Win or Lose: Maximizing Success in Dota 2 through Prediction of Professional Matches and Analysis of Machine Learning Models

Stephen Chan[1] and Sean Uy[2]

[1]MS in Data Science, Ateneo de Manila University

[2]MS in Computer Science, Ateneo de Manila University

May 2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

Introduction

1.1 Brief introduction into Dota 2

Dota 2 is a wildly popular multiplayer online battle arena (MOBA) video game developed by Valve Corporation. The game is a team-based 5v5 isometric third-person MOBA which has its roots in a Warcraft 3 custom map called ”Defense of the Ancients: Allstars”.

The objective of the game is to destroy the opposing team’s megastructure called ’ancient’, although there are smaller structures like towers or barracks that players can destroy to gain an advantage. Players use hero avatars, and gain experience and gold through various activities such as killing enemy foot soldiers called ’creeps’, killing enemy heroes, assists, and the like. Heroes are also able to spend gold to acquire items.

There are currently 115 heroes available, and more than 166 items in-game. Combining with the selection of 10 different heroes per game, this presents a level of complexity in the permutations and combinations of various heroes and items.

Rooted in the professional scene of Dota 2, players began to specialize in their roles in-game. The rank or position dictate gold or farm priority. Table 1.1 shows the roles and order of the specialization, with the 1st having the most gold and and experience.

Position	Role
1	Hard carry
2	Mid / Playmaker
3	Offlane / Utility
4	Playmaking Support
5	Hard Support

Table 1.1: Specialization roles of players in a team

Dota 2 is also regularly given patches or game updates. In a year, the game can receive as much as three to four major patches, and hundreds of smaller patch changes. This, in turn, drives the meta, or 'most effective tactics available'. In this context, meta refers to the overarching strategies or approaches that is generally seen as the most effective. For instance, some patches favor a 'push' meta, where the emphasis is grouping together and taking down towers and objectives to win quickly. In a push meta, players would pick heroes, items, and tactics that revolve around this general strategy.

Professional Dota 2 matches are done in a game mode called Captain's Mode, where each team bans six heroes and picks five heroes in a specific order from the 115 hero pool. Figure 2.1 shows the order by which each team ban and pick a hero. This collection is known as the 'draft picks'.

1.2 Objectives

Because Dota 2 is an incredibly complex game, the group chose to focus on two specific objectives:

1. Create a model that can best predict a win or loss given certain features or variables.
2. Determine whether draft picks or the team's overall performance within the first ten minutes is a better predictor of a match outcome.

1.3 Scope and Limitations

In light of the discussion on Dota 2's ever evolving game state, the group chose to focus on professional matches. Professional matches are defined as matches in a tournament setting where there is a cash prize pool. The assumption is that there is a greater incentive for teams to perform better in this context. The aim is to discourage 'throwing' - a phenomenon whereby a player or team intentionally gives up and performs actions that are disadvantageous for his team and advantageous for the opponent's. Interestingly, this happens in public matches even when the players are of a high rank. However, throwing almost never occurs in a professional setting. Thus, by limiting the scope to professional matches, we can preserve a higher-than-average gameplay level, and minimize outliers where players or teams play in bad faith.

The group also chose to limit the scope of data to professional matches played within the latest patch (7.21). This aims to preserve the bias of meta shifts. As discussed above, meta can change, often significantly, from patch to patch. Since patch releases are so frequent (under 7.21 alone, there have been four sub patch releases and at least ten smaller changes made), by considering this general patch release, the group can have a more or less stable meta, and still have enough data size for meaningful analysis.

Lastly, the group chose to consider team performance only for the first ten minutes. The group felt that while it is an arbitrary number, ten minutes signify the end of the 'early phase' or 'laning phase'. During this phase, the discrepancy between teams in terms of gold or experience isn't very high (relative to the potential outcome of the match). Hence, the individual and nuanced team performance's significance is greatly highlighted. There's also the concept of 'snowballing', where a team who has managed to gain an advantage at the early stages of the game tend to accrue bigger advantages as the game progress, ultimately beating their opponent. By selecting a smaller subset of early game metrics, there's less numeric data highlighting snowballing. Else, when looking at team performance at the end of the game, it isn't surprising why one team won over the other.

There are a few things that the group purposely did not take into consideration, namely: item choices and item timings, and individual player performance history. The choice to exclude items in the analysis is largely technical - the sheer number of items, combinations, and time elements make the data irregular and difficult to compare to one another. Since the group chose to limit to the team's performance within 10 minutes, at this point, item disparity isn't significant yet.

