Predicting DotA 2 Winning Team based on (Draft Selection) Hero Selection using Machine Learning Algorithms

Manorath Bajaj, Rajeshree Kale
Information Systems, Northeastern University

ABSTRACT

In this paper we have implemented machine learning algorithm for the prediction of the winning team based on the draft (hero selection) before the game is played by using the features that we have engineered. We are using a dataset from UCI repository [1] which has 100,000 matches in which 90,000 will be used for training the model and 10,000 are used for testing the model using techniques like Logistic Regression, Random Forests, Gradient Boosting and Multilayer Perceptron. Multilayer Perceptron model which was trained with pick rates worked best with accuracy 59.59%.

Keywords-Dota 2, Logistic Regression, Random Forest, Gradient Boosting, Multilayer Perceptron, Feature Extraction, Feature Engineering

INTRODUCTION

DOTA 2 is a popular online MOBA (Multiplayer online battle arena) game that is played between two teams consisting of five players with each team defeating their opponents. Each Player chooses from over 115 heroes. [2] The main objective of the game is to destroy the opponent teams, called the "Ancient". Each hero has unique abilities. The abilities of a hero can be used to counter a different hero. (For example: A hero named Riki deals more damage when he hits the opponents from behind but in comparison, a Bristleback takes less damage when gets hit from the opponents from behind) and hence there is a draft advantage to the team which has smarter picks against the opponents.

The research has already been done on DotA 2 to predict which team will win based on the skill set of the players. This was done using the Naïve Bayes and XGBoosting algorithm. [3] One more paper that was used as a reference had predicted the winning team based on the experience gain (XP) per minute, gold per minute (GPM), kills, deaths and assists. Also, the paper compared the hero kills per death and the hero game duration. They created two different win predictors. In which, the first win predictor was used for full post-match data and the second predictor was used for hero selection data. The prediction was done using logistic regression and random forest classifier wherein the accuracies were nearly the same around 99%. (Note: These results were calculated at the end of the game). [4]

METHODS

Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. We have engineered the following:

a) Winrate: All the heroes in DotA 2 don’t have the same win rate. Some heroes win way more games than other heroes. The win rate values are normalized to fit between 0 and 1. Then multiplied to training dataset. This features value ranges from -1 to 1.
b) Pickrate: Some heroes are more popular than other heroes among players. This might affect the outcome of the game and hence taken as a feature. Extracted in the same way as winrate
c) Winrate*pickrate: In our model we selected these features named win rate and pick rate and estimated their products.

Win rates and pick rates were calculated using the algorithm mentioned below:

```
iterate over hero columns
{
    iterate over training data
    {
        if row is equal to win column
        {
            increase win count and increase total count
            if row is 1 {
                increase radiant win and total radiant count
                }
            else
            {
                increase dire win and total dire count
            }
        }
        else if row is equal to win or 0
        {
                increase total games count
                similar check for radiant and dire total count
        }
     // for Pickrates :
         if hero column is not 0 {
         increase pick count and total pick count
         }
         else
         {
         increase total pick count
         }
     find winrates by dividing winrates by total count
     find pickrates by dividing pickrates by total count
     store in an array
     }
}
```

Learning Algorithms

1. Logit classification:

The logistic classification model (or logit model) is a binary classification model in which the conditional probability of one of the two possible realizations of the output variable is assumed to be equal to a linear combination of the input variables, transformed by the logistic function.[5]

2. Random Forest:

Random forests are a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.[6]

3. Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems. It produces a prediction model in form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.[6]

4. MLP

The field of artificial neural networks is often just called neural networks or multi-layer perceptrons after perhaps the most useful type of neural network. A perceptron is a single neuron model that was a precursor to larger neural networks. It is a field that investigates how simple models of biological brains can be used to solve difficult computational tasks like the predictive modeling tasks we see in machine learning. The goal is not to create realistic models of the brain, but instead to develop robust algorithms and data structures that we can use to model difficult problems.

