLAYER CHURN PREDICTION IN FREE TO PLAY GAME USING ENSEMBLE LEARNING

David[1], Amalia Zahra[2]
Universitas Bina Nusantara, Jakarta, Indonesia
*Email untuk Korespondensi: david004@binus.ac.id[1], amalia.zahra@binus.edu[2]

ABSTRAK

Keywords:

player churn prediction
public dataset
data mining
steam
ensemble learning

Player churn is a prevalent challenge in the gaming industry. Most predictions of player churn utilize private datasets that are not easily accessible to the public. This study aims to investigate the performance of Logistic Regression, Random Forest, Support Vector Machines, and Ensemble Learning models using a dataset from a public API for predicting player churn, in comparison to other studies that typically rely on private game logs. In this research, the dataset consists of 418 unique player IDs, with a churn rate of 15%. After training the models, it was found that Logistic Regression and SVM achieved an accuracy of 95%, Random Forest achieved an accuracy of 96%, and Ensemble Learning, with Neural Network as the meta-learner, achieved an accuracy of 92%. These results underscore the validity of using public API data as an alternative data source for predicting player churn.

INTRODUCTION

Lost Ark, a free-to-play (F2P) Massive Multiplayer Online Role-Playing Game (MMORPG) by Smile Gate RPG, launched in Korea in December 2019, expanded to Japan, and then to Korea, reaching NA/EU on Steam in February 2022. On launch day, Lost Ark soared to second place on Steam with a peak of 1,324,761 concurrent players. (Aggarwal, 2018; Belyadi & Haghighat, 2021)

However, despite its successful launch, Lost Ark faced many challenges and controversies, leading to a notable decline in player numbers within approximately three months. Lost Ark lost about 60% of its players, including both bots and legitimate players. Fig. 1 depicts the player decline (Pedregosa et al., 2011; Villavicencio et al., 2021).

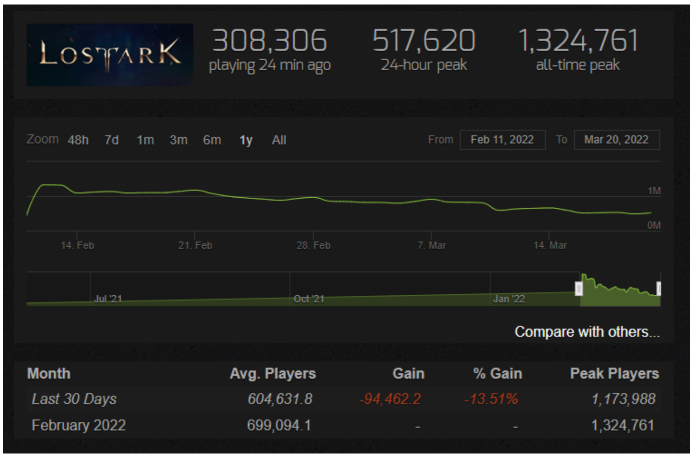

Fig. 1. Steam Chart 20/5/2022

Based on prior research, obtaining public datasets for MMORPGs is challenging due to the private nature of game logs. This study aims to identify factors influencing player churn for predicting churn levels using Data Mining techniques. Such factors can assist developers in enhancing player retention and preventing churn. The research utilizes public datasets, such as public APIs, hoping to achieve results comparable to game logs. If public datasets prove effective, subsequent research could progress more rapidly, benefiting the gaming industry.

The definition of player churn varies; in this context, it is defined as a player ceasing to play for two weeks following their last activity. Logistic Regression, Random Tree Forest, and Support Vector Machine (SVM) are commonly used for predicting player churn, with Random Tree Forest being particularly effective. This study explores a combination of Logistic Regression, Random Tree Forest, SVM, and Ensemble Learning Stacking, where the meta-learner is a Neural Network. Ensemble Learning aims to determine if a combined model can outperform individual models, and Stacking assigns higher weights to better-performing base-learners. The neural network, as a meta-learner, adjusts weights adaptively, giving higher importance to stronger base-learners. Hence, the models used are Logistic Regression, Random Tree Forest, SVM, and Ensemble Learning (Scholkopf & Smola, 2018).

Siqueira, et al. (2017) in their research, "A Data Analysis in World of Warcraft using Game Data Mining," found several relevant features such as Playtime, Playing Density, and Level. The model built was Logistic Regression with 90% accuracy (Siqueira et al., 2017).