The group also did not consider the player behind the heroes. In high-level Dota 2 analysis, analysts often compare a player's performance on a hero with another player, and cite the different nuances either player exhibits. Certain specific players also tend to favor a certain pool of heroes, and this can affect the matchup.

The group chose to exclude those factors because the attempt was to create a generalized approach and model for prediction. There's also the issue where while there is much history and information for top-tier teams, low-tier teams, who are significantly less famous, have little to no historical data to analyse. Hence, these factors were excluded from analysis.

CHAPTER II

Methodology

2.1 Data Gathering

Data was obtained using an API from OpenDota, a volunteer-driven open source platform that provides Dota 2 data. As mentioned above, the group chose to select the draft picks and team performance for the first ten minutes, from professional matches within the 7.21 patch, between January 29 and May 8, 2019. Ultimately, the dataset will be a collection of professional matches. However, the group also chose to separate the matches between low-tier teams and top-tier teams.

2.1.1 Low-tier vs Top-tier Teams

The group chose to differentiate low-tier teams and top-tier teams because the group wanted to see if there is some difference in the way the two groups perform, in both draft picks and overall team performance.

Top-tier teams are defined to be professional teams that have accrued Dota Pro Circuit (DPC) points. The DPC points are important if one wants to compete at the biggest yearly tournament - The International. Hence, an assumption the group has made is that teams who have accrued, or are trying to accrue DPC points are those teams who are more serious and thus, have invested significantly into their Dota 2 careers.

Since DPC points are awarded to teams who have placed in big Valve-sponsored tournaments, like the recently concluded MDL Disneyland or Chongqing Major, it can be said that those who have garnered the points also tend to perform better than the rest, in general. In total, there were 33 teams found to have DPC points, and these 33 teams will be the basis for a top-tier team (some teams were found to have been disband at the time of writing)[1]. Low-tier teams are any team who have participated in a tournament with a cash prize pool but aren't part of those 33. The group considers a match in the top-tier category if at least one of the 33 teams are included. Table 2.1 features the entire list of top-tier teams.

Rank	Team	Rank	Team
1	Team Secret	18	Forward Gaming
2	Virtus.pro	19	paiN Gaming
3	Evil Geniuses	20	Natus Vincere
4	Vici Gaming	21	compLexity Gaming
5	PSG.LGD	22	Thunder Predator
6	Fnatic	23	BOOM ID
7	Team Liquid	24	Team Empire
8	Ninjas in Pyjamas	25	beastcoast
9	KEEN GAMING	26	Infamous Gaming
10	OG	27	Team Aster
11	TNC Predator	28	Royal Never Give Up
12	EHOME	29	The Pango
13	J.Storm	30	Old but Gold
14	Chaos Sports Club	31	Flying Penguins
15	Alliance	32	Majestic esports
16	Mineski	33	The Final Tribe
17	Gambit Esports		

Table 2.1: Top 33 Dota 2 teams

Apart from splitting the data between top-tier teams and low-tier teams, the group also split the data between draft picks and team performance. This way, separate analysis can be done on each group of data that can aid in answering the second objective.

Low-tier Team	Top-tier Team
Draft Table Low	Draft Table Top
Team Table Low	Team Table Top

Table 2.2: Data split to four sets

2.2 Feature Engineering

Because the depth of data is quite verbose, the group had to select specific features to be used for both tables. Otherwise, the resulting data table would be uneven, whereby certain rows would be of different length than the others.

2.2.1 Draft Table

Creating the draft table is direct. The entire bans and picks were selected in chronological order, without differentiating between a pick or a ban, or for which team, whereby the first feature is a ban for the first team, the second feature is a ban for the second team, and so on. Figure 2.1 is a representation of the 22 features for the draft table[2]. Next two features are the team ID numbers. The last column of the draft table is the target, where 1 represents a radiant win and 0 represents a dire win. The values for the features are the hero ID numbers.