The power of neural networks come from their ability to learn the representation in your training data and how to best relate it to the output variable that you want to predict. In this sense neural networks learn a mapping. Mathematically, they can learn any mapping function and have been proven to be a universal approximation algorithm. The predictive capability of neural networks comes from the hierarchical or multi-layered structure of the networks. The data structure can pick out (learn to represent) features at different scales or resolutions and combine them into higher-order features. For example from lines, to collections of lines to shapes[7] . We found the best hyper parameters for baseline using the above algorithms and applied them on the features to predict the results.

RESULTS

Tables

We found the best hyper parameters for each model using the baseline variables and applied them on the features to predict the results. Based on the performance of all the features in the dataset using the learning algorithms, we got the accuracies mentioned in the table below:

	Features	Win Rate	Pick Rate	Win Rate*Pick Rate	Baseline
Algorithms	Logit Classification	56.54%	56.53%	56.53%	56.04%
	Random Forest	53.99%	53.62%	52.74%	52.56%
	Gradient Boosting	53.19%	52.9%	52.74%	52.34%
	MLP	53.12%	60.03%	59.86%	53.26%

Table 1: Displays the results of the accuracies based on the validation data

	Features	Win Rate	Pick Rate	Win Rate*Pick Rate	Baseline
Algorithms	Logit Classification	57.21%	57.21%	55.87%	57.06%
	Random Forest	53.92%	53.96%	53.45%	53.08%
	Gradient Boosting	53.47%	53.47%	52.92%	52.11%
	MLP	53.04%	59.59%	59.65%	53.26%

Table 2: Displays the results of the accuracies based on the test data

Graphs

The performance of the models can be seen based on the box plots and bar charts.

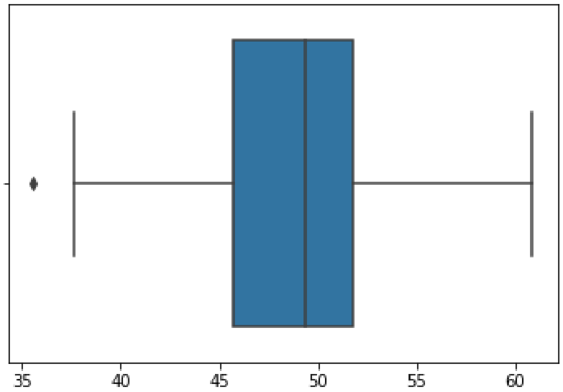

Figure 1: Displays the boxplot for the hero win rates

The above boxplot represents a median of the win rate is around 48%. Most heroes have their Win rates in (40% – 60%) bracket, with exception of Heroes like ‘IO’ who has a terrible Win Rate in public servers but is almost banned every game in a professional match of DotA 2.

