The odds are with us. Or are they?
A regression approach to predicting a winner in Dota 2

Caleb Holdener
Belmont University
Email: Caleb.Holdener@pop.belmont.edu

Abstract – In this paper multiple models were created and presented in order to predict the outcome of an multiplayer online battle arena computer game called Dota 2. Similar work has been done, however most of these models and datasets are extremely old at this point, causing them to be inaccurate for the current game state.

The details associated with building these models are presented, along with how the data was gathered and processed. By presenting more current issues and problems associated with modern Dota prediction algorithms, further progress can be made to develop more accurate prediction algorithms.

I. Introduction

Dota 2, also known as Defense of the Ancients 2, is a MOBA (multiplayer online battle arena) game, which is a subgenera of strategy games. In Dota there are two teams of five players called Radiant and Dire. Each player controls, in most cases, a single unit called their Hero in order to destroy the opposing teams base, also known as their ancient. Players can choose from 112 different Heroes, each with different designs, strengths and weaknesses. Players select their Hero during a drafting phase at the beginning of the game. Once a Hero is picked, then no other player can choose that Hero. Because of this, and the uniqueness of each hero each match is unique. The goal of the project is the generation of a machine learning algorithm, that will be able to predict the outcome of a Dota 2 match based only on the Heroes that were drafted for each team.

II. Related Work

In the past, others have attempted to create a prediction algorithm for Dota 2 matches. The most well documented model used a dataset from 2013, meaning that a more recent study has yet to be done and well documented. In 2016, a TensorFlow model was created for Dota 2 by Mark Dunne [1], who used a dataset from a Stanford project that used a logistic regression model [2]. The TensorFlow model was able to achieve a 72.21% test accuracy, while the logistic regression model obtained a 69.8% validation accuracy. In a paper by Nicholas Kindade and Kyung yul Kevin Lim, at the University of California, San Diego [3], the dataset used was collected in 2015 which was two years after the other papers. They were able to achieve a 64% accuracy by using only the hero matchup data. This shows a potential negative trend in accuracy using only matchup data.

Various organizations, programs, and websites have also aided the Dota 2 data science and analytics scene. One of the most notable organizations/websites is OpenDota [4], which is an open source Dota 2 data platform. OpenDota also offers a well-documented web API which allows for the collection of match data provided one has an API key. This API was used in our data collection efforts to parse and gather the required match data.

III. Dataset

A. Data Collection

In order to perform predictive analytics, data must first be collected. The data that had to be collected must have included which players are playing on each team and what Heroes those players are playing. This was accomplished by first gathering a large number of match ID numbers. We gathered a range of match IDs (5214334377 to 5214534376). These ID numbers can be used to further query the needed features. This could be accomplished with either the OpenDota query API, or the Steam (the company who owns Dota) API. The OpenDota API was utilized for this dataset. The advantage of OpenDota API platform, is the rate limit of 1200 calls per minute, versus the rate limit of 60 calls per second with the Steam API. By using the Requests Python Library, calls were able to be made to the API in order to return full match data. during a single patch. This gave a raw dataset that could then be processed.

B. Data Processing

This data then had to be filtered and the correct features appended to a new dataset that would be used in the final dataset. Because the Hero data is categorial, and technically a part of 2 groups (the two different teams), in order to perform one singular regression, a new dataset containing dummy variables was created. This gave a matrix, where each row was a match and the X represented a possible hero. This is typically described as one hot encoding.

IV. Methodology

A logistical regression and a deep learning model were utilized to predict the match output based on the binary matrix described in section III B. This matrix was used to train our model.

A. Logistic Regression

A logistic model was chosen due to its ability to predict a binary output, a simple model was created to accomplish this based on this.

a. Model Creation and Tuning

The model was creating by utilizing the scikit library. Initially by using the default cross validation, upwards of 1.2% accuracy score was lost during validation. In order to better train our model a variety of parameters were added to attempt to increase the accuracy, specifically by increasing the C parameter, and by changing the cross-validation method to StratifiedKfold. However, the highest validation score I could achieve was within 1.1% of the model.

b. Analysis

Based on the model created we can conclude that currently in Dota, pure draft without any other factors (features) is not very important to the outcome of the game. While some heroes had a high impact on the win rate, because of the lower sample, with both individual heroes, and overall matches.

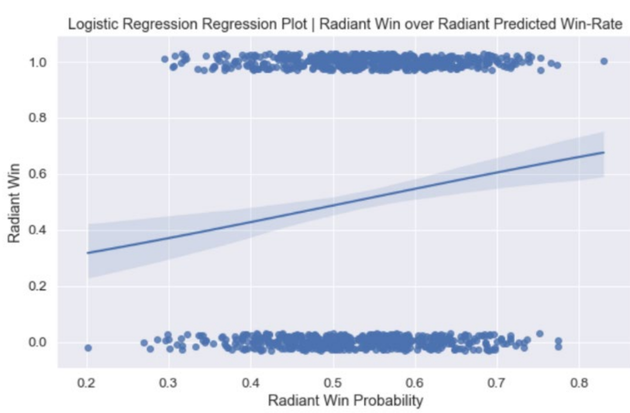

Fig. 1. Regression Radiant Win Actual over Predicted Win-rate

B. Deep Learning

A deep learning model was chosen because it is an emerging field of computer science research. TensorFlow is a leader in deep learning model creations and their Keras API allows for highly customized models when compared to other libraries.

