# DotA 2 Game Prediction

**Yu-Chun Chen** yuc330@ucsd.edu
**Yanyu Tao** yat017@ucsd.edu
**Shuibenyang Yuan** shy166@ucsd.edu
**Yue Xiao** yux164@ucsd.edu
Halıcıoğlu Data Science Institute, University of California San Diego

## Abstract

As of now, eSport has emerged as a new kind of competitive sport. Its popularity has inspired attempts to predict game results by using machine learning techniques. Here, an evaluation of models and feature representations are presented to predict the results of DotA 2, a well-known game in eSport currently. Models used include Logistic Regression, Decision Tree, and Deep Learning. The main goal is to reach an accurate prediction of DotA 2 games with data we have.

## 1. Introduction

Defense of the Ancients 2, also known as DotA 2, is a multiplayer, 5 vs. 5, online battle arena (MOBA) video game that developed and published by Valve. It is played in matches between two teams who compete to destroy the 'Ancient,' a large structure in the opposing base, while defending their own collectively on the selected maps (McDonald). Players can choose their 'heroes', or game characters with unique skills and styles of play, from a pool of 119. According to their skills, 'heroes' can be generally divided into four categories – carry, support, initiator, and tanker. Among which, carries and supports are the most essential roles to each team.

While 'hero' selection is one of the cores to victories of matches, many other properties, such as team cooperation, game mode, and etc., will also affect the quality and result of game. Thus, in this project, we are going to build machine learning models of winning prediction to predict the binary results of DotA 2 game matches given some property data.

## 2. Literature Review

Dota2 Games Results Data Set is the dataset collected by UCI Machine Learning Repository with statistics of games played in a space of 2 hours on the 13th of August, 2016. It consists the information of win or lose, location, game mode, game type and hero selection based on every match. This dataset can be preprocessed with one hot encoder and is useful for our explorative analysis and model training. To study this type of data, logistic regression is one of the popular methods currently found in many related papers.

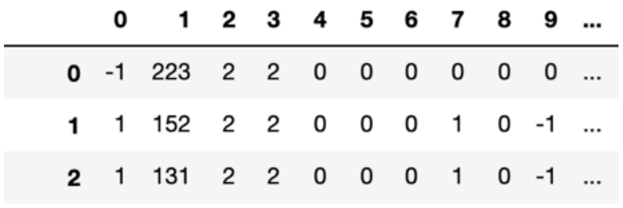
The report *To Win or Not to Win* has studied a similar problem and built two prediction models to determine the outcome of a DotA 2 match. The dataset that has been studied in this report came from Valve's Steam API, documenting information of 30,426 DotA 2 games from 1/23/2015 to 02/20/2015. It consists similar data as the above dataset but based on every player. The first prediction model used simple logistic regression without any regularization on a binary feature vector of heroes with a score of 69.42%. The second model used an augmented logistic regression combining the first model with a genetic fitness metric, which performed a score of 74.1% (Kalyanaraman).

The report *Learning DotA 2 Team Compositions* has also built a logistic regression model to predict the result of matches but with PCA dimensionality reduced DotA 2 match data. The model predicted at 57% for 7 PCA dimensions, compared to 62% for a full dataset (Agarwala).

Our finding that logistic regression outperformed decision tree on this type of data corresponds to the conclusions of the above works. However, the overall performance of our models is not as high as the scores from the existing works due to the difference of datasets. The dataset we used lacks player information while that used in *To Win or Not to Win* had full information about players and their heroes.