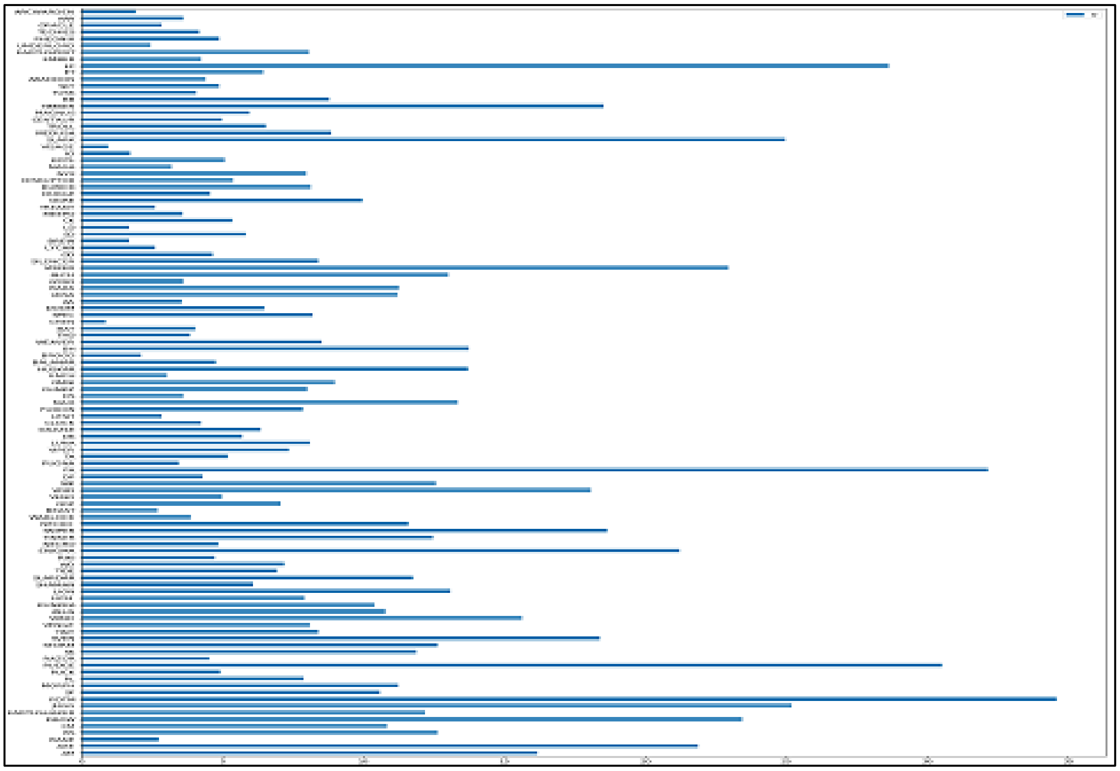

Figure 2: Displays the bar graph for the hero pick rates

The bar graph displays the hero pick rates in which the hero with the highest pick rate is Earthshaker and the hero with the lowest pick rate is Io.

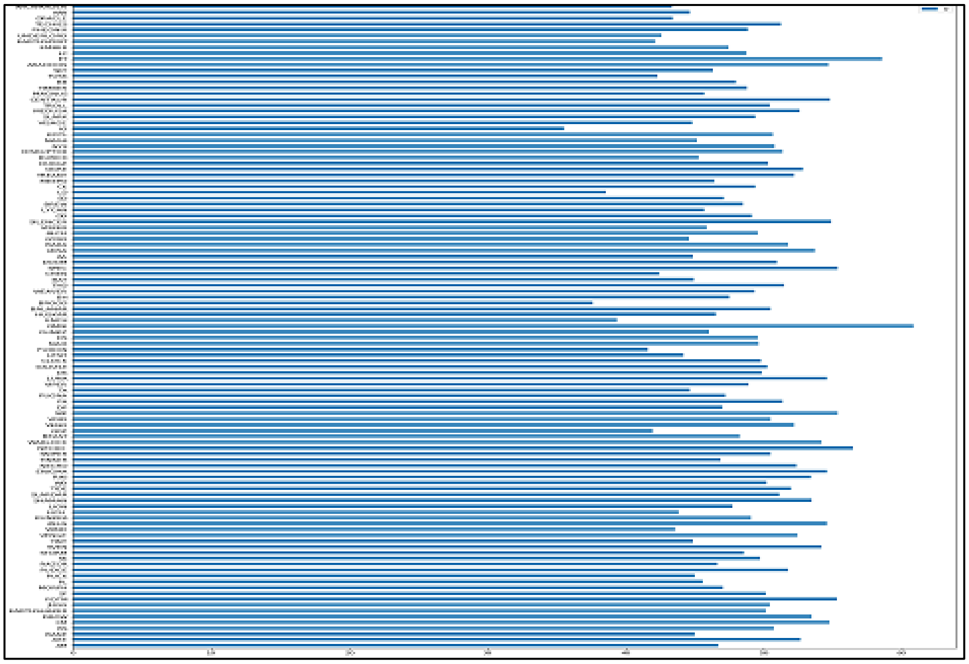

Figure 2: Displays the bar graph for the hero win rates

The bar graph displays the hero win rates in which the hero with the highest win rate is of POTM (Mirana) and the hero with the lowest win rate is Io.

