League of Legends: EZWIN

Alec Louie A. Marohom

School of Electrical, Electronics, and Computer Engineering, Mapúa University  
658 Muralla Street, Intramuros, Manila 1002, Philippines

alecmarohom15.p@gmail.com

Abstract— League of legends is one of the leading MOBA-genre games that is played across the globe. Along with it is an uprising form of sport competition called Electronic Sports (eSports) which garnered an interest in application of data science called Game Analytics in which multiple variables are considered to optimize and determine the odds of a team winning the game. The game uses a competitive index to measure the skills of a player in which multiple players are grouped to establish a balanced game instance. Recent studies are in demand of exploration and experimentation in the emerging field. Hence the study is an exploratory attempt to advance the field of research. The study utilizes a dataset collected from the game and neural networks architecture utilizing dense layers with adam optimizer and relu activation function. Using the data of the most skilful players, and different experimental iterations of dense network architecture performed by the study, the model predicts the outcome of a game with 97.62% accuracy.

Keywords— eSports, Neural Networks, MOBA, Data Science, Game Analytics, League of Legends

1. Background

The competitive scene of computer games (eSports) and their fans collectively brought an unsaturated and relatively new field of research and study called Game Analytics. The sport itself is driven by the advancement of technology hence its tremendous digital footprints in its wake. It can be segregated into three parts: collection and transformation, visualization, and, analyzation. Ani R., et al [1] experimented with Random Forest, Adaboost, Gradient Boosting, and Extreme Gradient Boosting algorithms which resulted to 97.01% to 99.75% accuracy. These algorithms were evaluated before and within the match of a game and are combined. Other related studies that have been made were focused on correlation of human psychology and players’ skill in a specific game to forecast the outcome of the game [2]. However, the study is based on quantifiable metrics within the game architecture only. According to Krotin, A., et al. (2019) one of three relevant research areas in electronic sports is prediction related. With the field in its early stages, current studies lack in experimental works in which this study aims to suffice.

The main objective of the study is to experiment classification algorithms to predict the outcome of a game specifically: League of Legends. The specific objectives of the study are the following:

1. Experiment different dense network architectures
2. Evaluate experimental dense network architectures to determine its accuracy
3. Compare the evaluated models from the experiment to determine the optimal setting for the dataset

One of many interesting topics in games research is forecasting players’ performance [2]. Players’ performance also translates well to the outcome of the game which it plays a big factor. The significance of this study is to progress the application and explore the possibilities of neural networks in game analytics. More specifically, the study will be able:

1. to lessen the demand of exploratory work and research in Game Analytics
2. to apply game-centric data points as factor of predictive analysis as compare to other studies which mainly includes player-centric data points
3. Methodology
4. Dataset

The dataset used in the study is a combination of quantifiable objectives in the game of League of Legends. The game has its own competitive index which players are binned into their respective skill group in which the dataset is based on the three most skillful bins. It is outsourced from Kaggle and prepared by Shin, M. (2020). It contains 190,000 concatenated rows, 49 quantitative columns, and 1 unique ID column of data which a single row affects the outcome of a game [4]. It is under the licensing CC0 1.0 which enables anyone from the public to work with it [5]. The dataset contains values from both team which is denoted as blue and red. The following is a summary of the dataset regardless of the team it belongs to:

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| gameId | Unique Riot game ID. Can be used with the Riot Games API. |
| gameDuraton | Game Duration(seconds) |
| Wins | target variance. 1 if the team win, 0 lose |
| FirstBlood | team whether first blood |
| FirstTower | team whether first tower |
| FirstBaron | team whether first baron |
| FirstDragon | team whether first dragon |
| FirstInhibitor | team whether first inhibitor |
| DragonKills | team dragon kill counts |
| BaronKills | team baron kill counts |
| TowerKills | team tower kill counts |
| InhibitorKills | team inhibitors kill counts |
| Ward Placed | team ward placed counts (Number of warding totems) |
| Wardkills | team ward killed counts (Number of warding killed) |
| Kills | team enemy champion kill counts |
| Death | team death counts |
| Assist | team enemy kill assist counts |
| ChampionDamageDealt | team damage dealt to enemy champion |
| TotalGold | team earned total gold |
| TotalMinionKills | team kill minion counts |
| TotalLevel | team total champion levels |
| AvgLevel | team average champion levels |
| JungleMinionKills | team kill jungle minion counts |
| KillingSpree | team squence kill counts |
| TotalHeal | team heal amounts |
| ObjectDamageDealt | team damage dealt to objects |
| rank | skill bin of the game |

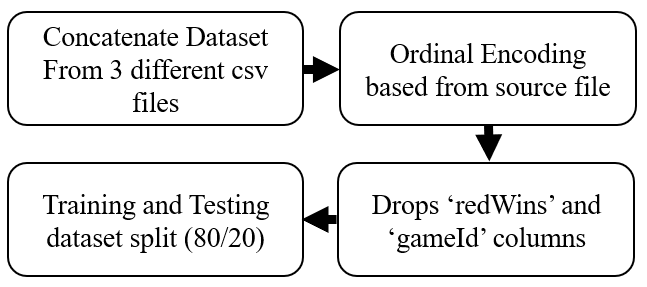
1. Pre-processing

Table 2.1 Summary of Dataset

Since the dataset comes in three different bins, the researcher manually labeled its source in an ordinal encoding fashion. A column named ‘redWins’ and ‘gameId’ was also removed since the former mirrors the column ‘blueWins’ and the latter is a unique ID for a game instance which will not be helpful for predictive algorithms. All columns are set to ‘int64’ datatype since all the data does not contain floats. Testing dataset is the 20% of the dataset while the other 80% served as the training dataset.