Kim, et al. (2017) in their research, "Churn Prediction of Mobile and Online Casual Games using Play Log Data," found that Deep Learning models have similar results just slightly below Traditional Models. They found two relevant features which were active Duration and play Count. The models built were Logistic Regression with 77% accuracy, Gradient Boosting with 78% accuracy, Random Forest with 77% accuracy, CNN with 75% Accuracy, LTSM with 76% accuracy (Kim et al., 2017).

Martins, et al. (2018) in their research, "Applying Commitment to Churn and Remaining Players Lifetime Prediction," found the game model, free-to-play, or subscription-based does not matter. The relevant features found were playtime and playerLevel. The models built were C4.5 with 72% accuracy, RepTREE with 72% accuracy, MLP with 65% accuracy, SVM with 72% accuracy (Kummer et al., 2018).

Rothmeier, et al. (2020) in their research, "Activity Data of a Freemium Online Strategy Game," found that Random Forest worked best in Player Churn prediction. The models built were Random Forest with 97% accuracy, followed by Large Decision Tree, Small Decision Tree, Gradient Boosting Tree, SVM, Logistic Regression, Naïve Bayes, and KNN. From Highest to Lowest (Rothmeier et al., 2020).

Karlberg & Axen (2020) in their research, "Binary classification for Predicting Customer Churn," found the most relevant feature is revenue. The models built were Random Forest with 84% accuracy, Logistic Regression with 79% accuracy, SVM with 77% accuracy (Melis et al., 2022).

Lee, et al. (2019) in their research, "Game Data Mining Competition on Churn Prediction and Survival Analysis Using Commercial Game Log Data," found that loyal users (players that paid frequently) had around 20% less churn accuracy than average users. The models built were Random Forest, Logistic Regression, Extra Gradient Boosting, Generalized Boosting Model, Conditional Inference tree, ranging around 60-72% accuracy (Lee et al., 2018).

Chan, et al. (2016) in their research, "A comparative analysis of SVM and its stacking with other classification algorithms for intrusion detection," found that not every model combined with SVM provides good results; the best result was from SVM with Random Forest. Models built were SVM 91.81%, SVM+BayesNet 92.14%, SVM+AdaBoost 90.53%, SVM + Logistic Regression 92.58%, SVM + IBK 95.79%, SVM +J48 97.12%, SVM + Random Forest 97.5%, SVM+JRip 97.21%, SVM+OneR 91.77%, SVM+Simple Cart 97.49% (Chand et al., 2016).

Mustač, et al. (2022) in their research, "Predicting Player Churn of a free-to-play Mobile Video Game using Supervised Machine Learning," found that most churns happen during game launch and decrease each day. The models built were Random Forest with 94% accuracy and Logistic Regression with 78% accuracy (Mustač et al., 2022).

Based on the results of various studies, two predominant approaches emerge in predicting Player Churn: Traditional (Regression, Classification, Random Forest, K-Means clustering), and Deep Learning (CNN, LTSM). In terms of accuracy, Traditional methods generally have slightly better accuracy.

While both approaches exhibit similar accuracy, Random Forest stands out as the most frequently used and highest-performing machine learning model. The game model, distinguishing between Free-to-play and subscription, has limited impact on Churn Level. Influential metrics such as playtime, player level, and playing density reflect a player's commitment to the game, but not every metric is significant; thus, feature selection is essential.

The primary challenge lies in acquiring sufficient and public datasets. Future research could explore datasets more readily available, such as public data, to validate accuracy and achieve similar outcomes.

For the proposed model, Logistic Regression, Random Tree Forest, SVM, and Ensemble Learning are selected. Logistic Regression, widely used in prior research, averages an accuracy of 81%. Random Forest, identified as the best approach with 97% accuracy in a specific study, holds an average accuracy of 88%. SVM, with an average accuracy of 74.5%, is also commonly employed. Ensemble Learning, especially stacking, combines these models with a Neural Network meta-learner. The Neural network's adaptability in assigning weights ensures that stronger base-learners receive greater emphasis, while weaker ones receive less. LTSM and CNN are omitted as Deep Learning, per literature review, is more suitable for complex models like Image Classification and Computer Vision.

METHOD

The entire experiment was conducted using Python on Google Colab. The Python version used was Python 3.9.16, with the GNU Compiler Collection (GCC) 9.4.0. The experiment ran on an Intel® Xeon® CPU @ 2.20 GHz.

