

# To win or not to win? A prediction model to determine the outcome of a DotA2 match

Kaushik Kalyanaraman\* Department of  
Computer Science  
University of California, San Diego  
kkalyana@eng.ucsd.edu

**Abstract**—In this paper, we present an augmentation to an existing machine learning algorithm used to predict the outcome of a DotA2 match and as a hero recommender in a recommendation engine. We briefly discuss existing work on DotA2 recommendation engines as well another effort in applying traditional machine learning algorithms to predict its outcome.

We then detail the augmented algorithm used to improve the prediction results of the existing model and detail the entire process involved i.e., data collection, feature extraction and feature encoding. We then expound upon various aspects and possible improvements to the algorithm and different directions for future work.

## I. INTRODUCTION

**DotA2** or Defense of the Ancients 2 is a 5v5 online, multiplayer, arena-based game which originally started off as a custom mod for the Warcraft III Frozen Throne platform by Blizzard [1]. It originated in 2003 and has had a steadily growing player base since then. The objective of this scenario is to destroy the opponents' primary structure known as the Ancient, which itself is heavily guarded. The two warring sides were originally known as "The Sentinel" and "The Scourge" but were renamed to "The Radiant" and "The Dire" when the DotA 2 project was taken over by Valve Corporation. It is an extraordinarily popular game in the electronic sports domain as is evident by the prize pool of the most recent Valve DotA 2 tournament, The International 4, which had a prize pool exceeding \$10 million [3].

DotA 2 matches support many game modes such as "All Pick", "Random Draft", "Single Draft", "Captains Mode" etc. Each player selects his/her own hero to control before the beginning of the match, from a pool of 109 heroes and that is the only hero that player will be able to control for the rest of the game. Players level up their heroes by earning experience points, obtained by killing enemy heroes, killing "enemy creeps" or "neutral creeps". They also obtain gold through the same methods as obtaining experience points and can gain special abilities called "spells" unique to each hero. A hero is essentially the sum of its spells in that they are critical to its success and heroes with spells that work in tandem with one another are said to be "synergistic" or "complementary".

Players carefully consider the hero to play with, based not only on the heroes' strengths and weaknesses but also in their synergy with allied heroes and how well they perform against enemy heroes. Synergy with allies and team work is

quintessential to this game and is in fact, the very heart and core of DotA 2. Without well coordinated team work and a team composition with heroes which complement each other, it is nearly impossible to win. Considering this, a well chosen team of heroes can often even make up for sloppy gameplay and lower skill of the players themselves, because such a team offers a significant advantage over the opponents even before the start of the game.

We see that this fits a classic predictive modelling task and we seek to exploit the synergistic relationships between heroes to augment existing prediction algorithms already used to hero recommendation systems such as Dota2cp [2]. The number of possible unique hero combinations is huge, about  $1.074 \times 10^{16}$ . Such a large state space almost always ensures that each game is unique and merits significance. The task of predicting the outcome of a match based solely on hero combinations is a challenging one, because it tries to model through statistical means, what experienced players have gained through thousands of hours of gameplay.

## II. RELATED WORK

Dota2cp [2] is a recommendation engine developed for hero recommendations for DotA 2 matches. An accuracy of 63% is reported by the author of Dota2cp for the winning team prediction. Hero selection is modeled as a zero-sum-game and the matrix is learned using logistic regression. Players are regarded as min-max agents that take turns picking a hero, one at a time.

An alternate piece of literature also exists on DotA 2, in which, the authors compare logistic regression vs. a customized K-nearest neighbors approach in building a recommendation engine [4]. They report a test accuracy of 67.43% during k-fold cross-validation for a K-nearest neighbors model and an accuracy of 69.8% with a simple logistic regression model. This shows that the hero composition of a team contributes to a significant extent, towards the probability of victory but it fails to explain and/or take into consideration synergistic relationships between allied heroes as well as antagonistic relationships between ally and enemy heroes.