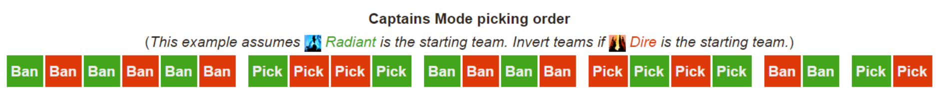


Figure 2.1: Captain's Mode draft order

2.2.2 Team Table

Selecting features for the team performance table requires a bit more intuition and domain knowledge, as well as a review of the features common to all matches. Table 2.4 is a table representing all features and a brief explanation for each. Intuitively, the most salient features are the gold and experience discrepancy called 'advantage'. The value is positive if the discrepancy favors the radiant side, and negative if it favors the dire side. Tower kills are also included. Sentry and observer wards are in-game items that provide vision. The group included them because in the game, vision can be a powerful mechanism to gain an advantage over one's opponents, hence ideally the more vision-granting items used, the more likely to gain an advantage.

The next four sets of features refer to average and standard deviation of certain in-game metrics of each team by player. These features indicate an aggregation of individual performance, as well as an indication on how well the team members have performed their roles outlined above. That is, a highly effective and specialized team would have a higher standard deviation of their net worth, net experience, last hits, and so on.

Lane efficiency is a game-generated metric that indicates the ratio of gold each player has over the total amount of gold one can have (assuming they have successfully last hit all the lane creeps) at a given time. High lane efficiency indicates a player who make efficient movements within the map. Lastly, overall kills are included. The last column is the target or the y to be predicted.

2.3 Preprocessing

Currently, there are four datasets, namely: draft_tables_low, draft_tables_top, team_tables_low, and team_tables_top, which is represented in Table 2.2. However, to allow for further analysis, the datasets must undergo preprocessing. Incidentally, the elements of the draft tables are all categorical whereas those of team tables are numeric.

Since the categorical values of the data tables refer to hero IDs, a multi-level binarizer is used. This is similar to one-hot encoding except instead of one encoding per category, this results to a 115-long list of zeroes except for heroes who were picked or banned having values of 1.

For the team tables, since the values are numeric, we have chosen to scale the values using a standard scaler with the formula:

$$z = \frac{x - \mu}{\sigma}$$

As a result, we end up with eight datasets as shown in Table 2.3.

Top-tier	Alias	Low-tier	Alias
Draft table top	DTT	Draft table low	DTL
Team table top	TTT	Team table low	TTL
Draft table one hot top	DOT	Draft table one hot low	DOL
Team table scaled top	TST	Team table scaled low	TSL

Table 2.3: Summary of tables after preprocessing

Variable name	Description
gold_adv	Aggregate gold advantage. Positive if advantage is for radiant, negative if for dire.
xp_adv	Experience advantage. Positive if advantage is for radiant, negative if for dire.
rad_tower_kill	Number of towers destroyed by the radiant team
dire_tower_kill	Number of towers destroyed by the dire team
rad_obs	Number of observers used by the radiant team. (observers provide vision)
dire_obs	Number of observers used by the dire team
rad_sens	Number of sentries used by the radiant team (sentries provide deeper vision)
dire_sens	Number of sentries used by the dire team.
rad_xp10	Average experience for all the members of the radiant team
dire_xp10	Average experience for all the members of the dire team
rad_xpstd	Standard deviation of rad_xp10. Higher is better
dire_xpstd	Standard deviation of dire_xp10. Higher is better
rad_g10	Average gold for all the members of the radiant team
dire_g10	Average gold for all the members of the dire team
rad_gstd	Standard deviation of rad_g10. Higher is better
dire_gstd	Standard deviation of dire_g10. Higher is better
rad_lh10	Average number of last hits for all the members of the radiant team
dire_lh10	Average number of last hits for all the members of the dire team
rad_lhstd	Standard deviation of rad_lh10. Higher is better
dire_lhstd	Standard deviation of dire_lh10. Higher is better
rad_le10	Average lane efficiency of the members of the radiant team
dire_le10	Average lane efficiency of the members of the dire team
rad_lestd	Standard deviation of rad_le10. Higher is better
dire_lestd	Standard deviation of dire_le10. Higher is better
rad_kills	Number of kills by the radiant team
dire_kills	Number of kills by the dire team
radiant_win	1 if radiant team wins. 0 if dire team wins.

Table 2.4: Table of selected features for team performance

2.4 Machine Learning Algorithms

2.4.1 Logistic Regression

Logistic regression is a popular regression model that can classify features into binary targets. It uses the logit function to approximate the probability that the y or target belongs to one particular category[3]. Logistic regression is also robust in that it can handle categorical and numeric feature values, and thus, is applied to both draft and team tables.