The proposed method involved utilizing the Steam public API to compile a public dataset. This was achieved by accessing store.steampowered.com/appreviews/1599340?json=1&filter=recent&num_per_page=100 to retrieve a JSON response with the 100 latest reviews. Fig 2 illustrates an example of the response.


```
{
    success: 1,
  - query_summary: {
        num_reviews: 100
    },
  - reviews: [
      - {
            recommendationid: "113041632",
          - author: {
                steamid: "76561198171069679",
                num_games_owned: 0,
                num_reviews: 1,
                playtime_forever: 69164,
                playtime_last_two_weeks: 0,
                playtime_at_review: 23131,
                last_played: 1692058040
            },
            language: "english",
            review: "If you come to this game for pvp turn 180 degree
            timestamp_created: 1648593299,
            timestamp_updated: 1648593299,
            voted_up: false,
            votes_up: 0,
            votes_funny: 0,
            weighted_vote_score: "0.47486037015914917",
            comment_count: 0,
            steam_purchase: true,
            received_for_free: false,
            written_during_early_access: false,
            hidden_in_steam_china: false,
            steam_china_location: ""
        },
```


Fig 2. Review List JSON Response

To filter public profiles, we just needed to use api.steampowered.com/IPlayerService/GetRecentlyPlayedGames/v0001/?key={{developer_key}}&steamid={{id}}&format=json to loop through compiled Steam IDs and check if it returns anything. If it's empty, that means it's a private profile. Around 500 people are expected to be compiled and assumed to be random.

After the public profiles were compiled, we used api.steampowered.com/IPlayerService/GetRecentlyPlayedGames/v0001/?key={{developer_key}}&steamid={{id}}&format=json to get the playtime_2weeks and playtime_forever. Fig 3 shows the JSON response for recently played games.

```
response: {
    total_count: 4,
  - games: [
      - {
            appid: 1135230,
            name: "Ember Knights",
            playtime_2weeks: 1788,
            playtime_forever: 1760,
            img_icon_url: "6309ed12efcbdc62ef1d1c00175d2a0398d4eb39"
        },
      - {
            appid: 632360,
            name: "Risk of Rain 2",
            playtime_2weeks: 146,
            playtime_forever: 7877,
            img_icon_url: "0b809ac6f25e6570fecae5fc47bca0139a7bf70c"
        },
      - {
            appid: 252950,
            name: "Rocket League",
            playtime_2weeks: 105,
            playtime_forever: 219364,
            img_icon_url: "9ad6dd3d173523354385955b5fb2af87639c4163"
        },
      - {
            appid: 289070,
            name: "Sid Meier's Civilization VI",
            playtime_2weeks: 98,
            playtime_forever: 16551,
            img_icon_url: "9dc914132fec244adcede62fb8e7524a72a7398c"
        }
    ]
```


Fig 3. Recently Played JSON Response

We hit the API for one month daily to track changes in playtime_forever, determining daily play hours. Additionally, we used api.steampowered.com/ISteamUserStats/GetUserStatsForGame/v0002/?appid=1599340&key={{developer_key}}&steamid={{id}}&format=json to retrieve the achievement count, providing insights into the players' progress within the game. Fig 4 shows the response containing a list of achievements in the game.


```
- playerstats: {
      steamID: "76561198058459239",
      gameName: "Lost Ark Test 1",
    - achievements: [
        - {
              name: "2010007",
              achieved: 1
          },
        - {
              name: "1050011",
              achieved: 1
          },
        - {
              name: "1050012",
              achieved: 1
          },
        - {
              name: "6080002",
              achieved: 1
          },
        - {
              name: "1030001",
              achieved: 1
          },
        - {
              name: "1030002",
              achieved: 1
          },
```


Fig 4. Achievement List JSON Response

The data is then pre-processed into a suitable dataset with features such as id, review, playtime_forever, playtime_2weeks, average_playtime, achievement, and churn. Here, churn is defined as not logging into the game for two weeks. The pre-processing includes some rules, such as:

1. If an account is banned or made private abruptly without a two-week churn, the outlier is deleted.
2. If there's churn and then a return to play for a short duration (1-2 days only), the time outlier is deleted, and it's considered as churn.
3. If there's churn and then a return but with a long and consistent playtime, it's considered not churn.

For Logistic Regression, Random Forest, and SVM, we employed a 60% training, 20% validation, and 20% testing split. In contrast, Ensemble Learning combined training and validation into an 80% training set, reserving 20% for testing.

RESULT AND DISCUSSION

In Ensemble Learning, each base-learner (Logistic Regression, Random Forest, SVM) undergoes individual training with 80% of the data, subjected to 10-fold cross-validation. The outputs from these models serve as input to construct the meta-learner (Neural Network). Subsequently, the meta-learner makes predictions using the 20% test data.

