Dota2 Match Outcome Prediction with ML

Member 1	Muhammad Ali Qadri	muhammadaliq
Member 2	Suchith Suddala	suchiths
Member 3	Xiaolu Dedman	xiaolu92

Introduction

The newly emerged area competition, ESports, is an organized and competitive video gaming world that millions of people around the world are taking part of. Esports come in various genres and have exploded in popularity over the last decade, with massive tournaments of superstar Esports teams and Esports players, professional athletics, competing in front of a live audience and millions of fans on live streams such as Twitch.tv, YouTube Gaming, or Facebook Gaming. [8]

The most popular genre among ESports is Multiplayer Online Battle Arena or MOBA games, with titles like Dota 2 and League of Legends, which are some of the biggest names in ESports. At the tournament or league level, these games have a viewership of above 3 million viewers, watching over 130 million hours of competitive gameplay, with Dota 2 having a prize pool of $40 million for the 2021 world tournament. [9]

Dota 2 is played by two teams of five players each. These two teams play against each other to win or lose the match. Each player can choose a hero out of 121 heroes that have different skills and rankings. Choosing to invest or sponsor a potentially winning team will be highly profitable and critical for both teams/players and investors/sponsors. That is why using Machine Learning to predict a match outcome will be vital for the team itself to have an edge to win. So our goal is to create a supervised machine learning algorithm to determine match outcomes in Dota 2.

Problem Statement

Thousands and thousands of Dota2 matches are played world wide every day, if not every hour. Each of these matches had varying characters, from the map, type of game, player to the characters chosen by the players. Each attribute of a match has an impact on the result of the game: win or loss. The end goal of this project is to create ML algorithms that predict match outcomes based on the teams' selection heroes, team and match characteristics. Specifically, given two sets of five heroes, team statistics, and the characteristics of the match, the algorithm would predict which team is more likely to win against the other team. The successful construction of this algorithm would improve the number of victories for both teams and players.

The resulting ML prediction model can be used by stakeholders, such as potential Esports sponsors, to find the best combination of players to win a match within a certain budget. They can utilize the model to test different variations of players and teams, in order to weigh their options and recruit teams. Similarly, individual professional players themselves could also leverage this algorithm to find the best team builds, using publicly available player statistics on team mate prospects. There are a few ways we can evaluate these improvements made by using our ML results. One obvious measure of improvement would be the number of tournament victories won by the teams, but more importantly, the individual players of the winning team. Additionally, we can compare the performance of the player to the average performance of the hero chosen in recent games.

Dataset

The Dota2 dataset of matches played by pro-teams were all fetched from the Dota2 API, which allows for both traditional API calls and SQL queries. Although the API provides access to detailed match and player statistics, its access is limited to 60 calls per minute. In contrast, the SQL query structure allows us to fetch all instances at once, it restricts access to player statistics. Initially, the goal was to retrieve match and player statistics for every match individually to build the dataset; however, due to the call limit, fetching a few hundred thousand records would require dedicating 80-100 hours of computation time. So, while the API gives access to larger feature space, the SQL query allows for feasible access to large amounts of instances. In order to properly consider all elements that affect a Dota2 match, we created and examined ML models with two datasets: one small and one large.

While the features and the size of the dataset are different, there are still commonalities between both. Each instance represents a pro-match that is played by two opposing pro-teams, identified as dire and radiant. The target variable for both is just a boolean indicating whether the radiant team won or not. Additionally, after retrieving the necessary data from the API, the datasets were doubled by duplicating the exacting matches and switching the radiant and dire teams. This preprocessing step not only doubled our dataset, but it helped remove any bias that may have naturally occurred in the curation process or by the API.

* The shapes described below are the result of the preprocessing steps. Since we put together the datasets ourselves the original shape of the data is ambiguous.