## 3. Dataset

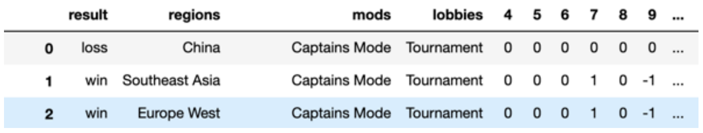
Our dataset without null values, credited to UCI Machine Learning Repository, contains 117 categorical value columns of 92,650 DotA 2 games, in which one column represents the result of the game, either win or lose, while other 116 columns describe the properties of the game. Following is a snap of the dataset.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | -1 | 223 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| **1** | 1 | 152 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | -1 | ... |
| **2** | 1 | 131 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | -1 | ... |

*Figure 1: Dataset*

As shown above, first four columns of the dataset is given in integer codes that represent different column values, and we are given a couple json files to decode them. Specifically, the first column represent the result. The rest 113 columns are the one-hotted result of the heroes chosen for a game: 1 indicates the hero chosen for a team, while -1 indicates the hero chosen for the opposing team. After using the json files, the first four columns of the dataset can be decoded to the following Figure 2.



| | result | regions | mods | lobbies | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | loss | China | Captains Mode | Tournament | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| **1** | win | Southeast Asia | Captains Mode | Tournament | 0 | 0 | 0 | 1 | 0 | -1 | ... |
| **2** | win | Europe West | Captains Mode | Tournament | 0 | 0 | 0 | 1 | 0 | -1 | ... |

*Figure 2: Decoded Dataset*

## 4. Exploratory Analysis

Then, an exploratory analysis is performed on this dataset. The dataset contains 48,782 wins and 43,868 losses. Although slightly different, the dataset is generally balanced with respect to results. In the following Figure 3, we would like to compare the number of games in each region, game mode, and game lobby in the left column, along with the number of games in the same categories while separated according to game results in the right column.

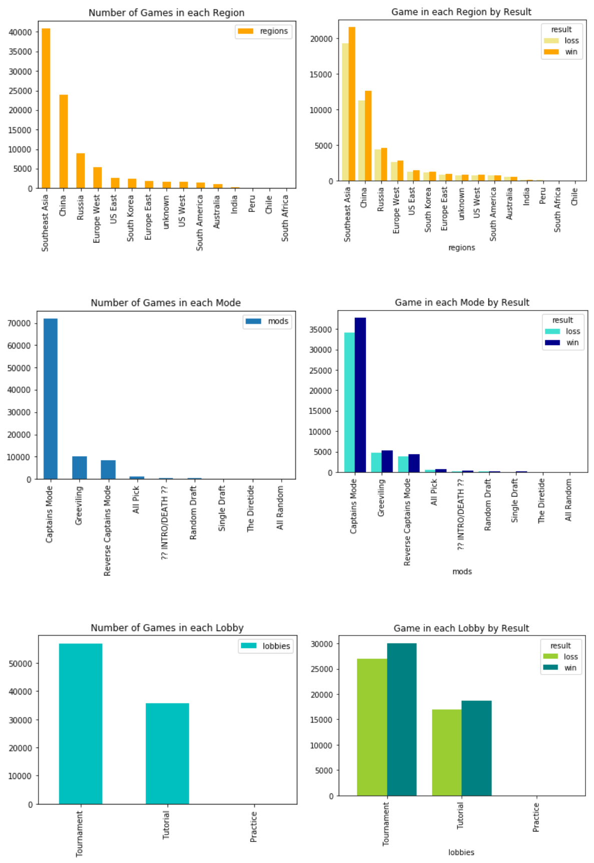


*Figure 3. Number of Games in each Region/Mode/Lobby and by Result*

From left column, we see that South East Asia hosts the most games, nearly half of the games, followed by China who hosts more than 1/4 of the games. Then, around 15,000 of the games are hosted in Russia and Western Europe. We can also observe that majority of the games in the dataset are played in Captain Mode, while around 10,000 are played in each of Greeviling and Reverse Captain Mode. Then in the third row, we see that more games are tournaments than tutorial, while there are no practice games. From the right column, in which distributions of each categories separated by the game results are shown, we observe that the distributions are similar in patterns compared to the left column, and winning games are all slightly higher than lost games generally, following the overall trend of the dataset. This similar trends is intuitive as location and lobby should not have much influence on results. The small influence of game mode, on the other hand, might by due to hero imbalance.

Then, we take a deeper look into information of heroes and compositions. From Figure 4 below, we observe that most frequently chosen heroes are generally the same among all the compositions and among the winning compositions. The top five chose heroes are: Mirana, Phanton Assasin, Pudge, Legion Commander, and Juggermaut.