a. Model Creation and Tuning

The model was created using Keras API in TensorFlow 2.0. A network of two hidden dense layers was created and an output layer using the Sigmoid Activation function, which gives a range between 0 and 1, perfect for predicting the probability of a binary output. When analyzing initial model prototypes, it became obvious that the model was overfitting the training data, therefor leading to lowered validation accuracy. To better tune our model and prevent overfitting, an early stopping callback which monitored validation loss was introduced. In addition, model complexity was lowered, leading to a more accurate and faster model.

b. Analysis

Using a tuned deep learning model, I was able to achieve, on average, an even – if not slightly higher, accuracy compared to the logistic regression mode. Even with a relatively simple model, we were able to achieve a consistent 56% accuracy, which was 4% higher than blind guessing the winner.

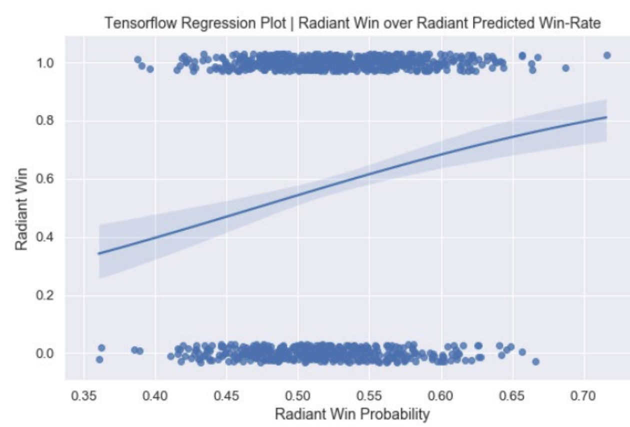

Fig. 2. TensorFlow Radiant Win over Predicted Win-rate

V. Conclusion

In this report, logistic regression and deep learning models were used in an attempt to predict match outcomes based only on the heroes picked at the start of the game. Using regression, an accuracy score of ~55% was achieved, while deep learning gave an accuracy score of ~56%. When examining the confusion matrix that each model gave.

Comparing the models, there is a distinct benefit of using a deep learning model, especially with the large amount of data. TensorFlow also offers a large amount of customizability that has the potential to better combat under and overfitting. One disadvantage of TensorFlow is the required background knowledge required to create an optimal model is much larger compared to a logistical regression model using a library such as scikit-learn. Another disadvantage is the processing time, the logistic model was incredibly fast to process while the deep learning model, in comparison, took much more time and processing power. For a simple model, creating a logistic regression model would be the better idea, due to the low time commitment and faster processing speed. However, if a more complex model is required, or the amount of data needing to be processed is exponentially higher, then a deep learning model could be beneficial.

The accuracy score of ~55-56% was only 3-4% higher than predicting the Radiant each time (which had an average win rate of 52%). This is lower than similar models utilizing older datasets. Two main options that could cause this, the dataset used was flawed, or Dota has become more balanced overall. For our model, it is most likely and a mix of both options. Not enough information was extracted during the data gathering process, which meant filtering options were decreased. Because of less match filtering, lower quality games got included in our dataset. These factors can include players abandoning the game, games played by matchmaking bots, or even games including smurfs (a player intentionally playing at a skill level lower than their own) and boosted players (players playing at a skill level higher than their own). In recent years, there has been an increasing number of these players who play at a skill level not of their own and in these games typically hero choice has a much lower impact of the outcome of the game.

Given the results of our model however, it can be concluded that the current model (given the current dataset) is only an average model for predicting win-rate.

VI. Future Work

While the model was simple, there is great expansion that can still take place in order to further increase the accuracy and efficiency of the prediction algorithm. This model looked strictly at both the dire and radiant teams

Addressing the multicollinearity between heroes could be done to increase model accuracy. Some heroes, when picked together, have a much higher chance to lead to a victory. In regression models, this multicollinearity could cause problems, so additional processing would have to be done in order to compensate for this connection. Manually addition combinations of two heroes to the feature set could compensate for this multicollinearity.

Better cleaning of the data, along with more data processing would also be needed. Matches were not checked for full completion, whereby one player abandoned and forced the apposing team victory. By not ensuring these games were absent in out dataset, accuracy could be lowered. In addition, all the matches, no matter what skill level, were combined. The viability of some heroes depends on the skill level (MMR) that the hero is played in. By separating matches by skill level, the model could be tailored for a specific MMR range and be overall more accurate.

Overall there are many various avenues for further exploration to take place and for accuracy to be increased.

References

[1] M. Dunne, "Tensorflow Dota Predictor," *Github*, 2016. [Online]. Available: http://markdunne.github.io/2016/02/07/TensorFlow-Dota-Model/#Tensorflow-Dota-Predictor.

[2] K. Conley and D. Perry, "Using Machine Learning to Recommend Heroes for Dota 2 Matches," Stanford, 2013.

[3] N. Kinkade and K. yul K. Lim, "DOTA 2 Win Prediction," San Diego, 2015.

[4] OpenDota, "OpenDota," *The OpenDota Project*, 2020. [Online]. Available: https://www.opendota.com/.