2.4.2 Naive Bayes Classifier

Naive Bayes is a generative probabilistic model based on Bayes' theorem that acts as a classifier with the assumption that each feature is independent. In reality, however, each particular pick or ban prompts a response from the opponent since the picks and bans are alternating. For example, if hero X is banned by team A, there is a likelihood team B will respond by banning hero Y, and so on. In this regard, naive Bayes may have some limitations in its predictive power. This model will be used for both draft and team tables also due to handle both categorical and numerical features.

2.4.3 Random Forest Classifier

Random Forest Classifier is an ensemble model made up of a number of decision trees, where the aggregate produces an output that tends to be more accurate than simply using a single decision tree. Random forest classifiers subset not just a bootstrap resample of the instances (ie. resampling with replacement), but also a sample of the features in each decision tree made[3]. The final predictions are made by averaging the results of each individual tree[4]. This model will also be applied to both draft and team tables.

2.4.4 k-Nearest Neighbors

k-Nearest Neighbors (k-NN) is a simple classification algorithm that considers the class of a new instance depending on the class of its k nearest neighbors, when the features are plot out in a dimension space. This method works only for numeric features and thus will not be used for the drafts table. Since there are team tables with standardized feature values, it is expected to perform better than on the non-standardized tables.

2.5 Model Interpretability

Often there exists some variables in the regression models that are deemed to be of less importance to the prediction of the target. The inclusion of these irrelevant variables often makes the model complex while not contributing much to prediction. By identifying these variables, the model can be more easily interpretable by highlighting specific variables in trying understanding how the model predicts the target[3].

2.5.1 Feature Importance through t-test

For feature importance, the idea of significance testing is to check the likelihood that the feature values for each class has significant differences for differentiating the classes. At values near 0, the idea that the feature does not contribute to identifying the difference of the two classes must be true while on the other hand a large value may indicate that scores can be different significantly. Sorting all features based on the absolute value of their t-test will indicate feature importance[3].

2.5.2 Step-wise Selection

Computationally, step-wise selection is a better approach than other methods under the category of subset selection. It does not suffer over-fitting even with a large feature set while the other methods can over-fit the model[3].

Forward Step-wise Selection

Forward step-wise selection starts with a model with no features, then slowly adds one feature at a time until all are present in the model. Before increasing the number of features chosen in the current step for the model, all different features are tested and only the best features with the lowest residual sum of squares (RSS) is chosen. RSS is computed by taking the sum squared difference of the predicted and the actual[3].

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

The best model for the current number of features is added to a pool of best models, then the process is repeated until it reaches the maximum number of features. The best model in the pool is then selected.

Backward Step-wise Selection

Backward step-wise selection starts with the model having all features and iteratively removing the least useful among the group. Similar to forward selection, backward selection uses RSS to choose the best model, and stores it in a pool of best model, with the last step finally choosing the best model[3].

Mixed Step-wise Selection

A hybrid approach can be done on both backward and forward selection by combining the two selection methods. Mixed step-wise selection is the result of incorporating the two methods while preserving their advantages. The selection starts by choosing the best feature to be added like forward selection, however, when the method adds newer and newer features, the method also can remove features that do not contribute to the model similarly in backward selection[3].

CHAPTER III

Results

The results of the models on different data sets are shown in 3.1. Interestingly, the draft tables for any model have shown poor accuracy. In fact, doing one hot encoding on the draft tables yielded worse results in general. This may be attributed to the fact that team IDs were omitted in the new draft tables. Model interpretability can explain the cause of this discrepancy.

Similarly, scaling the team tables also yielded very little improvements, the most significant is a 1% increase for team tables for low-tier teams.

What's surprising is that there is a clear difference between top-tier teams and low-tier teams across the board on all models and all team tables. A likely interpretation is that low-tier teams are much more predictable in their outcome compared to top-tier teams, who have the better ability to turn around the general trend of the match (ie. if a match is going poorly for them with respect to kills, gold, and experience, a top-tier team may better possess the skills to counter that trend and still win in the end).

With respect to both objectives, a simple k-NN model generally yielded the best results out of all the models for the team tables. It may be possible that at such a high calibre of games, early team performance is a better indicator of match outcome than the draft picks. However, a study by Andono et al. was able to use naive Bayes and Adaboost with Gaussian distribution kernel to achieve an 80% accuracy using the draft picks alone[5].