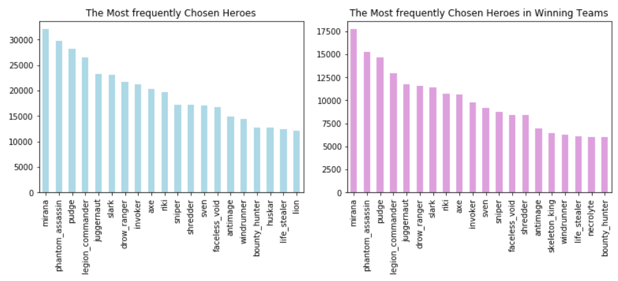


Figure 4. Most Frequent Heroes among all Teams and among Winning Teams

Moreover, winning rate is recorded for each hero. From Figure 5 below, we observe that heroes with highest winning rate are Ominiknight with 0.61 win rate, Elder Titan with 0.59 win rate, and Necrolyte with 0.56 win rate. Heroes with highest winning rates are fairly different from most frequently chosen heroes. This might be because that mostly chosen heroes possess necessary functions for a team composition so that every team has to pick them, while winning heroes are keys to victories. Another possible reason is that these heroes with highest winning rates are chosen for only a small portion of games, and thus produce biased result.

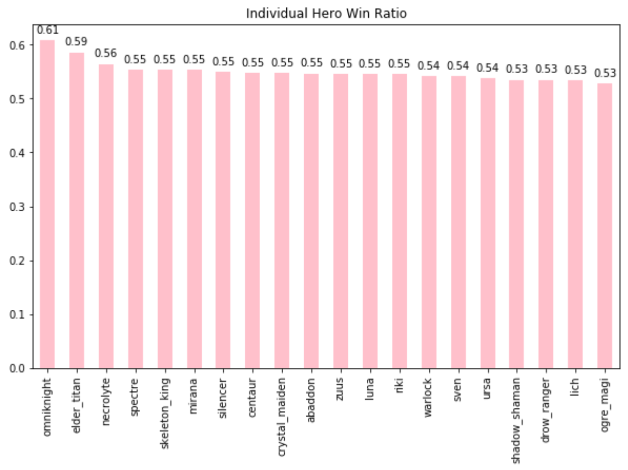


Figure 5. Win Rate for the Top 20 Heroes

Besides individual heroes, we also investigate hero pairs. From figure 6 below, we observe that the top chosen hero pairs are also very similar between all compositions and those among winning compositions. The top five hero pairs are: (Pudge, Phanton Assasin), (Mirana, Pudge), (Mirana, Legion Commander), (Mirana, Phanton Asssasin), and (Jauggernaut, Mirana).

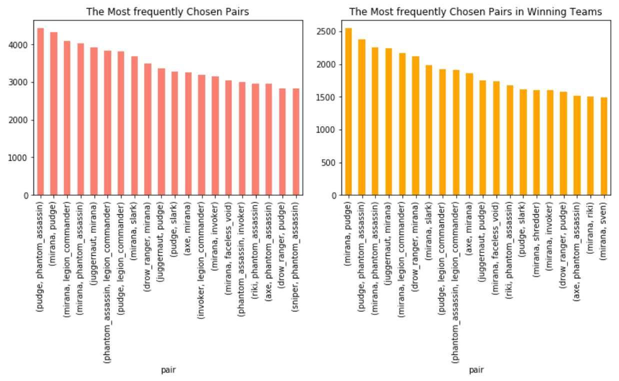


Figure 6. Most Frequent Hero Pairs

Again, winning rates are recorded for each pair and shown in Figure 7 below. The top five pairs with highest win rates are: (Venomancer, Omniknight), (Chen, Brewmaster), (Omnikight, Treant), (Death Prophet, Visage), and (Chen, Troll Warlord). Interestingly, Omniknight shows up in two of these pairs, which is consistent with the individual hero win rate ranking above. We also confirm that frequently chosen pairs are different from pairs with highest win rates, due to either or both of the reasons mentioned before.