## III. BACKGROUND

We believe that a succinct description of the deeper "mechanics" of the game is essential in understanding the rest of this paper. Each hero in the game has one of three primary

attributes, “Strength”, “Agility” and “Intelligence” and these primary attributes are what defines the essential role of a hero in a game. As a hero levels up, so does their attributes and their primary attribute levels up by a larger amount compared to their secondary attributes. Heroes can also be “ranged” i.e., able to attack from a distance or “melee” i.e., able to attack only when nearby. DotA 2 heroes are broadly classified into the following four categories, based upon their larger role and impact in the game:

- **Carry** - These heroes are the core of almost every team composition. Games are almost always lost if the “carry” has not had enough levels and items. Such a role often merits the requirement of the highest skilled player in the team to don it and should not be underestimated. A carry’s primary task is to obtain as much gold and experience, as fast as possible and obtain high level items using that gold. A carry’s ability to kill opponent heroes usually exponentially increases as a function of their items and this is reflected in what is known as the “snowball” effect in the DotA 2 community i.e., a well played carry hero often leads his / her team to victory almost single-handedly. Carry heroes come in many forms, but a majority of them are in the Agility class of heroes with a small number in the Intelligence and Strength classes.
- **Support** - An often under-appreciated and unrecognized role, the essential “support”, as its name suggests, offers aid to the primary carry heroes, defends the carry heroes against enemy threats, buys items such as couriers and wards, which aid the carry in their hero progression. A support hero may sacrifice himself / herself to save the carry and exists solely to ensure that the carry does what it is intended to do. Usually, these heroes are much stronger than carry heroes early in the game, during which time, carry heroes are quite weak and susceptible to deaths.
- **Ganker** - This is a role solely designed to surprise and eliminate enemy heroes i.e., picking them off at opportune times. These heroes often have abilities which aid them in this task, such as invisibility and being able to quickly deal large amounts of damage. A ganker’s job is to ensure that enemy heroes are as uncomfortable as possible anywhere in the map and also to give their own team’s carry, as much space as possible, to earn gold and experience. Heroes like Pudge, Bounty Hunter and Skywrath Mage are very common ganker type heroes.
- **Initiator** - An initiator is one who is critical in team fights i.e., fights where at least 3 out of 5 heroes in each team engage in each other in a climactic showdown. Initiators usually subsist under the above classes. Their primary role is to spark off team fights, by enabling their own team to start off on an advantageous footing. Such heroes usually have large area of effect skills that disable and / or “stun” enemy heroes into being inactive for a duration of time. “Enigma” and “Tidehunter” are two of the classic examples of initiators since both of

their final abilities “Black Hole” and “Ravage” have a big area of effect (aoe) and either disable and / or stun several enemy heroes for several seconds.

This is not to say that this is the only rigid classification of heroes in the game. It is extremely flexible in that heroes can sometimes switch roles such as the “support” becoming a carry and vice versa somewhere in the middle of a game, the existence of hybrid roles such as the “support-initiator” and much more. An exhaustive discussion of these details is not merited here.

#### IV. EXPLORATORY ANALYSIS

Since we consider only the pre-game state for our predictive task i.e., the game state upto and including the hero picks, our analysis is only upon the different hero compositions and not on any other aspect of the game or the players of the game. It is true that a game as complex and unpredictable as DotA 2 is influenced by hundreds of factors other than the team composition, such as, player skill, gold and exp growth (displayed as gold and exp graphs), the ability of the team members to coordinate among one another, any sudden and critical teamfights that could change the game etc. There are too many factors to enumerate here and we concentrate solely upon the different hero picks players can partake in, from the hero pool.