Table 2.3 Dense Network Architecture Iterations Results

Figure 2.1 Data Pre-processing Dataflow



1. Neural Network Architecture and Rationale

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Hidden Layer | | |  |
| **Iteration No.** | **Input** | **Dense 1** | **Dense 2** | **Dense 3** | **Output** |
| 1 | 48 | 50 | x | x | 2 |
| 2 | 48 | 100 | x | x | 2 |
| 3 | 48 | 50 | 80 | x | 2 |
| 4 | 48 | 50 | 100 | x | 2 |
| 5 | 48 | 100 | 80 | x | 2 |
| 6 | 48 | 100 | 100 | x | 2 |
| 7 | 48 | 100 | 80 | 50 | 2 |
| 8 | 48 | 100 | 100 | 100 | 2 |

The predictive model implemented utilizes dense network architecture. The first iteration of the model utilized Support Vector Machine with Radial Basis Function kernel which resulted to 71.14% accuracy. However, neural networks work better than support vector machine for identifying patterns [6]. Hence, the researcher proceeded with dense neural network as indicated in figure 2.1 using the optimal architecture based from several iterations of the model indicated in table 2.2 and 2.3. The architecture utilizes hidden layers with ‘*relu’* activation function which is more efficient than other functions while the output utilizes ‘*softmax’* activation which provided a probability distribution of outcomes [7].

Table 2.2 Dense Network Architecture Iterations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration No.** | **Total Parameters** | **Accuracy** | **Precision** | **Recall** |
| 1 | 2,552 | 95.81% | 92.70% | 99.36% |
| 2 | 5,102 | 96.55% | 99.22% | 93.76% |
| 3 | 6,692 | 97.47% | 96.92% | 98.03% |
| 4 | 7,752 | 96.36% | 95.79% | 96.92% |
| 5 | 13,142 | 97.62% | 98.38% | 96.40% |
| 6 | 15,202 | 97.46% | 96.45% | 98.50% |
| 7 | 17,132 | 96.41% | 95.83% | 96.97% |
| 8 | 25,302 | 94.39% | 99.33% | 89.28% |

1. Training Methods

It is compiled with ‘*adam’* optimizer and ‘*sparse categorical cross entropy*’ loss function. The optimizer is a widely used optimization algorithm that offers efficiency and is suited for large datasets [8]. On the other hand, the loss function is an efficient algorithm for sparse labels. The model is trained over 10 epochs in batches of 32. It is also equipped with model checkpoints that saves the best model based on the loss from the testing dataset which avoids overfitting.

1. Postprocessing

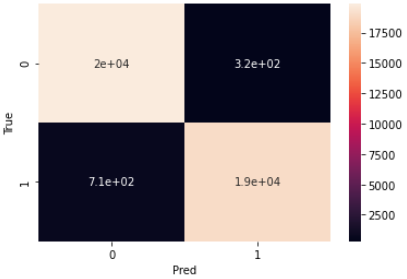
The output of the model, which used ‘*softmax’* activation function, is an array with two elements that corresponds to 0 and 1. The value of each elements denotes the probability of 0 (lose) or 1 (win) respectively. This array is passed through a function called ‘argmax’ which returns the argument with a larger value. It is most used in machine learning which is under the NumPy library.

1. Results and Discussion

The predictive model resulted to 97.43% accurate with 98.38% precision and 96.40% recall. Accuracy is the overall ratio of correctly predicted to the total observations. However, most of the time, data is not symmetric which demands for precision and recall. Precision is the ratio of correctly predicted positive observations over the total observations while recall or sensitivity is the ratio of correctly predicted positive observations over total actual class [9]. With high precision and recall, it can be deduced that the model architecture is stable and consistent. These metrics are important for the model as it supports and justifies the quality of the model especially in predictive/classification models.

1. Conclusion

Figure 3.1 Confusion Matrix



While most of the related researches in the field focuses on the players’ psychology and skill to predict the outcome of the game, the researcher successfully explored other parameters that are included within the game. The researcher successfully explored different dense network architectures and evaluated its accuracy. Based from the 8 iterations of the model architecture, the fifth iteration produced the most promising results. It can also be observed that an overfitting occurs as the number of parameters increases. The study has obtained 97.62% accuracy in predicting the outcome of a game.

1. References
2. A. R, V. Harikumar, A. K. Devan and O. S. Deepa, "Victory prediction in League of Legends using Feature Selection and Ensemble methods," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 74-77, doi: 10.1109/ICCS45141.2019.9065758. M. Aung, et al., Predicting Skill Learning in a Large, Longitudinal MOBA Dataset, 2018
3. Krotin, A., et al., Towards Understanding of eSports Athletes' Potentialities: The Sensing System for Data Collection and Analysis, 2019
4. M. Shin, League Of Legends High elo Ranked Games (2020), 2020. [Online]. Available: https://www.kaggle.com/gyejr95/league-of-legends-challenger-ranked-games2020. [Accessed: 28-JUL-2020].
5. [Online]. Available: https://creativecommons.org/publicdomain/zero/1.0/legalcode. [Accessed: 28-JUL-2020].
6. M. Sheeva, The Scuffle Between Two Algorithms -Neural Network vs. Support Vector Machine, 2018
7. S. Sharma, Activation Functions in Neural Networks, 2017
8. J. Brownlee, Gentle Introduction to the Adam Optimization Algorithm for Deep Learning, 2017
9. R. Joshi, Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures, 2016