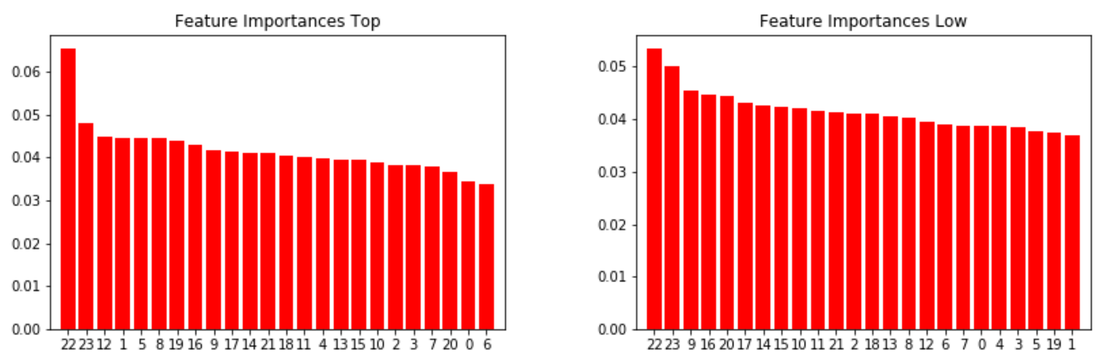
	Dataset							
Model	DTT	DTL	TTT	TTL	DOT	DOL	TST	TSL
Logistic Regression	57.65%	**57.32%**	66.66%	72.99%	55.55%	52.35%	66.15%	74.37%
Naïve Bayes	**57.84%**	55.45%	66.92%	73.82%	53.79%	44.90%	66.92%	73.68%
Random Forest	54.48	54.10%	66.92%	73.27%	**55.73%**	**54.03%**	66.92%	72.99%
k-Nearest Neighbors	-	-	**68.91%**	**75.21%**	-	-	**68.41%**	**76.18%**
k-value	-	-	21	85	-	-	37	97

Table 3.1: Accuracy scores of the different datasets and the different models used.

3.1 Model Interpretability Results

3.1.1 Random Forest Classifier

Random forest classifiers are useful for its ability to showcase feature importance quickly. As seen in Figure 3.1, both figures tend to highlight the 23rd and 24th features (x-axis lists 22 and 23 since it starts at 0), namely, the team IDs, more than the actual draft picks. However, the overall percentage or values are very low across the board, so it's very difficult to say with certainty that the certain features are indeed considered inherently important. Removing the team IDs as features shows no discernable feature as important. This coincides with the low accuracy scores generated on draft tables.



(a) Random Forest DTT

(b) Random Forest DTL

Figure 3.1: Feature importance of random forest classifier on draft tables

	TTT-B	TTT-F	TTT-M	TST-B	TST-F	TST-M	TTL-B	TTL-F	TTL-M	TSL-B	TSL-F	TSL-M
gold_adv	**0.56%**	**0.74%**	**0.56%**		48.83%			80.64%		12.20%	87.40%	12.20%
xp_adv		84.64%			26.78%			64.57%			48.46%	
rad_tower_kill	13.44%	18.87%	13.44%		17.17%		**0.08%**	**0.24%**	**0.08%**	**0.05%**	**0.23%**	**0.05%**
dire_tower_kill		33.00%			33.76%			76.93%			78.65%	
rad_obs		32.10%			29.87%			64.67%			66.52%	
dire_obs		87.24%			86.38%			39.61%			38.92%	
rad_sen	**2.42%**	**2.97%**	**2.42%**	**2.66%**	**3.11%**	**2.66%**		99.31%			98.48%	
dire_sen		83.40%			82.71%			99.03%			98.99%	
rad_xp10	**0.93%**	13.70%	**0.93%**	**0.42%**	26.78%	**0.42%**	**2.02%**	17.17%	**2.02%**	**0.92%**	48.46%	**0.92%**
dire_xp10					26.78%						48.46%	
rad_xpstd		70.91%			67.34%			36.91%			37.75%	
dire_xpstd		72.68%			71.10%			89.51%			89.09%	
rad_g10		16.64%			48.83%			29.20%			87.40%	
dire_g10				**0.23%**	48.83%	**0.23%**				**1.92%**	87.40%	**1.92%**
rad_gstd	5.69%	**2.06%**	5.69%	**0.97%**	**2.13%**	**0.97%**	**0.99%**	60.84%	**0.99%**	**0.50%**	59.71%	**0.50%**
dire_gstd		83.43%			79.66%		**2.47%**	**3.98%**	**2.47%**	**0.89%**	**3.83%**	**0.89%**
rad_lh10	8.14%	38.57%	8.14%	8.07%	39.91%	8.07%	**0.46%**	**1.87%**	**0.46%**	**0.62%**	**1.94%**	**0.62%**
dire_lh10	**2.24%**	**3.50%**	**2.24%**	**1.54%**	**3.29%**	**1.54%**	**0.00%**	**0.30%**	**0.00%**	**0.00%**	**0.29%**	**0.00%**
rad_lhstd		70.10%			69.67%			35.85%			37.32%	
dire_lhstd		62.76%			64.28%			57.90%			58.35%	
rad_le10	**0.00%**	**1.63%**	**0.00%**	**0.07%**	**1.64%**	**0.07%**	**0.00%**	**0.50%**	**0.00%**	**0.02%**	**0.48%**	**0.02%**
dire_le10	**0.00%**	**0.02%**	**0.00%**	**0.00%**	**0.02%**	**0.00%**	**1.73%**	25.39%	**1.73%**		25.85%	
rad_lestd	8.78%	5.54%	8.78%	**1.83%**	5.58%	**1.83%**		62.29%			63.41%	
dire_lestd		98.45%			95.81%		**2.70%**	**3.59%**	**2.70%**	**0.84%**	**3.48%**	**0.84%**
rad_kills		94.71%			93.78%			59.57%			60.93%	
dire_kills		34.02%			32.59%		**0.04%**	**0.64%**	**0.04%**	**0.01%**	**0.65%**	**0.01%**