##### A. Effect of individual heroes on win rate

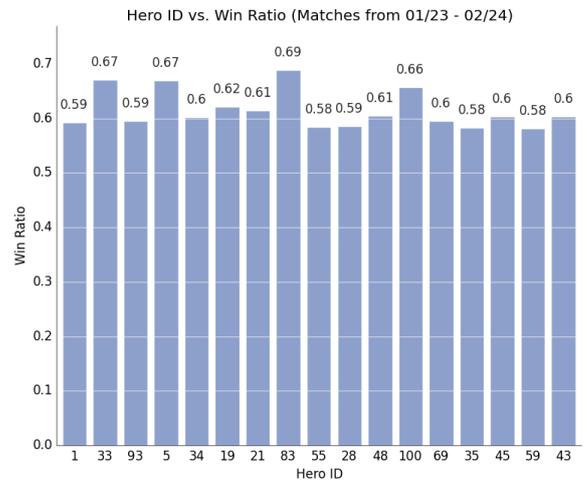


Fig. 1. A plot showing the influence of single heroes on the win ratio of DotA 2 matches. Refer Appendix for hero names

From the bar plot (thresholded above a ratio of 0.58), we clearly notice a correlation between a hero and its corresponding win rate. Hero IDs 83 with a ratio of 0.69 and 5 and 33 with a ratio of 0.67 correspond to the DotA 2 heroes, Drow Ranger and Sniper respectively (Appendix A has a map of hero IDs to hero names). Drow Ranger and Sniper are carry heroes (refer Section III) and are nearly impossible to overcome once they have their necessary items and hero level. In the current state of the game (a.k.a. game

meta), these three heroes are very strong in their own right and also synergistic with almost every possible hero lineup in the game. These two possible reasons could explain their high win to games ratio. However, a single hero does not win a team game but only contribute to a higher probability of victory. Hence, we continue our exploratory analysis to uncover, if any, possible correlations between pairs of heroes and their success rate.

### B. Effect of pairs of heroes on win rate

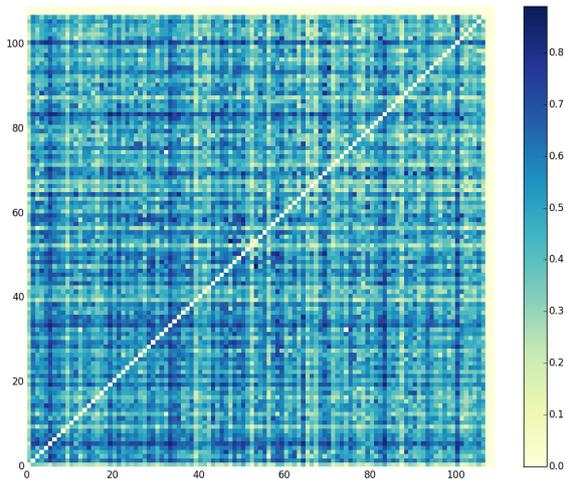


Fig. 2. A heat map showing the win rate of pairs of heroes.

Interestingly, the heat map shows a different picture altogether. It is constructed out of data from 27,362 DotA 2 matches and it shows the different hero compositions which when chosen in pairs, have a much higher win rate than just individual hero selection themselves. For instance, heroes 47 and 53 have the highest ratio of 0.89, which is almost a 90% win rate, which is extremely high in DotA 2 and these figures are statistically significant i.e. this hero pair has been picked in about 1200 games out of the total 27,362 games. Heroes 47 and 53 alone had a much lower win rate (less than 0.58) when considered separately because of their strength in complementing each other’s unique abilities when played together. We try to incorporate this information into the existing predictive model, which we felt was crucial to the essence of choosing heroes and optimal team compositions in this game.

## V. DATASET

We use Valve’s Steam API, documented in the DotA 2 dev forums [5] to pull information for 30,426 DotA 2 games from 01/23/2015 to 02/20/2015. The Steam API returns JSON data for each match and we then store the results in a MongoDB database. The constraints applied on the data we use are as follows:

- The game mode is one of “All Pick”, “Random Draft”, “Single Draft”, “All Random”, “Least Played” and

“Captains Draft”. The recommendation engine developed by Kevin et. al. [4] uses a similar list of modes to constrain their matches and the reasoning they state is that these modes are very representative of a “true DotA 2 game” as every hero has a chance of appearing on either team.