The model will be evaluated using a Confusion Matrix to obtain comprehensive performance metrics. The performance results will be compared to the previous study Karlberg & Axen, to assess accuracy. This analysis aims to determine if the public dataset can achieve a sufficiently high accuracy level (Axén & Karlberg, 2020).

As depicted in Table 1, the dataset comprises 417 observed IDs, and the corresponding ratios for reviews and churn are outlined below.

Table 1. Dataset Exploration

Review	Churn	Number
Bad	No	79
	Yes	17
Good	No	276
	Yes	45

From the data presented, it is evident that 14% of good reviews resulted in churn, 17.7% of bad reviews led to churn, and the overall churn rate is 15%. The scatterplot in Fig. 4.1 illustrates the distribution of average play hours and churn. Remarkably, individuals who churn rarely exceed 150 minutes of average playtime over the course of a month.

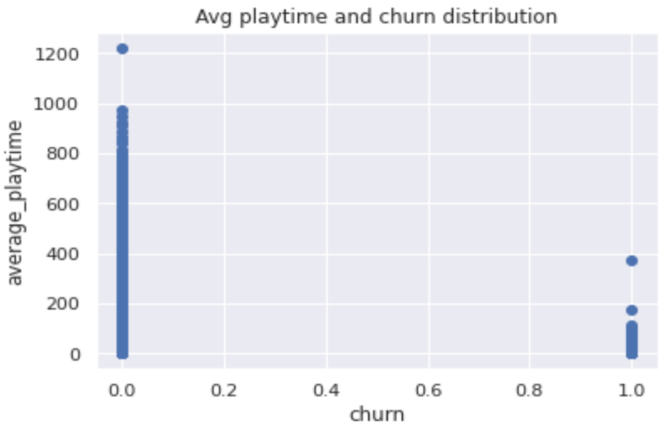

Fig 5. Distribution of playtime and churn

Table 2. Models performance

Model	*F1*	*Accuracy*	*Precision*	*Recall*
Logistic Regression	88%	95%	80%	100%
Random Forest	90%	96%	93%	87.5%
SVM	88%	95%	80%	100%
Ensemble (Neural network)	84%	92%	72%	100%

As evident from Table 2, the best-performing models, following Random Forest, are Logistic Regression and SVM, both achieving identical results: an F1 score of 88% and an accuracy of 95%. The ensemble stacking model, utilizing Logistic Regression, Random Forest, and SVM as base models with Neural Network as the meta-learner, ranks slightly lower with an F1 Score of 84% and an accuracy of 92%. The lower accuracy of the Ensemble Learning model compared to base models aligns with findings in other research, such as Matloob, where EnsembleSVM was 3% lower than LIBSVM for the covtype dataset (Matloob et al., 2021).

This experiment has shown that Random Forest was the best model with an F1 Score of 90%, Accuracy of 96%, Precision of 93%, and Recall of 87.5%. Fig. 6.2 shows the confusion matrix of Random Forest, indicating that out of 84 samples, 67 were True Positive, 14 were True Negative, 1 was False Positive, and 2 were False Negative.

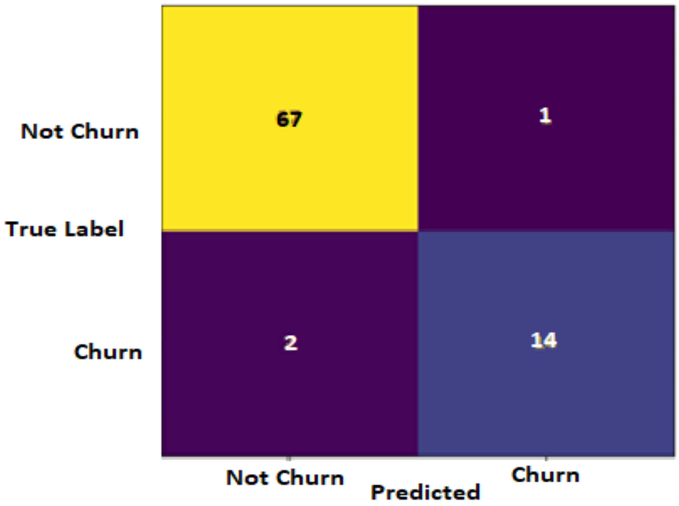

Fig 6. Random Forest Confusion Matrix

After completing the data exploration of the specific dataset, it was found that the three incorrect predictions were caused by those particular samples being outliers. The experiment results demonstrate that the dataset from a public API is effective for modeling player churn predictions. In contrast to studies like Karlberg & Axen, 2020, which used private datasets and achieved a Random Forest accuracy of 84%, Logistic Regression accuracy of 79%, and SVM accuracy of 77%, our publicly sourced dataset shows comparable accuracy levels (Axén & Karlberg, 2020).