Dataset 1

The first dataset was built using multiple Dota2 API endpoints. It contains 844 instances with 213 features, that were constructed of the following thing:

1. Player statistics for every member of both radiant and dire teams
    a. Assists
    b. Deaths
    c. Kills
    d. Competitive rank
    e. Solo Competitive rank
    f. Mmr estimate* (a numerical measure of player's skill dedicated by Dota2)
    g. The win-Lose ratio of a player with their chosen hero
2. Game mode (eg. All Pick) represents the match environment during gameplay. This is nominal data, represented numerically from numbers ranging from 0-24.
3. Game League (Premium or Professional)
4. Cluster-ID (related to location) Represents the region in which the match took place. (eg. US EAST)
5. The rest of the columns indicate whether one of the 121 characters are being used in the game
    a. 0 for none
    b. 1 for team dire
    c. 2 for team radiant

Dataset 2

The second dataset was constructed by using the SQL query endpoint. It contains 209890 match instances and 324 features, primarily made out of the following:

1. Team statistics for both dire and radiant team
    a. Dota2 rating
    b. Total number of wins
    c. Total number of losses
2. Lobby Type
3. Game mode (eg. All Pick) represents the match environment during gameplay. This is nominal data, represented numerically from numbers ranging from 0-24.
4. Game League (Premium or Professional)
5. Cluster-ID (related to location) Represents the region in which the match took place. (eg. US EAST)
6. The rest of the columns indicate whether one of the 121 characters are being used in the game
    a. 0 for none
    b. 1 for team dire
    c. 2 for team radiant

Dataset 1 vs Dataset 2

Although we initially started off building two different datasets due to computing limitations, it provided us with a great opportunity to explore the value of all the features. As presented above there is an abundance of statistics related to a single Dota2 match, each of which affects the result of the game differently. In order to properly use ML models to predict the outcome of a match or even just scout for players and teams, it helps to know which statistics help dictate that outcome.

While both datasets are designed to be used for the same purpose, the first provides a lot more detailed statistics than the second. The first dataset includes various statistics on the skill and ranks of each individual player of both teams along with general match environment details. In contrast, the second dataset simply contains three statistics for the team as a whole. However, the second dataset maintains its own advantages. The dimensions of the first dataset tower the second, but it also contains a lot of missing data, since every player's statistics are not readily available. Meanwhile, the second dataset contains nearly no missing data and confident high-level team statistics for nearly 200K matches. It could be argued that simple ratings are the best predictors, compared to the through-player details.

Preprocessing steps

Due to the structural differences between the datasets the preprocessing steps also varied greatly between the two. However, in both the instances were first doubled by simply switching the all dire values to radiant and vice versa; the target values were flipped as well. This not only provided more instances but helped remove any natural bias. Additionally, the 121 heroes' chosen attributes were structured to use 0, 1, 2 to indicate a choice, but after experimenting with one-hot encoding those columns, we found that dataset 1 performed much better without encoding, and dataset 2 performed slightly better with encoding. Alternate dataset values are included in Appendix A. The encoding was done such that hero choice attributes were created for both dire and radiant teams (121 columns each), and each value was 0 or 1 (not chosen or chosen). Furthermore, to increase dimensionality, the location clusters were label encoded to discrete numerical values, then one-hot encoding was applied to the cluster, league, and game-mode attributes to create 28 new attributes. Next, the match id feature was removed, as it is a clearly irrelevant value that is arbitrary. Lastly, before the training, the classification models on either dataset each fold in the cross-validation was scaled and normalized, primarily to ensure logistic regression converged. For the rest of the models (other than ANN), it was simply due to consistency; the other models are not affected by normalized scaling, so no error is created.

Nothing additional was done for preprocessing dataset 1, but more substantial changes were made to dataset 2, before feeding them to the ML models, mostly using the scikit-learn python library. When fetching the team statistics from the Dota2 API, they are retrieved for each individual player on every team. However, since the order of the team members' statistics is inconsequential to the prediction, each team's statistical attributes (average, standard deviation, max, min) were calculated and stored instead, so as to not confuse the ML models. Unlike dataset 1, this process created missing values for teams that had players with no game statistics stored in the API. These values were simply imputed with the mean of the column since the rest were all pro matches as well. Lastly, dataset 2 contained a lobby type attribute, which only contained one value, so that was removed as well.