- Games have been filtered to be of a very high skill level i.e., all players in the game satisfy a minimum MMR (a match making rating assigned by Valve to players, similar to Starcraft II’s ELO Rating). We believe that these matches are representative of heroes being played to their full potential.
- They all have 10 players in the game, playing up until the end. We believe that games which have less than 10 players do not truly encompass the idea of a 5v5 multiplayer game and those ones which have disconnected human players a.k.a. players who have quit much before either team decisively won, are not indicative of what actually happened in a match.
- All games are of duration at least 900 seconds. We believe that games with a higher duration than 900 truly reveal how well heroes work in tandem with each other. This is not to say that really fast games are not indicative of it but we noticed that most games with a duration of less than 900 seconds are won and/or lost because of a tremendous difference in aggregate skill level between the opposing teams.
- Valve released a hero called “Winter Wyvern” on 02/13/2015, bringing the total tally of heroes up to 110. We did not have enough time to reset our data and gather enough matches and hence, all matches with this new hero are discarded so that it does not skew existing results.

This JSON data consists of player information such as player ID and player rating, the heroes they played, the sequence of skills learned by their respective heroes, the level up time and much more. About 90% or 17,000 matches is exported as training set and the rest 10% or 1,500 matches, as the test set.

## VI. METHODOLOGY

### A. Feature Vector for Regression

We use the same feature vector as described in the paper by Kevin et. al. [4], which is a binary vector  $x \in \mathbb{R}^{218}$  encoding the presence or absence of a hero in the “Radiant” and the “Dire” teams. Thus, the input feature vector can be written as:

$$x_i = \begin{cases} 1 & \text{if Team Radiant has hero with id } i \\ 0 & \text{otherwise} \end{cases}$$

$$x_{109+i} = \begin{cases} 1 & \text{if Team Dire has hero with id } i \\ 0 & \text{otherwise} \end{cases}$$

The output is a binary label  $y \in \mathbb{R}$  such that:

$$y_i = \begin{cases} 1 & \text{if Team Radiant won} \\ 0 & \text{otherwise} \end{cases}$$

## B. Making Predictions using Regression

We follow the same procedure to make a prediction as in [4] to predict the outcome of a match. We calculate the overall probability as the average of the winning probability and (1 - losing probability) with the teams interchanged and threshold it at 0.5 i.e., “Radiant” is predicted the winner if the overall probability is  $> 0.5$  and “Dire” otherwise.

## C. Co-occurrence network

A co-occurrence network is a frequently used visual representation of relationships between data in a graphical form from which their underlying hidden structures can be discovered using well-known algorithms. This form of representation is highly sought after when modeling relationships between many “nodes” which is flexible enough to represent several things.

In order to encode the relationship between pairs of heroes inherent in a match, we create nodes for each hero and increment the weight of that edge if both end points (heroes) co-occur in the same team [6] i.e., an edge between hero “15” and hero “85” exists if both co-occur in the same team in a match.

## D. Community Detection

Network communities are a part of network science that is used in predictive modeling of related phenomena. Many metrics exist that try to quantitatively evaluate the importance of a node in a network, such as the modularity of a node, its clustering coefficient, various centrality measures [7] etc. Nodes usually tend to cluster together in tight knit groups with “looser connections between them and such groups of nodes are referred to as “communities. Nodes in a community are said to be related due to their tight knit nature.

The Girvan-Newman community detection algorithm uses an edge-betweenness measure to progressively remove edges of highest betweenness from the graph [8] and repeat the process until the graph is disconnected. Newman and Girvan [9] developed a measure to evaluate the structure of a graph, known as the modularity  $Q$ .

$$Q = \sum_{i=1}^n (e_{ii} - a_i^2)$$

where  $e_{ii} = |\{(u, v) : u \in V_i, v \in V_i, (u, v) \in E\}| / |E|$

i.e., the percentage of edges in module  $i$  (probability that an edge is in module  $i$ ) and

$$a_i = |\{(u, v) : u \in V_i, (u, v) \in E\}| / |E|$$

i.e., the percentage of edges with at least one end in module  $i$  (probability that a random edge would fall into module  $i$ ).