Table 3.2: p-value of team tables with backward (B), forward (F), and mixed (M) selection

3.1.2 Step-wise Selection for Team Tables

The results of the step-wise selection for the different team tables data set, as seen in Table 3.2, shows the p-value using the different types of selection methods. Features with no values were not chosen by the selection methods as they do not contribute to target prediction. If the features shows a p-value less then or equal to 5%, then this shows that the feature shows significant evidence that it is relevant to the linear model. Interestingly both backwards and mixed selection shows the same results for all data sets.

Comparing both scaled versions and non-scaled versions of both top-tier and low-tier, more or less they give the same set features though there are some that are added and some removed. Based on the lowest p-value of the variables for top-tier teams, it can be seen that lane efficiency for both dire and radiant contributes more to the prediction of the target. While low-tier teams, depending on whether scaled or not, also has low p-values for lane efficiency and radiant experience, it can be seen that the standard deviation of gold, the number of last hits and radiant tower kills are present for this set of data.

	DTT-B	DTT-F	DTT-M	DTL-B	DTL-F	DTL-M
team0_ban1		99.29%			17.60%	
team1_ban1	**0.21%**	**0.19%**	**0.21%**		22.14%	
team0_ban2		79.15%			76.47%	
team1_ban2		57.20%			41.80%	
team0_ban3		54.87%			77.06%	
team1_ban3	10.70%	12.63%	10.70%		87.10%	
team0_pick1		54.88%			46.07%	
team1_pick1		83.21%			84.84%	
team1_pick2	11.27%	10.11%	11.27%		61.33%	
team0_pick2		39.37%			28.09%	
team0_ban4		84.28%			64.83%	
team1_ban4		53.24%			34.26%	
team0_ban5		24.12%			94.85%	
team1_ban5	**4.14%**	**3.43%**	**4.14%**	15.20%	13.23%	15.20%
team1_pick3		73.79%			57.14%	
team0_pick3		29.27%			17.36%	
team1_pick4		58.55%			73.06%	
team0_pick4		19.34%		8.49%	8.43%	8.49%
team1_ban6		90.76%		7.47%	7.18%	7.47%
team0_ban6	10.97%	7.40%	10.97%		90.51%	
team0_pick5		27.68%		**1.24%**	**1.33%**	**1.24%**
team1_pick5		21.13%			53.38%	
team_0_radiant	**0.01%**	**0.00%**	**0.01%**		34.57%	
team_1_dire	**0.00%**	**0.00%**	**0.00%**		22.40%	

Table 3.3: p-value of draft tables with backward (B), forward (F), and mixed (M) selection

3.1.3 Step-wise Selection for Draft Tables

While both the top-tier team tables and low-tier team tables do not immediately show clear distinctions in feature selection, for draft table step-wise selection, as shown in Table 3.3, it is much clearer. For top-tier teams, the feature that has the possibility to contribute to the prediction of a win or a loss are the features that contain the team IDs. This is similar to the findings of feature importance of random forest classifiers. Noticeably in top-tier games, the matching of the teams can determine whether or not a win can happen for some teams. Some teams do not perform well against their rivals as they often mention that their strategies have already been well studied by their opponents when interviewed about the match-up[6].