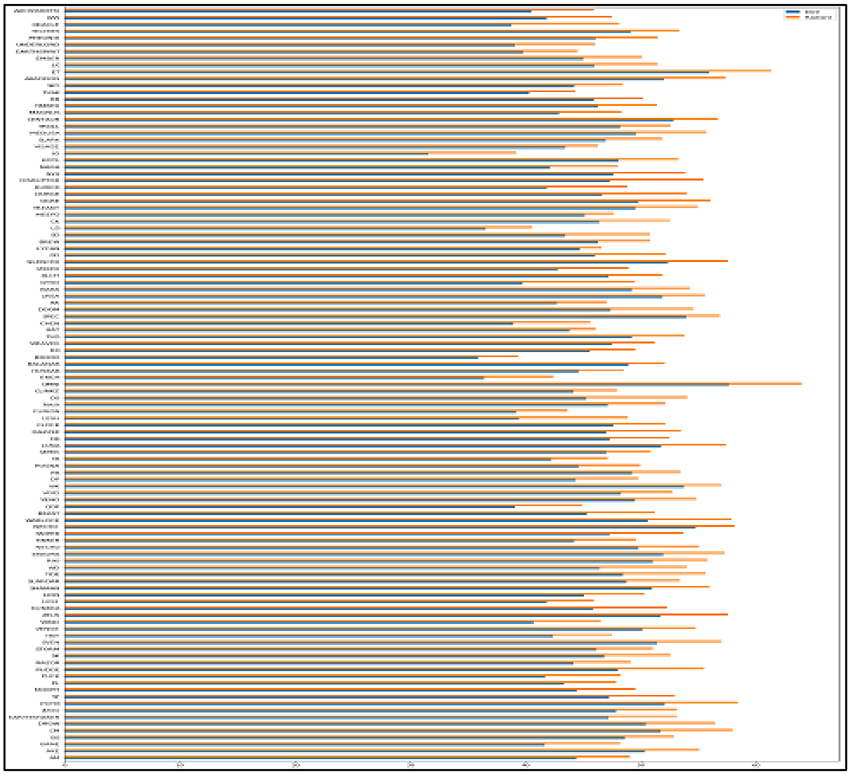

Figure 2: Displays the bar graph for radiant & dire win rates

The bar graph displays the radiant and dire hero win rates in which the hero on the radiant side has higher probability for winning the game while the dire side has a lower probability of winning the game.

CONCLUSION

The only learning algorithm that had a significant increase using our features was Multilayer Perceptron which had the best accuracy with the model trained with pick rate feature variable as seen from Table 2. Furthermore, we were planning to add side-based win rates as features but by graphical observations we found out that all heroes have clear radiant advantage by almost the same amount. The learning algorithm which had the least effect from our feature variables was logit classification, but it has accuracy better than Random forest and Gradient Boosting algorithms. All classifiers (other than Multilayer perceptron) heavily favor radiant to win, this discrepancy can be seen on the confusion matrix where radiant classification error rate is almost negligible and dire classification error rate ranges from 95 – 99.99%.

REFERENCES

1. https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results.
2. https://en.wikipedia.org/wiki/Dota_2
3. Semenov, A., Romov, P., Korolev, S., Yashkov, D. and Neklyudov, K., 2016, April. Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. In International Conference on Analysis of Images, Social Networks and Texts (pp. 26-37). Springer, Cham.
4. Kinkade, N., Jolla, L. and Lim, K., 2015. Dota 2-win prediction. Technical Report. tech. rep., University of California San Diego.
5. https://www.statlect.com/fundamentals-of-statistics/logistic-classification-model#hid3
6. https://en.wikipedia.org/wiki/Gradient_boosting/random forest.
7. https://machinelearningmastery.com/neural-networks-crash-course/