Modularity is a measure of the density of connections between nodes. Nodes within a module are densely connected whereas nodes in different modules are sparsely connected. This measure is often used in community detection algorithms since modularity is a natural measure of the strength of connections between interdependent nodes within a community. Modularity optimization is an NP-Hard

problem [10]. We find the Girvan-Newman algorithm to be computationally inefficient with respect to large, dense graphs with a runtime of  $O(|V||E|^2)$  and use the Louvain community detection [11]. Uncovering communities is crucial because highly interconnected nodes of a particular community imply the existence of a pattern of gene elements among movies and this is an insight into how movies change over time, semantically.

The Louvain method is an efficient, greedy optimization based community detection algorithm which has been used with considerable success for very large networks (for up to 100 million nodes and billions of links). It uncovers hierarchies of communities and allows fine grained control over the size of communities, number of communities and the discovery of sub-communities. It involves a two-step optimization strategy where the first step optimizes modularity locally by searching for small communities and the second step performs node aggregation of nodes belonging to the same community. These two steps are iterated repeatedly until a target modularity is reached and a hierarchy of communities is obtained during this procedure.

We use the Gephi suite of tools to perform community detection on the graphs with a resolution measure of “1.0” to give us sets of communities in the graph. Resolution [12] is a measure of controlling the size as well as the number of communities desired in a graph and tweaking this parameter gives us a reasonable estimate of the number of communities and their sizes to expect.

## E. Hero Communities

We parse 250 DotA 2 matches and build the co-occurrence graphs for both the winning and the losing teams. On running the community detection algorithm discussed in the previous section, we obtain 7 distinct communities in both graphs. Community detection, we believe, helps us uncover hidden relationships between heroes in a match because a community essentially a closely related cluster of heroes that contribute to how successful or unsuccessful a match is going to turn out. The influence of nodes within such a community ideally describe the contribution of different heroes to the success or failure of a team.

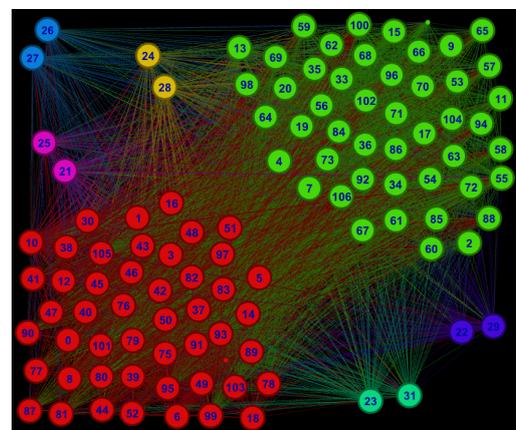


Fig. 3. The HCG of 250 victorious games

These seven distinct communities roughly correspond to some of the hero classes mentioned in section III. For instance, the community “24, 28” are the heroes “Lion” and “Witch Doctor” which are “Intelligence, support” heroes and frequently played together because of their synergistic abilities. The two big communities (red consisting of 45.79% of the nodes and green consisting of 44.86% of the nodes) roughly correspond to the general class of carry and support heroes respectively). We see that meaningful heroes, when picked together, contribute a lot more to the victory probability than individual picks and this is what we intend to uncover in the following algorithm i.e., the “success set of heroes” which contributes the most to victory. The relative sizes of the nodes are based on their eigen vector centrality, which indicates how “important” a node is in its own community.

#### F. Genetic Algorithm

The procedure described in Suman et. al.’s work on semantic information used to describe movies [13] is used in this paper to discover those set of heroes which contribute the most to victory. An initial population of 220 matches is selected, the fitness function is a modification of what is used in [13] i.e.,

$$fitness_x = 2 * \sum_{i \in WIN^{500}} \alpha_i - \sum_{j \in LOSS^{500}} \alpha_j$$

where  $i, j \in x$  are heroes occurring in match  $x$ ,  $\alpha_i$  is the eigen vector centrality of node  $i$ .

