A Review of the Leading Solutions for Predicting Victories in Video Games

Ali Mohammadi Esfahani – 101245278

Azhar Syed – 101150942

Françoise Blanc – 101209287

Sandeep Reddy Venna – 101214885

Introduction

The "IEEE BigData 2021 Cup - Predicting Victories in Video Games" is a data mining competition in which participants submitted solutions to predict victories in the turn-based game *Tactical Troops: Anthracite Shift*. The predictions were made at a given time in a game using logs from historical games. This paper offers an overview of the supervised machine learning solutions implemented by the four most successful teams. First, each paper from the top teams is summarized by providing the main key points and contributions from the team's work. Then, we conclude this review with a brief comparison between these solutions and a summary of the main gaps and research opportunities.

Related Work

1st Place Solution

The best solution proposed by Xiao et al. [1], which acquired the highest score in the competition, used a gradient boosting decision tree (GBDT), named WP-GBDT, to extract an optimal subset of the most important features of gameplay logs and predict the winner of the games. The authors relied on an exhaustive feature engineering process to generate effective features from tabular format logs, metadata, flattened format logs, and truncated format logs. One of the major contributors to Xiao et al. winning the competition includes the design of a group-based recursive feature elimination method combined with the LightGBM algorithm to obtain an optimal subset of features. Their feature selection process eliminates features based on the ranking feature importance. Using this method, the authors first divided the features into six groups. Then, they joined the basic features within those six groups and used the LightGBM algorithm for training. They used the recursive method to obtain the most important features of each group. Finally, the authors used the LightGBM algorithm as the base model and proposed an ensemble approach to improve the generalization performance of the models. For boosting method, the authors used "dropouts meet multiple additive regression trees" (DART) to improve the model prediction performance. For evaluating the results, Xiao et al. adopted stratified k-fold cross-validation as their local validation strategy to assess the generated features and trained models. The evaluation results showed that the gap between local and public validation scores was less than 0.009. The final results show that the proposed WP-GBDT approach obtained an Area Under the ROC Curve (AUC) score of 0.8997, which is the highest score in the challenge. Although the authors have used a wide range of gameplay logs, the use of screenshots format data could have perhaps increased the accuracy of their solution. In addition, the authors could have included in their work the results of other machine learning algorithms to compare accuracy.

2nd Place Solution

Feature engineering, along with a strong validation strategy, is the major contributor for Vu et al. [2] succeeding in taking the second position of the competition with an AUC score of 0.8980. The basic set of features was engineered from the tabular dataset and metadata of game objects. Distance between two units within a team and between teams and other distance-based features were extracted from snapshots. Features related to weapons and gadgets used during every game step were extracted from truncated log files. Vu et al. also considered interaction features and mean-encoding features. The interaction features encode the relationship between players and a specific unit type or weapon, while mean-encoding features relate to the probability of winning a game based on certain combinations of weapons, gadgets, and units selected by the player. The best features were selected from more than 2000 available features by incorporating the greedy forward search and greedy backward search. SHApley Additive exPlanations (SHAP) analysis is further utilized to attain confidence on a subset of features. Vu et al. is the only leading team that has considered a hybrid model using a general model and two specific models for each game mode: devastator and domination. The general model utilizes all the finalized best features, and the two other models use features specific to each game mode. After experimenting with logistic regression, XGBoost, LightGBM, Catboost and neural network, LightGBM is selected for its better performance and low training time. Stratified KFold model validation technique is used, and the model is validated in terms of both five-folds and 19-folds to avoid overfitting issues. The authors have been particularly thorough with this step as they lost two of the earlier challenges because of overfitting issues. GridSearchCV is employed on top of the Bayesian optimized parametric subspace for hyper-parameter tuning.