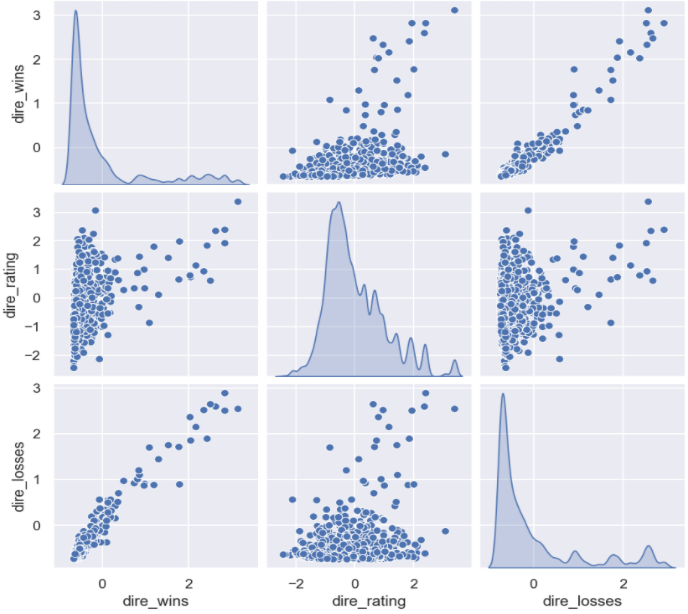


Figure 1

Feature analysis

Figure 1 tells the distribution of different meaningful features against one another.

This shows the overall spread in our dataset. We can clearly see in this dataset that any rating factor involved does not impact winning or losing a match.

Most of the dataset wins and ratings are centered around the mean indicating that each team does not vary too much from another. Also, an interesting thing seen here is that winning and losing are almost linearly related to each other.

We conducted experiments with finding similarities between our target variable and with each of the heroes, we have an average similarity score of **0.5** by using SMC (simple matching coefficient) for similarity score as each of these features is binary**.** This indicates that selecting a specific hero has a low effect on the outcome of a game. This again complements the variance seen in our dataset. Furthermore, for our analysis of continuous variables, we calculated the Pearson correlation coefficient for each column, in a matrix. [Appendix B] In both datasets, while the hero variables are not as correlated to many others, the attributes pertaining to team statistics are correlated to each other.

We can clearly see in Figure 1 above that none of the columns have a strong correlation between them and the target variable, and hence all will be vital for predicting. Also, another interesting thing is that some columns are correlated with each other.

PCA Analysis

PCA is done to reduce the size of the dataset significantly and to make the models less complex. Due to the very high dimensionality of our dataset, this was a vital step to include in our preprocessing.

We can clearly see in the above graph that PCA without standardization captures much more variance than with Standardization.

The above result shows more than 95% of the variance in the data can be expressed just by the first 2 components. This happens in cases where there is high collinearity; the presence of collinearity can cause the PCA to overemphasize the contribution of the variance from the highly correlated (or redundant) variables and gives less weight to the variables to the uncorrelated variables. This ultimately influences modeling/prediction results if such data is used.

Even if we utilize variance capture from unstandardized PCA, we will not be able to capture true variance in the dataset, and even the plot doesn't make sense here.

Hence now when we do standardization, we lose our variance capture. As we can see in the graph above, the variance capture is only 40% for 100 features. So this approach will not be fruitful in our modeling.