The standard genetic algorithm procedure is followed, whose steps are:

- Select an initial random population from the DotA 2 database. Set  $\phi = 10$
- Evaluate each population’s fitness
- While  $\phi > 0$  do
  - Filter with probability  $p_f$
  - Cross-over with probability  $p_c$
  - Mutate with probability  $p_m$
  - Generate new population on re-evaluating fitness
  - If no change in population,  $\phi = \phi - 1$

The selection probability  $p_f$ , mutation probability  $p_m$  and cross-over probability  $p_c$  are set according to [13] i.e.,  $p_c = 0.72$ ,  $p_m = 0.03$  and the selection probability according to the roulette wheel formula.

#### G. Making Predictions using Augmented Regression

After running thousands of rounds of evolution, we obtain a set of heroes which have the highest probability of winning matches i.e. a “success set”. For each new match, we define a new measure of success as follows:

$$Success\ prob = \frac{|\text{heroes picked} \cap \text{success heroes}|}{|\text{heroes picked}|}$$

From Kevin et. al.’s work, we have an overall win probability from applying simple regression to a binary feature vector and from the genetic algorithm, we have a success

probability. Combining the two success probabilities, we have the following prediction algorithm:

$$Final\ Success\ Prob = \frac{\text{regression prob} + \text{success prob}}{2}$$

If  $Final\ Success\ Prob > 0.5$ , we predict a “Radiant” win otherwise a “Dire” win otherwise. This metric takes into account, the individual contributions of heroes as well as a combined measure of how well heroes compliment one another in a match.

## VII. RESULTS

### A. Pure Logistic Regression

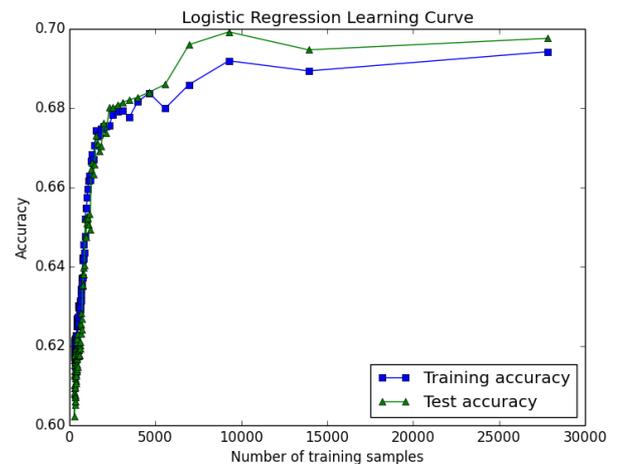


Fig. 4. Training accuracy vs. the #training samples for simple logistic regression

The test accuracy asymptotically approaches 69.42% and does not change much with an increase in the training set size. Also, the regression model does not overfit the training data as seen from the training accuracy. This algorithm merely captured the individual influences of heroes in a game and seems to do pretty well considering how well it models a game so complex, with a nearly infinite number of interactions and effects, many of which have not been modeled or taken into consideration. We have used simple logistic regression, without regularization from the sklearn.linear\_model library. From the observed results, we are able to conclude that just hero selection plays an important role in determining which team wins. In the next section, we talk about the results of the augmented regression model, with the added data from the genetic algorithm, to see if it has any impact on the predictive power of our model. An easily noticeable anomaly in this plot is that the test accuracy seems to be higher than the training accuracy for almost all training set sizes. From a Bias-Variance standpoint, this shows that our model has a high variance. Another explanation for this “anomaly” could be that the test set is pathologically similar to a subset of the training set.