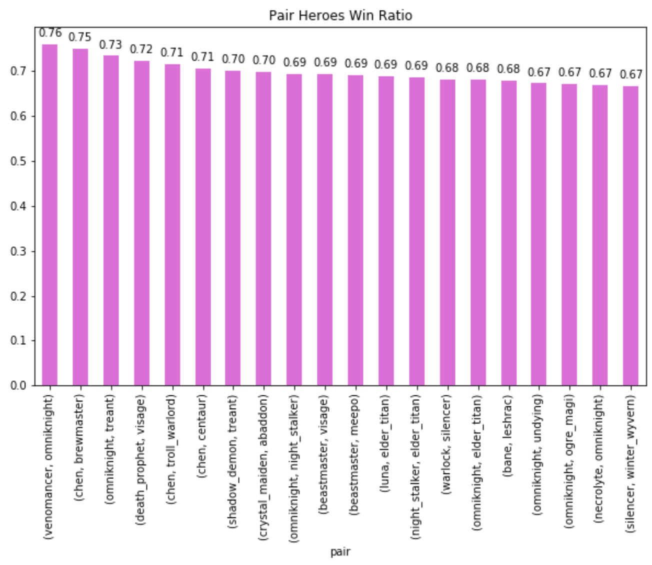


Figure 7. Win Rates for the Top 20 Pairs

After individual heroes and hero pairs, we investigate the whole five-hero compositions. The compositions that show up the most are: (vengefulspirit, beastmaster, luna, lycan, abaddon) and (dazzle, batrider, ancient_apparition, invoker, techies), both ten times. We realize that compositions must have been very different among all the teams so that the most chosen compositions only show up ten times. On the other hand, the compositions that show up the most among winning teams are: (vengefulspirit, beastmaster, luna, lycan, abaddon) and (axe, mirana, sven, lion, invoker), which show up ten times and four times respectively in winning teams. Note that the composition (vengefulspirit, beastmaster, luna, lycan, abaddon) wins all the ten games it shows up. We choose not to analyze the win rates for complete compositions as many compositions have 100% win rate due to small number of games they show up.

## 5. Predictive Task

In this paper, we will predict if the team wins when it chooses certain team composition competing the other team. We will use accuracy as our metric to evaluate our model. The predictive task will be interesting and practical since it can be applied in the DotA 2 gaming ban picks not only in personal use but also in professional games.

### 5.1 Feature Selection

As mentioned in exploratory analysis, we found the features other than heroes less interesting since the variance of winning result of each feature is more relatively consistent than the feature of heroes. To confirm this assumption, we built a naïve logistic regression feeding in features except heroes, and the accuracy ≈ 0.5, which nearly equals to random guessing in binary classification. Thus, our model will only consider the features of 113 heroes.

### 5.2 Data Preprocessing

The heroes feature in the original data is a 113-dimension one-hot encoded feature, with following condition:

$$\begin{cases} 0 \; the\ hero\ is\ not\ chosen \\ 1 \; the\ hero\ is\ chosen\ by\ this\ team \\ -1 \; the\ hero\ is\ chosen\ by\ the\ oppnents'team \end{cases}$$

Although the feature is preprocessed with one-hot encoding, we want to distinguish the difference between chosen by the target team and chosen by the opponents' team, so we split the feature into two sperate 113-dimension one-hot encoded features, noted as $X_1$ and $X_2$ respectively. $X_1$ represents the heroes chosen by this team, and $X_2$ represents the heroes chosen by the opponents' team with -1 changed to 1.

In other words, the original hero selection feature X is broken down into two matrices $X_1$ and $X_2$, in means of considering the general advantage of Radiant over Dire, is encoded as follows:

$X = [X_1, X_2]$, where

$$X_{1,ji} = \begin{cases} 1 \; if\ hero\ i\ is\ on\ Team\ 1\ during\ match\ j \\ 0 \; otherwise \end{cases}$$

$$X_{2,ji} = \begin{cases} 1 \; if\ hero\ i\ is\ on\ Team\ 2\ during\ match\ j \\ 0 \; otherwise \end{cases}$$

By default, we consider Team 1 as our target team. For evaluation purpose, we randomly shuffled $X = [X_1, X_2]$ and split the dataset into training and testing sets. The first 80% is used for training and the rest 20% is used for testing.