3rd Place Solution

The third best solution achieving an AUC score of 0.8978 was submitted by Tseng [3]. His work's success is partly due to the use of an XGBoost model with extensive feature engineering of game logs and metadata. The solution uses tabular, flattened, truncated log files and the metadata of the objects for training and making predictions. As part of his solution's feature engineering process, he applied basic transformations like one-hot encoding of categorical data and replacing missing values with an extreme value. This followed by using subtraction and percentage technique over aggregated features in the current game as well as a team's past games to introduce more informative features for target prediction. Finally, he applied target encoding on categorical variables as part of feature engineering. Due to the convenience in extracting feature importance from a tree-based model and high performance, Tseng used XGBoost implemented in Python as a classifier for his solution. He trained the model using Bayesian Hyperparameter Optimization with cross-validation technique to find the best hyperparameters for the model. During the training, he chose the number of folds in cross-validation as three because the training data size is double that of testing. After experimenting with various values of the number of rounds for early stopping and the learning rate, the values 200 and 0.2 respectively were concluded as this setup produced the best performance. Tseng's work suggests that features related to health points and the number of dead units have high information gain and are primary features in predicting the winner. The introduction of aggregated features calculated from truncated log files improved the performance significantly. It suggests scope for further improvements in performance by utilizing the truncated data more efficiently.

4th Place Solution

One of the solutions implemented by Ruta et al. [4] obtained an AUC score of 0.8963, placing them in fourth place in the challenge. To stand out among the leading teams, which also used GBDT, Ruta et al. focused their efforts on using a semi-automated feature engineering process and ensemble learning. Their feature engineering process involved a set of general guidelines mainly concerning the definition of categorical and numerical variables and data aggregation techniques. Their three main solutions used different ensemble averages of boosting models (i.e., XGBoost, LightGBM, and CatBoost) and a mix of levels of data representation. The authors could have explained why they have not considered other types of algorithms other than GBDT. It is assumed that this choice was made based on their experience, as GBDTs have been used in leading solutions of previous machine learning competitions. Ruta et al. implemented an effective hyperparameter optimization technique using hierarchical, greedy, rotational grid search. One interesting aspect of the work of Ruta et al. is that they investigated the use of a convolutional neural network (CNN) to predict the winner only using the game screenshots. Even though the resulting AUC of 0.67 for the CNN is significantly lower than any predictions made with data from the game logs, further investigation could be made to improve image-based predictors for video games. A key finding from Ruta et al. is that most of the predictability comes from the tabular format and features related to health points from the player's units. Adding features from the truncated and flattened logs format only contributed to a small percentage of the predictability. However, Ruta et al. observed that some individual features generated only using the truncated data reached greater AUC scores than individual features from the other data formats.

Conclusion

The final AUC scores of the four most successful teams in predicting victories in the game *Tactical Troops: Anthracite Shift* are respectively 0.8997, 0.8980, 0.8978, and 0.8963. These results are better by over 1% than the baseline model created by the challenge's organizers [5]. Since the top solutions and the baseline solution all used GBDT, the proposed solutions mainly differentiate from one another through their implemented feature engineering and feature selection method. Only one of the leading teams, Vu et al. [2], provided insights about other tested algorithms, including logistic regression, variants of GBDT, and neural networks. Future work could explore other commonly known algorithms in winning prediction, such as random forest and deep neural networks. Additionally, most of the teams made limited use of the game screenshots. Better feature extraction from the screenshots and binary layer masks could be considered. Lastly, most teams used the truncated data by aggregating the features present in them over time to develop new aggregated features that were later used in prediction. Instead of using aggregated features, the contestants could have used the features from the truncated logs as multivariate sequential input data in predicting the winner. Such sequential data can be processed with deep neural networks like recurrent neural networks (RNN). RNNs are suitable models for classifying sequential data as they are designed to capture dependencies between time steps.

References

[1] H. Xiao, Y. Liu, D. Du and Z. Lu, "WP-GBDT: an approach for winner prediction using gradient boosting decision tree," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 5691-5698, doi: 10.1109/BigData52589.2021.9671688.

[2] Q. H. Vu, D. Ruta, L. Cen and M. Liu, "A combination of general and specific models to predict victories in video games," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 5683-5690, doi: 10.1109/BigData52589.2021.9671285.

[3] K. Tseng, "Predicting victories in video games: using single XGBoost with feature engineering: IEEE BigData 2021 Cup - Predicting Victories in Video Games," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 5679-5682, doi: 10.1109/BigData52589.2021.9671454.

[4] D. Ruta, L. Cen, M. Liu and Q. H. Vu, "Automated feature engineering for prediction of victories in online computer games," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 5672-5678, doi: 10.1109/BigData52589.2021.9671345.

[5] M. Matraszek, A. Janusz, M. Świechowski and D. Ślęzak, "Predicting victories in video games - IEEE BigData 2021 Cup report," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 5664-5671, doi: 10.1109/BigData52589.2021.9671650.