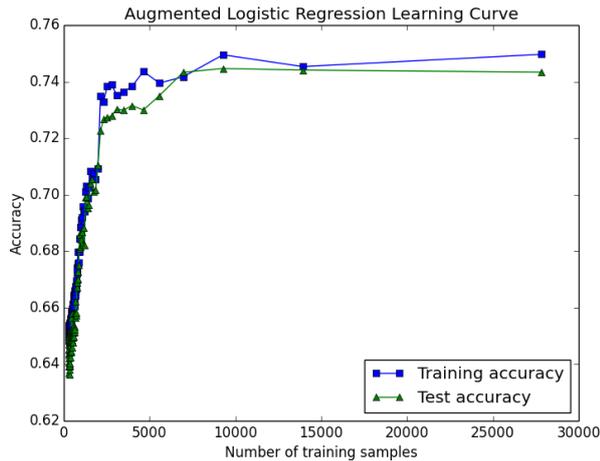


Fig. 5. Training accuracy vs. the #training samples for the augmented prediction algorithm

### B. Augmented Logistic Regression

We see that both training accuracy and test accuracy have significantly improved after augmenting the prediction procedure of the logistic regression with the output of the genetic algorithm. This is not surprising, as a purely weighted sum of feature vectors (logistic regression) does not capture hero synergistic relationships at all whereas the “success-set” of heroes captures such dual relations between heroes. The test accuracy asymptotically approaches a value of 74.1% and surprisingly, the trend of the training accuracy being less than the test accuracy is reversed here, when compared to the pure logistic regression model. We do notice that the overall prediction accuracy of the entire set has improved over the baseline model. Capturing additional relationships existing in the underlying data is indeed useful in improving the predictive ability of our model.

	Regression	GA	Augmented Regression
Precision	0.752	0.741	0.684
Recall	0.787	0.762	0.909
F-Score	0.769	0.752	0.781

TABLE I

TABLE I: PREC, RECALL AND F-SCORES

From the F-Scores for the different models, we see that the augmented regression model outperforms the baseline model of pure regression and a model based just on the genetic algorithm. This is expected as it combines information from both models and this information essentially compliments one another and describes the underlying data distribution better. We see that the precision score of the augmented model is less than the other two but it has a recall score of almost 1. The augmented classifier has a very low false negative rate i.e., in the absence of particular “success hero pairs”, the probability that a team is predicted to lose is less or in simpler terms, the model has correctly predicted that a team with a really bad hero composition will lose very

often. The low precision score which signifies a higher false positive rate, seems to indicate that the model predicts victory for a team if the team has a significantly high intersection with the “success hero pairs” but the underlying data set does not seem to indicate so. A reason for this could be that, as described earlier, a DotA 2 match is highly complex and we are not utilizing any other information than just the team compositions. Merely predicting with a high probability, that a team would win, based on its composition is not as effective as it seems which once again reiterates our initial hypothesis that it is insufficient to just consider hero picks when trying to predict the outcome of a game.

## VIII. CONCLUSIONS

A survey of existing literature on predicting the outcomes of DotA 2 matches is presented here, along with an improvement to an existing simple logistic regression model. We apply a graph-based genetic algorithm to discover the “most successful” set of heroes which essentially captures the various complimentary and antagonistic inter-relationships between DotA 2 heroes.

Furthermore, we note that even this model, which captures but a fraction of the myriad factors that influence the outcome of match, performs significantly well compared to the alternative baseline regression model. This shows us that hero composition plays a tremendously important role in tipping the scales of victory towards either team and should not be neglected in matches. The results do suggest that using more game data could be beneficial and improve the predictive power of the model, as in-game factors also affect game outcomes to a good extent.

## IX. FUTURE WORK

There are several promising directions for future research into this area. One of the ideas we had was for an online prediction algorithm that takes into account current game state such as the experience and gold difference, the current experience level of critical heroes and the number of towers lost/gained. This algorithm would switch the results based on the data it receives and would be interesting to see what trends in a game lead to a critical shift in the match outcomes.

Another possible interesting future direction we had is modelling a match as a reinforcement learning problem and considering a player as an autonomous agent. At each time epoch of the game, this agent can have one of several available actions to perform, such as “farm”, “defend”, “push” etc. Such an algorithm could potentially revolutionize the way newcomers approach DotA 2 in that they could view the actions of this agent based on current game state and learn quickly a pattern of inferences similar to what the agent makes. This would be invaluable in aiding newcomers in understanding the very complex mechanics underlying the game.