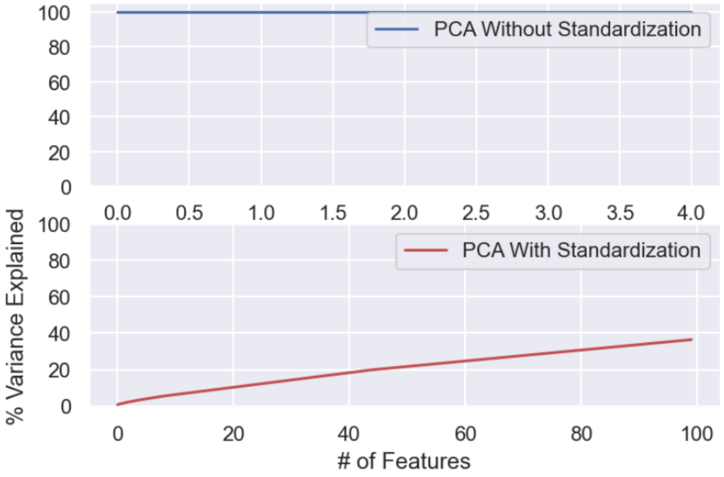


Figure 2

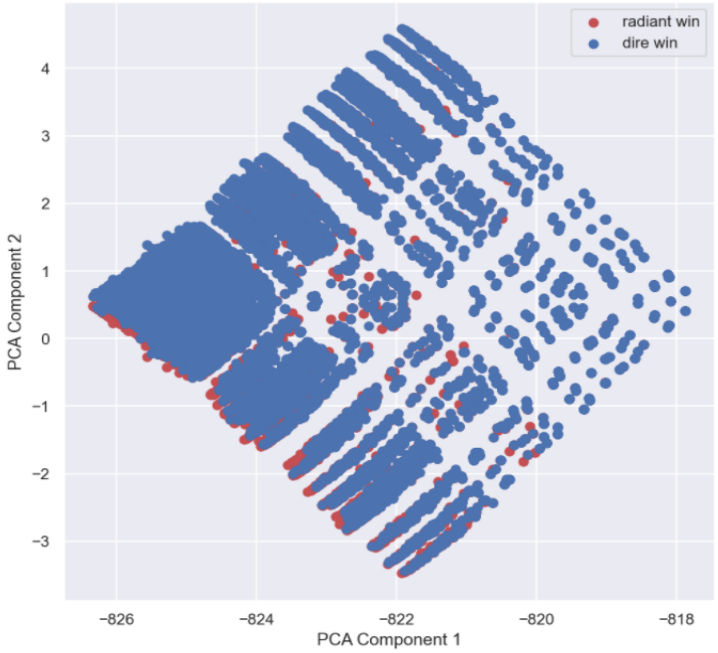


Figure 3

Methods and Models

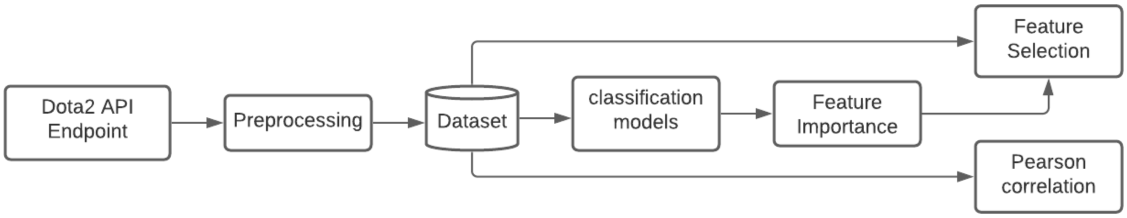


Figure 4: Project Pipeline

In order to accomplish our goal of predicting which team variations (team players and their chosen heroes) have the best probability of winning a match, we model it as a supervised ML problem. 5 different classification models on two datasets were built to accomplish this task. The datasets resulting from preprocessing are used as the features/independent variables, while the columns of whether the dire team won or lost are the target variables, in the respective datasets. For training, we will have the models classify a match between two teams as a win or lose. As in previous works, we will fit Decision Trees, Random Forest, Naive Bayes, and an Artificial Neural Network.[5,6] Logistic regression was fit as well to add variation.

All models were trained and tested with scikit-learn and Tensor flow (ANN). For each model, 5-fold cross validation with shuffling of the folds was used to train, test, and score each model, to guarantee randomness and generalization. Within each fold, all the features were normalized using the standard scaler to ensure that the logistic regression model converged. Lastly, each model was scored with precision, recall, and f1 to evaluate and compare the effectiveness of the models on the classification task at hand. However, we mainly discuss f1 scores as its a summary, precisely a harmonic sum, of both precision and recall.

Model 1: Decision Tree

A default decision tree from the scikit-learn library was utilized for this task, meaning no max-depth was set, so as to compare the results with the random forest model. We tested and recorded dataset 1 and 2 using the specified model and recorded the F1 scores in Table 1. As you can see, the decision tree works a bit better with dataset 2 than 1. As the number of instances increased in Dataset 2, the F1 score also increased slightly even when important features were dropped. This means that this modeling method is less sensitive to feature importance and cares more about the quantity of instances in our case. The F1 score of 0.5 - 0.605 is within our expectations and similar to other researchers results in Dota 2 match prediction.[7]

Dataset 1	**Dataset 2**
Recall: 0.4856 Precision: 0.5192 F1: 0.5003	Recall: 0.6044 Precision: 0.6051 F1: 0.6047

Table 1 - Decision Tree

Model 2: Random Forest

A default random forest model from the scikit-learn library was used for this model. We performed the same operations on datasets 1 and 2 using the random forest model and recorded the F1 scores in Table 2. Similar to Decision Tree, as the number of instances increased, the F1 score increased tremendously. It is surprising that Random Forest performed so badly with dataset 1 as it contains more meaningful features, but this also tells us that the Random Forest is more sensitive to the quantity of instances than the number of meaningful features. On the other hand, Random Forest performed very well with dataset 2 resulting in an F1 score of 0.665, which is the highest F1 score we got from all models.. This score suggests that Random Forests performed better than Decision Tree when the dataset is big enough with quality instances, while Decision Tree performed more stably overall with changing features and instances in our case.

