Dota 2 Matches Important Features for Prediction

Siddhesh Padwal

University of Denver

Siddhesh.Padwal@du.edu

**Abstract**

In this paper, I attempted to predict the winners of a Dota 2 match using some of the features in the Dota 2 Matches dataset from kaggle. The first attempt it's made by making use of the players, and the character each player has chosen And the second model uses 10 features taken directly from the players dataset, Which contains information descriptive of what happened during the match. For both methods, the algorithm used was the random forest classifier.

1. **Introduction**

Dota 2, Defense of The Ancients it's one of the most popular Multiplayers Online Battle Arena, also known has MOBAs developed by Valve. The game has more than 10 million players, that play the game actively and its last championship had a prize pool of $24,787,916.

The game has three lanes and consists of destroying the Ancient of the opposing team while defending your own, toIrs and barracks are the other structures that I must destroy to reach the Ancient, and to achieve that the two teams of compose a match have to select their characters. Both teams namely Radiant and Dire are composed of 5 characters, called heroes, which are selected at the beginning of every match. Each hero has a set of unique abilities and skills, which along with the three categories at which they belong dictate their Iakness and strengths. The combination of those heroes as Ill as the skill of the player, playing that hero play a major goal on whether one team has better chances of winning against the opposing team.

In the game, each hero belongs to a category and has a different nature, the nature of the heroes in the game are

* Intelligence - is the [attribute](https://dota2.gamepedia.com/Attribute) that grants mana, mana regeneration amplification and spell damage amplification to every hero, and additionally magic resistance and attack damage to intelligence heroes.
* Agility- is the [attribute](https://dota2.gamepedia.com/Attribute) that grants armor and attack speed to every hero, and additionally movement speed and attack damage to agility heroes.
* Strength- is the [attribute](https://dota2.gamepedia.com/Attribute) that grants health and health regen amplification, and additionally status resistance and attack damage to strength heroes.

And the categories are divided in 5 mains namely:

* Carry- Type of hero that become stronger for games longer than 40 min and are very items dependents.
* Support - type of hero that should protect the carries until late game and are not very dependent on items.
* Initiator - type of hero that can get involved in team fights early on the game have a good advantage against other heroes, and usually initiate the team fights.
* Ganker- type of hero that are good in one-on-one combat have a great chance of winning if they initiate the fight.

A good balance betIen teams generally dictate the game therefore sometimes it can be overruled by a better coordination of the opposing team.

A couple approaches using machine Learning algorithms have already been proposed therefore most of them lack in some of the components that are key to make a good prediction.

**2. Dataset**

The dataset used is Dota 2 Matches, this dataset contains several files with data on 50000 ranked ladder Matches.

To select the right model to set up our prediction problem the following information was taken from the files:

* Radiant\_Win- which specifies whether the radiant team won or lost a match.
* Hero\_id - which specifies which hero was chosen from a hero pool of 112 Heroes
* Player - who played the match and its stats, such has items acquired, gold earned and lost, number of kills and number of deaths.

Since I only had data for 5000 matches that I had to split for both training and test no reduction of the data was made, therefore it would be ideal. Reducing games with short duration and matches in which a player has disconnected, could have improved our observations.

The dataset also did not include the roles of each of the players and their statistics for that specific patch. That information was relevant because the stats and advantages of one hero over the other change on each update.

**3. Related Work**

Setting up a prediction problem Dota 2 [1], it’s one of the approaches taken by [Phalaris](https://www.kaggle.com/devinanzelmo) on Kaggle using the same dataset, therefore on its approach he only gets an efficiency of 50% which it's not much better than a random selector. And this happened because he uses the player's history to make the prediction, that's not a good approach because the player it's not one of the most important features to consider in a game, because people get payed to play in another's person account, people get matched with bad teammates that will not take the game seriously and this will decrease greatly the probability of that team winning the game.

Another related work I found on the [2] Hero picking and much result, on this work a better approach was made, they have considered the heroes picked to predict the result and the result was slightly improved from 50% to 53% using random forest classifier.

**4. Task Definition**

The task intended with this project, was to evaluate what features are metrics in order to predict the which team won a Dota 2 match and have an accuracy of at least 80%.

In a Dota 2 Match there it’s not a good way of predicting the winner before the match starts, and that's where, most of the works done predicting the winner of a match fail. Most work done only consider player, and hero chosen by that player therefore these two features cannot be used alone and expect a good prediction. And that is exactly where I started, by analyzing if those two combined would be enough to get a good prediction.

My approach was different in, the sense that I used some features that would depend on the performance of each players with the hero against that specific team. And those features are:

* Gold
* Experience Points, how much XP was earned per minute
* Kill, Deaths and Assists

Using those features I expected our accuracy to increase at least by 20% from what I got in the previous interaction.

**5. Tools And Infrastructure**

The programming language used to complete task was python. I made use of the SciKit Learn or SKlearn which is a machine learning library for python, which allows me to use their implementation several machine learning algorithms.

From SKlearn, I made use of the Random forest Classifier from the Ensemble package, in order to use the random forest algorithm, Accuracy Score from the Metrics package, and from the Support Vector Machine (SVM) package I used SVC, Support Vector Classification to apply an RBF classifier and see the accuracy.

Three other important libraries Ire used namely CSV, which alloId to load the data from the .csv files, Pandas, which alloId to create data frames to store the data taken from the csv files, as Ill as create new datagrams and Numpy which is a python library which allows me to deal with N dimensional arrays and use linear algebra operations.