The best-performing model in this study, utilizing a public dataset, achieves high performance comparable to other research, such as Rothmeier, which used private game logs from BLUE BYTE and achieved a Random Forest accuracy of 97%. Despite potential differences between public and private datasets, this study affirms that lower-quality public datasets can still yield commendable performance.

The results of this experiment also compared with other studies using private datasets, showing that the best model in this study, which used public datasets, was able to achieve high performance comparable to other studies using private datasets.

CONCLUSION

This experiment was conducted to assess the performance of Logistic Regression, SVM, Random Forest, and Ensemble Learning using a public dataset to predict player churn, comparing it with other related works where private datasets were commonly used.

The proposed models have demonstrated satisfactory performance, competing favorably with models from other studies. Similar to other research, Random Forest emerges as the dominant model with an F1 score of 90% and an accuracy of 96%. Logistic Regression and SVM follow closely, both achieving identical results: an F1 score of 88% and an accuracy of 95%. The Ensemble Stacking model, with Neural Network as the meta-learner, ranks slightly lower with an F1 score of 84% and an accuracy of 92%.

While the Ensemble model did not outperform individual models, its performance is deemed satisfactory. This suggests that the dataset from a public API is a valid approach for predicting player churn and can compete effectively with private game logs.

The current research's features are limited due to the selected platform and API. For future research, individuals can explore other platforms with a broader range of features. It’s also recommended to try other types of Ensemble learning besides Stacking, or using other meta-learners. Churn criteria can also be modified according to the type of game

REFERENCES

Aggarwal, C. C. (2018). *Neural networks and deep learning* (Vol. 10, Issue 978). Springer.

Axén, M., & Karlberg, J. (2020). *Binary Classification for Predicting Customer Churn*.

Belyadi, H., & Haghighat, A. (2021). *Machine learning guide for oil and gas using Python: A step-by-step breakdown with data, algorithms, codes, and applications*. Gulf Professional Publishing.

Chand, N., Mishra, P., Krishna, C. R., Pilli, E. S., & Govil, M. C. (2016). A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring)*, 1–6.

Kim, S., Choi, D., Lee, E., & Rhee, W. (2017). Churn prediction of mobile and online casual games using play log data. *PloS One*, *12*(7), e0180735.

Kummer, L. B. M., Nievola, J. C., & Paraiso, E. C. (2018). Applying commitment to churn and remaining players lifetime prediction. *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8.

Lee, E., Jang, Y., Yoon, D.-M., Jeon, J., Yang, S., Lee, S.-K., Kim, D.-W., Chen, P. P., Guitart, A., & Bertens, P. (2018). Game data mining competition on churn prediction and survival analysis using commercial game log data. *IEEE Transactions on Games*, *11*(3), 215–226.

Matloob, F., Ghazal, T. M., Taleb, N., Aftab, S., Ahmad, M., Khan, M. A., Abbas, S., & Soomro, T. R. (2021). Software defect prediction using ensemble learning: A systematic literature review. *IEEE Access*, *9*, 98754–98771.

Melis, A., Brown, J., & Coulter, C. (2022). *Designing sustainable and resilient cities: small interventions for stronger urban food-water-energy management*. Taylor & Francis.

Mustač, K., Bačić, K., Skorin-Kapov, L., & Sužnjević, M. (2022). Predicting player churn of a Free-to-Play mobile video game using supervised machine learning. *Applied Sciences*, *12*(6), 2795.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Rothmeier, K., Pflanzl, N., Hüllmann, J. A., & Preuss, M. (2020). Prediction of player churn and disengagement based on user activity data of a freemium online strategy game. *IEEE Transactions on Games*, *13*(1), 78–88.

Scholkopf, B., & Smola, A. J. (2018). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Siqueira, E. S., Castanho, C. D., Rodrigues, G. N., & Jacobi, R. P. (2017). A data analysis of player in world of warcraft using game data mining. *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 1–9.

Villavicencio, C. N., Macrohon, J. J. E., Inbaraj, X. A., Jeng, J.-H., & Hsieh, J.-G. (2021). Covid-19 prediction applying supervised machine learning algorithms with comparative analysis using weka. *Algorithms*, *14*(7), 201.