Dataset 1	**Dataset 2**
Recall: 0.1099 Precision: 0.1040 F1: 0.1058	Recall: 0.6729 Precision: 0.6566 F1: 0.6646

Table 2 - Random Forest

Model 3: Naive Bayes

Furthermore, we modeled datasets 1 and 2 with a Gaussian Naive Bayes model from the scikit-learn library. As shown in Table 3, the Naive Bayes model didn't perform as well as the Decision Tree and the Random Forest models in general. Both F1 scores are lower than 0.4; however, the F1 score increased slightly with the increase of numbers of instances just like Decision Tree and Random Forest did. So, similarly this indicates that Naive Bayes is more sensitive to the number of instances than the meaningful features just like Decision Tree and Random Forest for the Dota2 classification task. Also Naive Bayes performed pretty stably between the two datasets with an F1 score in the range of 0.33 - 0.38.

Dataset 1	**Dataset 2**
Recall: 0.3375 Precision: 0.3783 F1: 0.3356	Recall: 0.5473 Precision: 0.4394 F1: 0.3787

Table 3 - Naive Bayes

Model 4: Logistic Regression

In one of our literature reviews, we learned that Logistic Regression was used as the main modeling method to predict Dota 2 match outcomes by several machine learning teams.[7] So, we added this model as well to test its performance against others in accomplishing this classification task. The results in Table 4 show the scores of the scikit-learn Logistic regression model with a max iteration limit of 1000, to ensure the model converges. The model evidently performed much better on dataset 2, that it did dataset 1. This is to be expected, especially compared to dataset 2, since there are not enough instances to properly support the gradient descent of the loss process. However, this discrepancy supports that Logistic Regression is less sensitive to the number of meaningful features and more sensitive to the number of instances, in our classification.

Dataset 1	**Dataset 2**
Recall: 0.2622 Precision: 0.2662 F1: 0.2619	Recall: 0.6362 Precision: 0.6353 F1: 0.6358

Table 4 - Logistic Regression

Model 5: Artificial Neural Networks

We use ANN to predict the accuracy of both datasets. We use multiple layers with each layer having a decreasing number of perceptrons from first to last. With the first having perceptrons equal to the total feature size, and the last having the 1 perceptron for a boolean output.

Each layer has an activation function of Rectified linear unit (RELU), and the last layer has a 'SIGMOID' activation function. The model has been trained with an 'SGD' optimizer function, runs for 30 epochs, and has a validation set which is 20% of the training data. The models were trained with a 5-Fold cross-validation scenario and loss was plotted for only the last fold for brevity.

Since dataset 2 is very large and the network is complex, it takes an estimated 75 minutes to even reach 30 epochs. Here is the error plotted against epochs, left for dataset 1 and right for dataset 2.