The pre-processing done in the data was the creation of new dataFrames containing only the features that I considered to be more important to the task at hands considering our experience as a metric for feature selection. In the data there Ire 72 features, therefore only 15 of those features are relevant. Namely:

* Account\_ID - indicates the player
* Hero\_ID - represents the hero picked by the played.
* players\_Slot - indicates in which team the player is [Radiant, or Dire].
* Gold- which has three sub features
  + Gold-Total amount earned.
  + Gold\_spent- How much was spent.
  + Gold\_per\_min -Gold /Min.
* Kills - Represents how many times a player has killed an enemy hero.
* Deaths- Represents how many times a player has died to an enemy hero.
* Assists- Represents how many time a player has participated in an enemy hero death.
* Items- Correspond to 6 other features which are components that the players can acquire with the money they make as the game progresses. They improve the performance of the hero.

And out of those 15 features I decided to use only 10 because of the computation time required for it to run the algorithm.

**6. Model Selection and Assessment Metric**

To predict the winners of a dota-2

Match, I have used two statistical models on the dataset which I have found on kaggle.

1. Random forest
2. Soft SVM (using kernel)

Model selection:

For model selection, I used validation set as the test data. I used two models on training data and test the accuracy on validation data to predict the right model.

1. Random forest

As I have many features in the dataset, I first decided to go with random forest using n number of trees.

While using random forest algorithm, I used all 10 features from the player.csv file which contains all the detail about the players. I choose first to use random forest, so that I can get idea about which features are important to get the maximum accuracy.

I divided training set into 75 percent for training and 25 percent for test. I used splitting of data in 75-25 after trying the combination of 60-40 and 80-20. I got the best accuracy with 75-25. I converted our target variable into 1 and 0 where 1 represents radiant team won and 0 represents radiant team lose.

To represents right complexity of the model I varied the n\_estimator value which represents number of trees to see which one will give us the great accuracy.

With **n\_estimator =1**

Considering n\_estimator=1 and with the given features I have selected, I got 85.86% accuracy. I decided to check the accuracy with varying the parameter.

With **n\_estimator =100**

By varying our parameter, I got exceptionally Ill results. I got 87.21 accuracy on validation set and I finally set the parameter to 100.

Important features I got using Random Forest:

The main reason to use this model as it is simple and effective on the top of that I found the best features I should use to predict the outcome.I found each features are contributing equally with ‘player\_slot’ and ‘gold’ are must features to include while predicting.

The problem I face:

Even though finding the best accuracy with this model on validation set, I could not use this model on test data I got from kaggle dataset as it contains very less features in the data. Because of lack of features in test data, I unable to use this model, So I decided to use model with the features available in the test data I got from the kaggle dataset.

As test data contains only 3 features, I build a model with the given 3 features to predict the outcome with n\_estimator=100 where I got the best result with 10 features.

With only 3 features I got accuracy of 51.26% which is very poor as compared with the accuracy I got with 10 features.

So I decided to use another model.

**2. Soft SVM**

I use Soft SVM with the 3 feature data and the rbf kernel. To choose right complexity I vary the C parameter to get the good accuracy.

**With C=1**

I trained the model with C=1 and using ‘rbf’ kernel. I got the accuracy of 52.11%.

I decided to vary the parameter to get the better accuracy.

**With C=0.01**

I trained the model with C= 0.01 and using the same kernel still I got the same accuracy of 0.01.

I tried using other kernels also, but I got the best accuracy with rbf kernel using soft SVM model.

**Model Assessment and Selection:**

Comparing both the model on validation set, I decided to choose model with Random forest algorithm based on following observations:

The accuracy with both model is almost the same but in Soft SVM even though I tried to change the parameter I did not see any improvement and the amount of computational time is much higher than the random forest algorithm. So, I select the model with random forest algorithm to predict the output on test data.

I predict on test data with this model and I got the accuracy of 48%. The best thing of this model is if I increase number of trees the accuracy also increases. I increased the n\_estimator to 1000 and I got 49% accuracy.

**7. Results**

The results obtained when I first evaluated if hero\_id and player Ire a good metric to predict the winner of a match these Ire:

|  |  |
| --- | --- |
| Algorithm Used | Prediction Score |
| SVM Soft Margin | 52.11% |
| Random Forest Classifier | 51.26% |

**8.Conclusion**

I started this project considering what would be good features in order to predict the winner of a dota 2 match before the game start. And from the data that I got in the dataset I wouldn't be able to successfully predict the winner and get an high accuracy score, because several information Ire missing like, the winning rate of each hero in that specific patch, and what pairs of set of heroes are good against another set or pair of heroes. Having those making a prediction before the start would be much easier and the accuracy would have been above 70%.

Therefore since I did not had some of this information I had to change to predicting the winner of the game after the game ended given several features that Ire updated according to the game and having knowledge about the game I able to pick features that would give us good overall prediction score. Using that higher number of features I Ire able to increase 50% to 87% using random forest classifier.

**9. Future Work**

Based on the work done so far, there are two new approaches that could be included in future work.

1. Include the other 6 features that I consider to be important to predict the game winner which was not implemented in this interaction of the project because sometimes to run the algorithm it would take 6 hours or more so I had to decrease in order to finish the project on time. Including those features and checking how much they would influence the result would be important to further stretch this into an applicative for Dota 2 players.
2. Contact the company making the game and get information regarding winning rates of each hero and the collection of heroes that have good combined winning rates so that I can make an before the game starts prediction, and perhaps develop this into an app.

**10. References**

[1] Setting up a prediction problem dota <https://www.kaggle.com/devinanzelmo/setting-up-a-prediction-problem-dota-2>

[2] hero picking and match result

<https://www.kaggle.com/davidmercury/hero-picking-and-match-result>

[3] Dota 2 Matches <https://www.kaggle.com/devinanzelmo/dota-2-matches/>