### 5.3 Baseline Model

Our baseline is a random-guessing model. It is reasonable and intuitive to use random guessing since the result of a DotA 2 match, being highly dependent on player's performance rather than hero selection, is more arbitrary than other binary classification tasks. The accuracy of such random-guessing model can be estimated by the distribution of the two classes in our training and testing dataset (52.7%, 52.6% respectively).

## 6. Model

As our task is a binary classification on a homogeneous categorical dataset, the following two classification algorithms are chosen to be evaluated: Logistic Regression Classifier, Decision Tree Classifier.

As we have observed, Logistic Regression is a popular choice when all the features available are categorical. Logistic Regression classifier does not have a problem of overfitting when training set gets large. Decision Tree, along with the tree-based ensembled algorithms, is also an intuitive choice for performing binary classification. While tree-based classifiers normally perform better and more efficient than other prevalent classification algorithms, Decision Tree classifier may greatly suffer from overfitting when training set gets large.

### 6.1 Approach I

Approach I is a supervised machine learning model, which performs feature engineering as follows: Assuming that the heroes are of different levels of difficulty to play and possess different levels of strength. Consequently, some hero selections can be more powerful than the other ones. Thus we propose to calculate the probability of the target team winning the match as the overall probability of the heroes they selected winning a match. Since the choice of heroes is exclusive, i.e. a hero can be on only one side, the target team getting stronger heroes indicates that the opponent team can only choose from the weaker ones. Thus, this overall probability of winning can be simplified to the average of the selected heroes' individual winning probability.

Consider a composite score ω of hero individual win rates for the target team's hero composition; ω is taken as the average of selected heroes' individual win rates α, calculated as the formula below:

individual win rate of hero i $\alpha_i = \frac{1}{2}\left[\frac{y \cdot X_{1,i}}{|X_{1,i}|_1} + \left(1 - \frac{y \cdot X_{2,i}}{|X_{1,i}|_1}\right)\right]$

composite score for match j $\omega_j = \frac{1}{5} \times \sum \alpha_i \times X_{1,ji}$

Here $X_{k,i}$ is a feature vector which encodes whether hero i has been selected by Team k. Hero i's individual win rate $\alpha_i$ is defined as the average of hero i's win rate when played by Team 1 and by Team 2.

Note that when calculating the composite score for a match, we always divided it by 5, which is the default number of heroes on one team, even though there're observations in our dataset that some teams had less than 5 heroes. In this way, if a team has less than 5 players, its ω score will be penalized.

Then we set up a threshold for prediction as an intuitive value for filtering winning probability, 0.5.

$$y_{pred} = \mathbb{1}(\omega - 0.5)$$

The training and testing accuracies of the Approach I model are, as would be expected, slightly better than the first baseline model, 56.2% and 56.2% respectively. As we would show later, accuracy has been traded for computational easiness in this approach.

**6.2 Approach II**

The improvement in accuracy in Approach I indicates that the target team's hero composition and relative hero strength could be potentially informative features. Thus, in Approach II, we performed a deeper and more complicated feature engineering, which is then used for training a Logistic Regression classifier and a Decision Tree classifier.

Inspired by report *Dota 2 win prediction*, we propose two new features, SR score and C score, which together evaluate the win rate based on a team's hero composition, i.e. whether the 5 heroes chosen makes a strong team. Unlike in Approach I, Approach II takes the opponent's hero composition and the interaction between heroes into consideration.