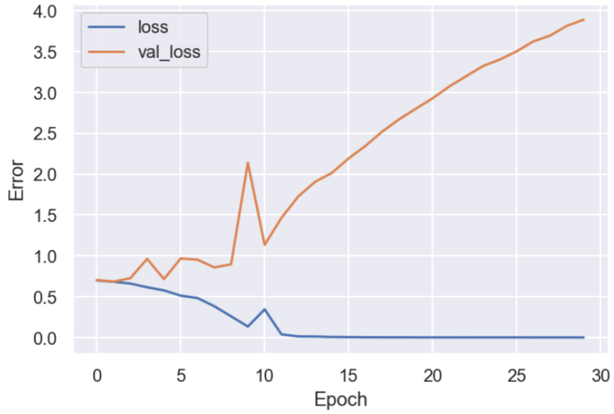


Figure 5

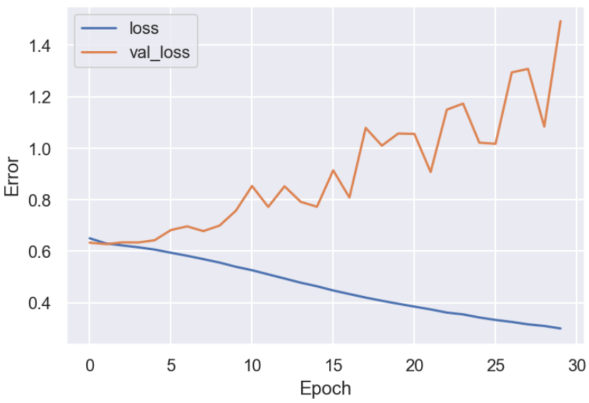


Figure 6

We can see that the models learn perfectly well on the training set, but as the training accuracy increases, the validation error increases as well.

Dataset 1	Dataset 2
Accuracy: 0.425 F1: 0.414	Accuracy: 0.569 F1: 0.551
Loss: 3.64	Loss: 1.307

Table 5 - ANN

From the results shown above, we can see that the loss and accuracy have comparatively increased in dataset 2. Also, there is a big improvement of F1 measures from dataset 1 with 0.414 to dataset 2 with 0.551. We can clearly see here that the increased accuracy for the second dataset is due to an increased number of data points available for training and validation.

Feature Importance and Selection

Not all attributes are important to predict whether a match will result in a victory or not. The determination of the most important attributes is valuable information to stakeholders who want to determine the most capable team. After running each of the 4 classification models, logistic regression, decision trees, random forest, and naive Bayes, the coefficients that are inherently calculated during training determine each feature's importance to the respective classifier in predicting the class. Using the scikit-learn python library, each of those coefficient lists was retrieved and used as a feature importance ranking.[10] Although the scores of the classification models are not high, they are sufficient enough to use the models' coefficients to evaluate feature importance. There are many methods available for feature selection, but the ones that provide feature ranks rather than help remove unnecessary features are needed here, so Pearson correlation would not be useful for the task. Lasso regression can be useful, but it results in all coefficients of 0, which provides no valuable information. However, the similar ridge regression doesn't shrink coefficients to 0, rather scales to their importance, so that is used instead.

While feature importance was crucial to determining what attributes are essential for the victory prediction, primarily for stakeholders, it can be further extended to apply feature selection before training the classification models. After running each of the 4 classification models, logistic regression, decision trees, random forest, and naive Bayes, we Aside from the classification coefficients, algorithms such as the Pearson correlation and lasso regression are excellent techniques for feature selection. Lasso regression attempts to shrink every coefficient to 0, based on its importance, so any that are not 0 by the end is clearly most important. However, for reasons discussed before lasso regression does not work for our datasets. In contrast, the collinearity matrix produced by the Pearson correlation would be extremely useful in removing columns that are redundant. Feature selection was implemented with scikit-learn as well. From each of the 5 feature selection methods, we retrieved the top 10% of important features (respectively determined): 21 features for dataset 1 and 32 features for dataset 2. Then to appropriately measure each one's effects, the 4 classification models were trained and fitted again with the new set of features to predict the same classes: win or lose.

Results

F1 Scores	Logistic Regression	Decision Trees	Random Forest	Naive Bayes	ANN
Dataset 1	0.262	0.500	0.106	0.336	0.414
Dataset 2	0.636	0.605	0.665	0.379	0.551

Table 6

In an overview, it is glaringly obvious that the classification models perform much better when predicting the outcome of a match with dataset 1 than with dataset 2. The highest F1 score in dataset 1 is .500 for the decision tree, while the highest F1 score is .665 on the random forest model for dataset 2. Contrary to the fact that dataset 1 contains more informative attributes, primarily the statistical summary of each team's player statistics, the models fail to effectively predict the outcome of the match. These discrepancies between the scores of dataset 1 and 2, highlight that it is more valuable to have a larger set of instances with a few meaningful features than it is to have a large degree of meaningful features and a small number of instances. Alternatively, this could be that the concentrated attributes describing both dire and radiant's skill (rating, wins, losses) are more crucial to determining future victories, compared to the detailed statistics of every player on a team. However, we cannot conclude without additional testing outside the scope of this report. Furthermore, the exceptions to the observations above, Naive Bayes scored consistently for both datasets, but still very poorly. This is most likely due to the abundance of hero attributes, which are heavily correlated to each other, which affects the Naive assumption of the model. The model's confusion is also visible in the feature analysis in Figure 7 and Figure 8. When other models clearly found team statistics to be more valuable, Naive Bayes failed to clearly and abundantly distinguish the importance of hero chosen attributes from team statistics. The multi-layer neural network performed comparatively better on the second dataset. This can be attributed to the fact that we had much more observations to feed to the neural network. The overall scores were much better for any given metric, and we sense that if we were to have the feature selection from dataset 1 into dataset 2, the neural network would outperform the rest of the models.