## APPENDIX

ID	Name	ID	Name	ID	Name
0	AM	36	BM	72	Invoker
1	Axe	37	QOP	73	Silencer
2	Bane	38	Venomancer	74	OD
3	BS	39	FV	75	Lycan
4	CM	40	Skeleton King	76	Brewmaster
5	DR	41	DP	77	SD
6	ES	42	PA	78	LD
7	Juggernaut	43	Pugna	79	CK
8	Mirana	44	TA	80	Meepo
9	Morphling	45	Viper	81	Treant
10	SF	46	Luna	82	Ogre Magi
11	PL	47	DK	83	Undying
12	Puck	48	Dazzle	84	Rubick
13	Pudge	49	CW	85	Disruptor
14	Razor	50	Leshrac	86	NA
15	Sand King	51	Furion	87	Naga Siren
16	Shadow Shaman	52	LS	88	KOTL
17	Sven	53	DS	89	Wisp
18	Tiny	54	Clinkz	90	Visage
19	VS	55	Omniknight	91	Slark
20	WR	56	Enchantress	92	Medusa
21	Zuus	57	Huskar	93	Troll Warlord
22	Kunkka	58	NS	94	Centaur
23	Lina	59	Broodmother	95	Magnataur
24	Lion	60	BH	96	Shredder
25	SS	61	Weaver	97	Bristleback
26	Slarder	62	Jakiro	98	Tusk
27	Tidehunter	63	Batrider	99	SM
28	WD	64	Chen	100	Abaddon
29	Lich	65	Spectre	101	ET
30	Riki	66	AA	102	LC
31	Enigma	67	Doom Bringer	103	Techies
32	Tinker	68	Ursa	104	Ember Spirit
33	Sniper	69	SB	105	Earth Spirit
34	Necrolyte	70	Gyrocopter	106	TB
35	WL	71	Alchemist	107	Phoenix
108	Oracle	-	-	-	-

TABLE A1

TABLE SHOWING MAPPING OF HERO IDS TO NAMES

## REFERENCES

- [1] [http://en.wikipedia.org/wiki/Defense\\_of\\_the\\_Ancients](http://en.wikipedia.org/wiki/Defense_of_the_Ancients)
- [2] Dota2 Counter-Pick. Dota2 Counter-Pick. N.p., n.d. Web. 14 Nov. 2013. <http://dota2cp.com/>.
- [3] [http://wiki.teamliquid.net/dota2/The\\_International/2014](http://wiki.teamliquid.net/dota2/The_International/2014)
- [4] Kevin Conley, Daniel Perry, "How does he saw me? A Recommendation Engine for Picking Heroes in Dota 2", N.p., n.d. Web. 7 Dec. 2013.
- [5] <http://dev.dota2.com/showthread.php?t=58317>
- [6] Cohen, A. M., Hersh, W. R., Dubay, C., & Spackman, K. "Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts." BMC bioinformatics 6.1 (2005): 103.
- [7] Newman, Mark EJ. "Modularity and community structure in networks." Proceedings of the National Academy of Sciences 103.23 (2006).
- [8] M. Girvan, M.E.J. Newman, Proc. Natl. Acad. Sci. USA 99, 7821 (2002)
- [9] M.E.J. Newman, M. Girvan, Phys. Rev. E 69, 026113 (2004)
- [10] Brandes, Ulrik, et al. "Maximizing modularity is hard." arXiv preprint physics/0608255 (2006).
- [11] Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks." Journal of Statistical Mechanics: Theory and Experiment 2008, no. 10 (2008): P10008.
- [12] Lambiotte, Renaud, J-C. Delvenne, and Mauricio Barahona. "Laplacian dynamics and multiscale modular structure in networks." arXiv preprint arXiv:0812.1770 (2008).
- [13] Roy, Suman Deb, and Wenjun Zeng. "The Hidden Potential of Movie Genome Communities: Analyzing Fine-Grained Semantic Information in Motion Pictures." Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on. IEEE, 2013.