C score measures whether the target team's choice of heroes have a higher win rate when played against their opponent's choice of heroes. In other words, if the target team has chosen a hero combination that "counters" their opponent team well, they will be assigned a higher C score. Following the definitions discussed in previous sections, C score for a given hero selection matrix X is calculated as follows:

$$C = \frac{yX_1^T \cdot X_2 - yX_1^T X_1}{X_1^T X_2}$$

S score measures the "properness" of a given hero combination, i.e. whether the heroes chosen work well as a team. A good hero combination will be assigned a higher S score. S score is calculated as follows:

$$S_1 = \frac{yX_1^T \cdot X_1}{X_1^T X_1}$$

$$S_2 = \frac{yX_2^T \cdot X_2}{X_2^T X_2}$$

$C, S_1, S_2$ are all 113 by 113 square matrices, where each column and each row maps to a hero in DotA 2. The $(i,j)th$ $(where\ i,j \in [0,112])$ entry in those matrices records the C score when i is played against j or the S score when i and j appear on the same team.

SR score is composite score given by the difference between S scores of two teams, measuring whether the target team has a better hero combination than their opponent. Given hero selection matrix X, the two new features $C_x$ and SR score are calculated as follow:

$$C_x = X_1 C X_2^T$$
$$SR = X_1 S_1 X_1^T - X_2 S_2 X_2^T$$

The feature matrix we used to train the classifiers is in the form $[C_x, SR]$. The hero selection matrix $X$ is discarded to avoid potential information redundancy in our feature matrix as $C_x$ and SR together already encode the information of both interaction within a team and between the two teams; also in the means to make the training process more computationally affordable.

The Decision Tree classifier quickly overfitted, producing an initial training and testing accuracy of 99% and 52%. Our attempt at tuning the Decision Tree classifier by performing grid search cross validation on the training set was unsuccessful; the best training and testing accuracy was similar to that of Approach I. However, Logistic Regression classifier showed an improvement in testing and training accuracy, reaching 64.7% and 59.3% respectively. Tuning the Logistic Regression classifier in a similar fashion did not produce significant improvement. Thus we suspected that if we have exhausted the information in our original hero selection dataset. To confirm our assumption, we proposed Approach III, which relies on deep learning algorithm to mine for the potential hidden pattern we missed in the training set.

In this approach, although computationally heavier feature engineering is involved, it has in turn produce higher test accuracy.

### 6.3 Approach III

Approach III directly uses the algorithm of Multilayer Perceptron, a deep learning technique, via Keras with original dataset (X in Data Preprocessing section). We use ‘Adam’ algorithm as our optimizer and binary cross entropy as our loss function since it is a binary classification. The layer structure of the model showed in below:

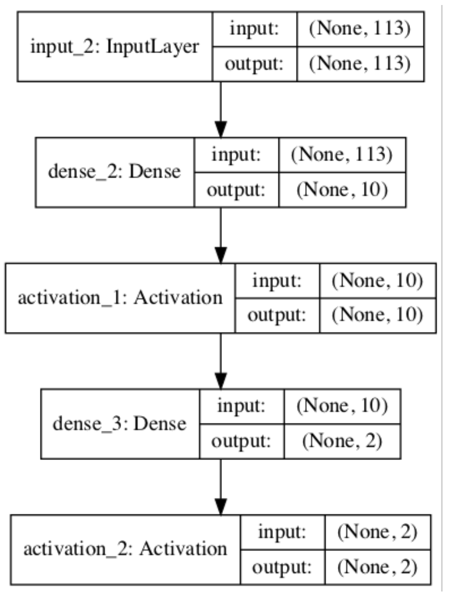

input_2: InputLayer | input: (None, 113) | output: (None, 113)
dense_2: Dense | input: (None, 113) | output: (None, 10)
activation_1: Activation | input: (None, 10) | output: (None, 10)
dense_3: Dense | input: (None, 10) | output: (None, 2)
activation_2: Activation | input: (None, 2) | output: (None, 2)

*Figure 8. Model Layer for Approach III*

The input layer is the original X of hero layer, and we use Dense layer with shape 10 as our hidden layer, which means it has 10 latent factors that may influence the prediction result. Then we use RELU to activate the layer. Finally, it has a shape 2 output layer with Sigmoid activation in the end. We train our model with 2 epochs and 8 batch size. The training curve of Approach III is shown below:

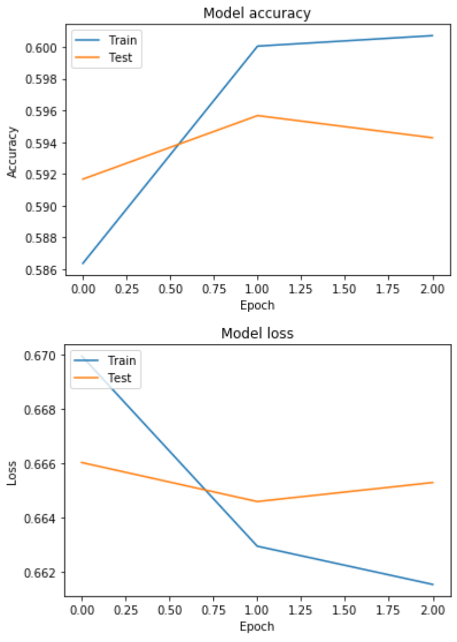


*Figure 9. Training Curve for Approach III*

From the plot, we observe that, the model will be overfit after epoch 1. We tried to optimize the result by changing batch size and adding dropout, but it would be still overfit after epoch 1. Then we make an assumption that the dataset is not very sufficient for deep learning.

The testing accuracy of the model is significantly better than the baseline model with 59.7% with 1 epoch.

### 6.4 Approach IV

Approach IV also used Multilayer Perceptron via Keras. Rather than using original dataset in Approach III, Approach IV uses the preprocessed hero feature ($X_1, X_2$ in Data Preprocessing Section). We also use ‘Adam’ as our optimizer and binary cross entropy as our loss function. The layer structure of the model is explained as follow:

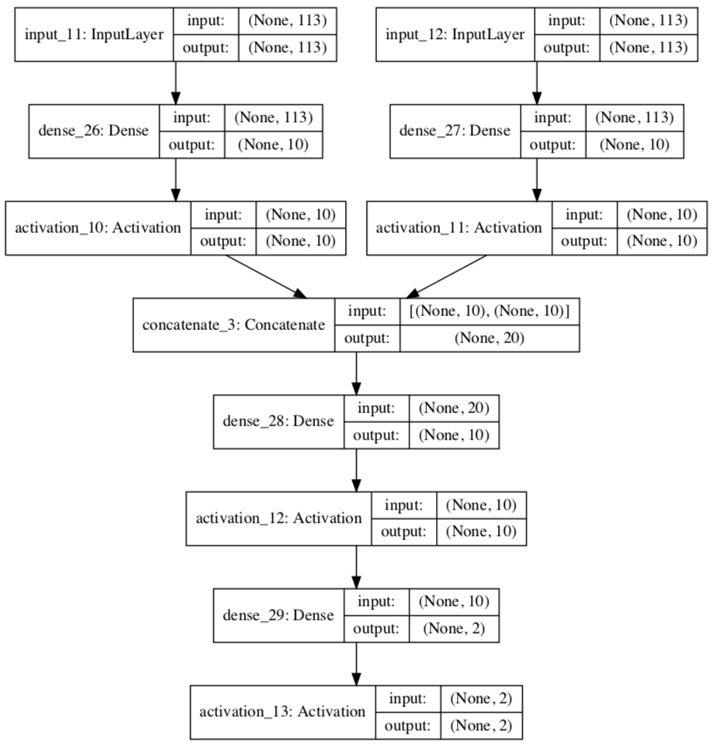
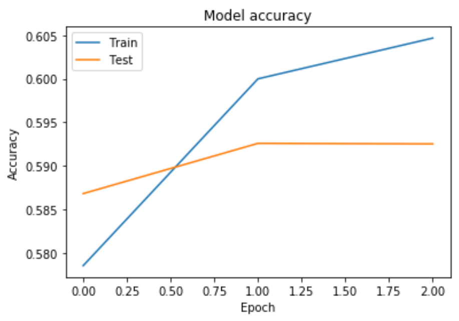