For low-tier teams, the most significant feature that can help contribute to a prediction of a win or a lose is the final hero pick of team Radiant. This makes sense since this locks the Radiant team to their final strategic lineup and gives a final chance for Dire to counter the hero and strategy that Radiant chooses to employ. For low-tier teams, not being able to counteract the strategies chosen by their opponents can easily determine a win or a lose unlike their top-tier teams where they can more or less play different strategies and use more of their skills to win rather than hero lineups.

TTT-Feature	TTT	TTL-Features	TTL	TST-Features	TST	TSL-Features	TSL
dire_lestd	0.02	rad_sen	0.01	dire_lestd	0.05	dire_sen	0.01
rad_kills	0.07	dire_sen	0.01	rad_kills	0.08	rad_sen	0.02
dire_obs	0.16	dire_xpstd	0.13	dire_obs	0.17	dire_xpstd	0.14
xp_adv	0.19	gold_adv	0.25	dire_sen	0.22	rad_g10	0.16
dire_gstd	0.21	dire_tower_kill	0.29	dire_gstd	0.26	gold_adv	0.16
dire_sen	0.21	rad_obs	0.46	dire_xpstd	0.37	dire_g10	0.16
dire_xpstd	0.35	xp_adv	0.46	rad_lhstd	0.39	dire_tower_kill	0.27
rad_xpstd	0.37	rad_lestd	0.49	rad_xpstd	0.42	rad_obs	0.43
rad_lhstd	0.38	rad_gstd	0.51	dire_lhstd	0.46	rad_lestd	0.48
dire_lhstd	0.49	rad_kills	0.53	dire_g10	0.69	rad_kills	0.51
rad_lh10	0.87	dire_lhstd	0.55	rad_g10	0.69	rad_gstd	0.53
dire_kills	0.95	dire_obs	0.85	gold_adv	0.69	dire_lhstd	0.55
dire_tower_kill	0.97	rad_xpstd	0.90	rad_lh10	0.84	xp_adv	0.70
rad_obs	0.99	rad_lhstd	0.92	dire_tower_kill	0.96	dire_xp10	0.70
rad_tower_kill	1.31	rad_g10	1.05	dire_kills	0.98	rad_xp10	0.70
rad_g10	1.38	dire_le10	1.14	rad_obs	1.04	dire_obs	0.86
rad_xp10	1.49	rad_xp10	1.37	xp_adv	1.11	rad_xpstd	0.88
rad_lestd	1.92	dire_gstd	2.06	dire_xp10	1.11	rad_lhstd	0.89
dire_lh10	2.11	dire_lestd	2.10	rad_xp10	1.11	dire_le10	1.13
rad_sen	2.18	rad_lh10	2.35	rad_tower_kill	1.37	dire_gstd	2.07
rad_gstd	2.32	dire_kills	2.73	rad_lestd	1.91	dire_lestd	2.11
rad_le10	2.41	rad_le10	2.81	dire_lh10	2.14	rad_lh10	2.34
gold_adv	2.68	dire_lh10	2.98	rad_sen	2.16	dire_kills	2.72
dire_le10	3.75	rad_tower_kill	3.04	rad_gstd	2.31	rad_le10	2.82
				rad_le10	2.40	dire_lh10	2.98
				dire_le10	3.78	rad_tower_kill	3.05

Table 3.4: t-statistic of features from team tables

DTT-Features	DTT	DTL-Features	DTL
team0_ban1	0.01	team0_ban5	0.06
team1_ban6	0.12	team0_ban6	0.12
team0_ban4	0.20	team1_ban3	0.16
team1_pick1	0.21	team1_pick1	0.19
team0_ban2	0.26	team0_ban3	0.29
team1_pick3	0.33	team0_ban2	0.30
team1_pick4	0.55	team1_pick4	0.34
team1_ban2	0.57	team0_ban4	0.46
team0_pick1	0.60	team1_pick2	0.51
team0_ban3	0.60	team1_pick3	0.57
team1_ban4	0.62	team1_pick5	0.62
team0_pick2	0.85	team0_pick1	0.74
team0_pick3	1.05	team1_ban2	0.81
team0_pick5	1.09	team_0_radiant	0.94
team0_ban5	1.17	team1_ban4	0.95
team1_pick5	1.25	team0_pick2	1.08
team0_pick4	1.30	team_1_dire	1.22
team1_ban3	1.53	team1_ban1	1.22
team1_pick2	1.64	team0_ban1	1.35
team0_ban6	1.79	team0_pick3	1.36
team1_ban5	2.12	team1_ban5	1.51
team1_ban1	3.12	team0_pick4	1.73
team_0_radiant	4.11	team1_ban6	1.80
team_1_dire	4.56	team0_pick5	2.48