*Figure 10. Model Layer for Approach IV*

The input layers are the input $X_1, X_2$, and we also used Dense 10 as our hidden layer. After the hidden layer, we use RELU to activate the hidden layer. Then, we concatenate two layers together, and use Dense 10 as our hidden layer of the concatenation layer with RELU activation. Finally, it has a shape 2 output layer with Sigmoid activation in the end. We train our model with 2 epochs and 8 batch size. The training curve of Approach IV is described as follow:



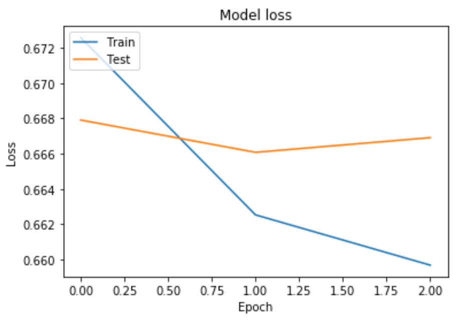


*Figure 11. Training Curve for Approach IV*

From the plot, we observe that, the model will be overfit after epoch 1. We tried to optimize the result by changing batch size and adding dropout, but it would still overfit after epoch 1. Then we make assumption that the dataset is not very sufficient for deep learning.

The testing accuracy of the model is 59.4% with 1 epoch, which is roughly the same with Approach III. Both Approach III/IV are computationally easier without feature engineering process, but produce slightly lower accuracies compared to Approach II.

### 6.5 Final Model Selection

Finally by comparing the test accuracies of all four approaches, we select Approach II, in which Logistic Regression model and an feature representation of individual hero influence as well as synergy among heroes are involved, as our final proposed model. Although it is computationally harder, it produced the highest test accuracy without overfitting to training set.

## 7. Conclusion

Although we have chosen Approach II as our final model, we found that Approach II to IV all produce similar test accuracies near 60%, probably due to the fact that we have reached the potential limit of current data. Even if we take the synergy and interactions among heroes into consideration besides individual influence in these approaches, the accuracy fails to improve much. This result indicates that such accuracy might be the ceiling considering the data we have. It is also a reasonable result as a competitive game should not only depend on hero compositions, but also players themselves. Moreover, eSport brings excitement only when the results are uncertain. A high accuracy of our models would only suggest hero imbalance, as we can accurately predict results merely by hero compositions without player information.

Comparing two kinds of feature representations used, one considering only individual hero influence and the other also considering synergistic effect among heroes, the latter offers better prediction. This is because how heroes work together and against each other are the keys in a team-based game like DotA 2, which is also the reason that leads to lower accuracy in Approach I compared to the other ones. Thus, mere individual hero influence is not an optimal choice for feature representation.

Model parameters are not an important factor in our case, as tuning process shows that no significant improvement is reached by changing them, again confirming that we have reached the potential limit of data. In Logistic Regression, C parameter serves as a regulator of overfitting. In Deep Learning, Dense layer shape represents number of latent factors we expect, while batch size controls learning rate.

## 8. Future Work

In the future, we suggest including player information besides hero compositions when training models to predict team-based game results. As the report *To win or not to win?* has shown, adding player information could likely boost the accuracy. The suggestion is also reasonable because different play styles, along with the synergy and interactions among them, can largely affect a competitive game.

## 9. References

Agarwala, Atish. “Learning Dota 2 Team Compositions.” (2014).

Dua, D. and Graff, C. (2019). UCI repository of machine learning databases.

Kalyanaraman (2014). To win or not to win? A prediction model to determine the outcome of a DotA2 match.http://cseweb.ucsd.edu/jmcauley/cse255/projects/KaushikKalyanaraman.pdf

Kinkade, N., Jolla, L., Lim, K.: Dota 2 win prediction. Technical report, University of California, San Diego (2015)

McDonald, Tim. "A Beginner's Guide to Dota 2: Part One – The Basics". *PC Invasion*. Archived from the original on August 11, 2016. Retrieved August 1,2016.