Table 3.5: t-statistics of features from the draft tables

3.1.4 Feature Importance through t-test

The ranked t-test of all data sets, shown in Table 3.4 and Table 3.5, shows how well the values of the feature can differentiate the prediction of a win or a lose. Interestingly, the higher values shown in Table 3.4 and Table 3.5 are those that have low p-values in the step-wise selection methods.

Both scaled and non-scaled team tables for top-tier teams show lane efficiency and gold as the features that can more clearly differentiate a win or a loss. For the draft data sets, the more salient features turn out to be the team IDs, meaning which team is playing against which.

Low-tier teams also follow the same observation with feature importance and selection. Last hits and tower kills, and last pick and last ban for the team and draft data sets respectively better help determine a win or a loss.

CHAPTER IV

Conclusion

While the accuracy scores in general were considered to be poor across the board, the difference between the accuracies of the top-tier and low-tier teams show that predicting outcomes for top-tier teams is much more difficult. The group theorizes that the top-tier teams have better ability and skill to stop the aforementioned 'snowball' effect, and quickly negate any disadvantage they may have at that time.

Additionally, the features that better contribute to a win or lose tend to be different for top-tier and low-tier teams. For top-tier teams, match-ups (ie. which team faces which team) are deemed more important than the actual draft picks. Perhaps due to a smaller pool, top-tier teams, who regularly scrim against each other, already have an innate and intimate knowledge of their opponents' capabilities.

For low-tier teams, it is the last pick and last ban, as these tend to dictate the approach of either team going into the game. Perhaps this is due to the lack of ability for low-tier teams to adapt quickly to certain matchups, hence the emphasis on the last pick and ban.

Top-tier teams also favor lane efficiency, according to the results generated, as the feature that better contributes to the outcome of the match. This makes sense since top-tier teams tend to be equally mechanically able, and the fight comes down to nuance.

In contrast, low-tier teams favor map control and gold as explained by the results where low-tier matches associate tower kills as a contributing feature to the outcome of a match. Map control is certainly an integral element for a successful match, and perhaps it is down to the execution of basic map control that can lead to a win or a loss.

Dota 2 inherently is a very complex game. The amount of permutations and combinations of heroes, items, item timings, skill selection and usage, among others, make it difficult to create a predictive model. In fact, from the results, we saw that draft picks generally weren't very useful in predicting outcomes, as compared to the performance of teams within the first ten minutes. Simply looking at accuracies generated by traditional machine learning algorithms did not yield much information in the way of prediction and interpretation.

However, by using statistical inference in interpreting the models, we are able to understand which features have a bigger impact to game outcomes. Interpreting the importance and impacts of these features leads to better appreciation and understanding of game itself, the mindset of professional teams, and how to play in order to maximize the chances of winning the game.

BIBLIOGRAPHY

[1] "Dota pro circuit 2018-19 rankings." [Online]. Available: https://liquipedia.net/dota2/Dota$_{Pro_Circuit/2018-19/Rankings/Full}$

[2] Dota, "Game modes," Mar 2019. [Online]. Available: https://dota2.gamepedia.com/Game$_{m}odesCaptains_{M}ode$

[3] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, ser. Springer Texts in Statistics. Springer New York, 2013. [Online]. Available: http://doi.org/10.1007/978-1-4614-7138-7

[4] W. Koehrsen, "An implementation and explanation of the random forest in python," Aug 2018. [Online]. Available: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76

[5] P. N. Andono, N. B. Kurniawan, and C. Supriyanto, "Dota 2 bots win prediction using naive bayes based on adaboost algorithm," in *Proceedings of the 3rd International Conference on Communication and Information Processing*, ser. ICCIP '17. New York, NY, USA: ACM, 2017, pp. 180–184. [Online]. Available: http://doi.acm.org/10.1145/3162957.3162981

[6] "Kuroky about team liquid: "we have two choices: we work hard or we make changes"." [Online]. Available: https://www.vpesports.com/dota2/news/kuroky-about-team-liquid-we-have-two-choices-we-work-hard-or-we-make-changes