Feature Importance

The rankings for each dataset are distinctive. With dataset 1, every model has a different set and ordering of top rankings as can be seen in Figure 7. The top 5 important features for each model are listed in appendix B. While no one ranking is put forward by all models, the max win-lose hero ratio feature was at the top for random forest and decision trees, which has the highest f1 scores for dataset 1. The rest of the rankings however vary too much. If we are to simply value the sets that are put forward by the highest-ranking models, the win-lose hero ratio statistics and competitive rank are valued the most. These features describe the skill of each team and will be excellent markers for the performances of future teams. Furthermore, the other features that are ranked at the top are individual heroes, Ember Spirit being at the top. Considering the scores of the models on dataset 1, these heroes ranked high are pivotal characters in the game that help determine the outcome of the game or are just part of the error. If they are not, it is most likely that a subset of those characters is pivotal to winning a match. For every scikit-learn model, the coefficients from the 5-fold cross-validation were averaged to calculate the feature importance ranking. For the logistic regression and Naive Bayes models, the absolute value of the coefficients was used, since the feature contribution to each class is not pertinent to the comparison we are making. The figures above show the top features in each model.

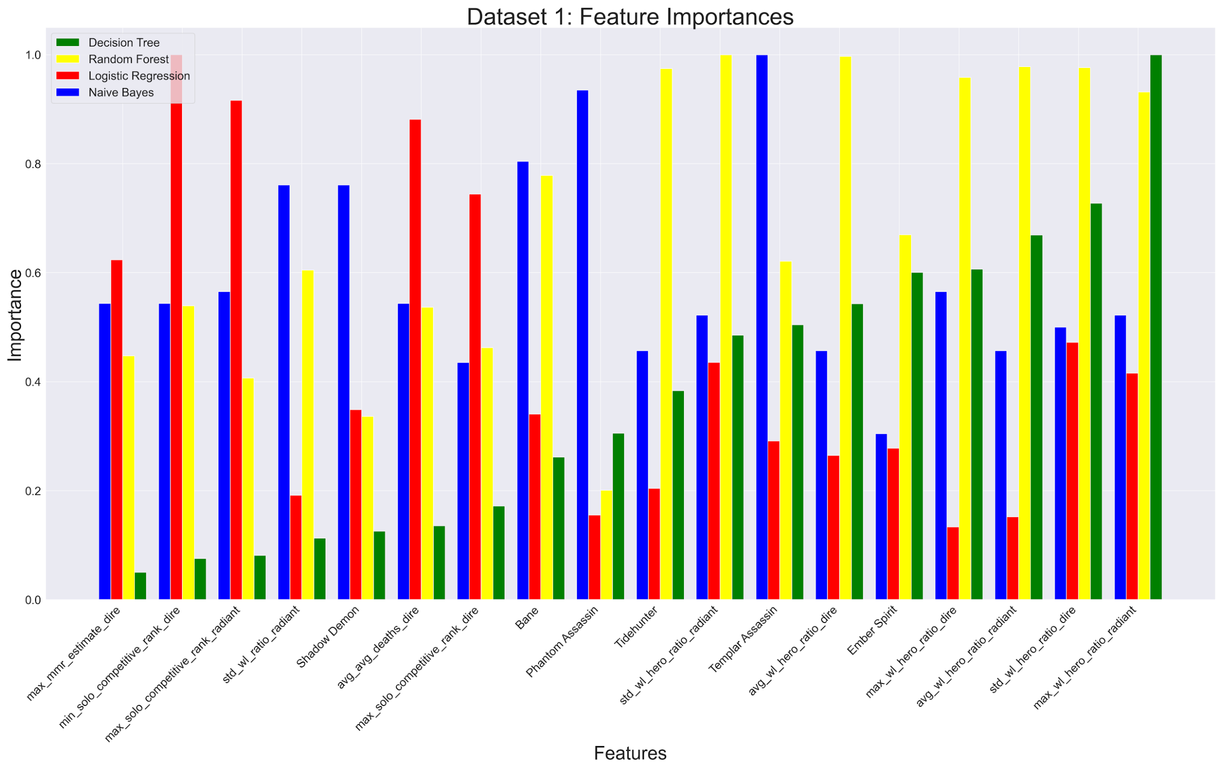


Figure 7

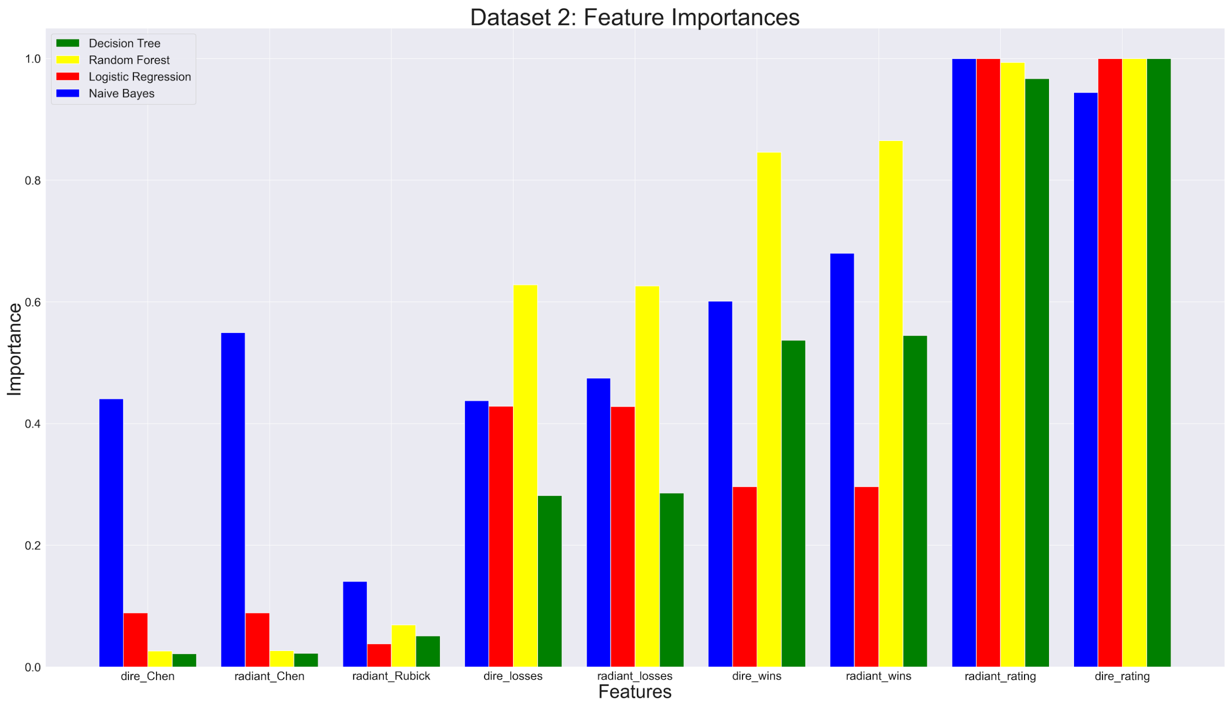


Figure 8

While there is no clear pattern in dataset 1, there is a very recognizable ranking of the attributes amongst most models in dataset 2 as can be seen in Figure 8. Similarly, all the models scored poorly for the first dataset, while the F1 scores for dataset 2 were much higher; Random Forest being at the top. The most evident pattern shows that the team ratings are the most important in predicting the result of a match. The next in the ranking is simply just the wins and losses of each team. The remaining attributes, chosen heroes, clusters, etc. are ranked significantly lower than the rest, by most models other than Naive Bayes. However, in both datasets Naive Bayes had the lowest F1 scores, so its rankings may not be as valuable. In overview, these rankings [rating, wins, and losses] are the most indicative of the future performance of teams in their matches. For stakeholders, such as E-sports scouts these would be the features to focus on when recruiting.

Feature Selection

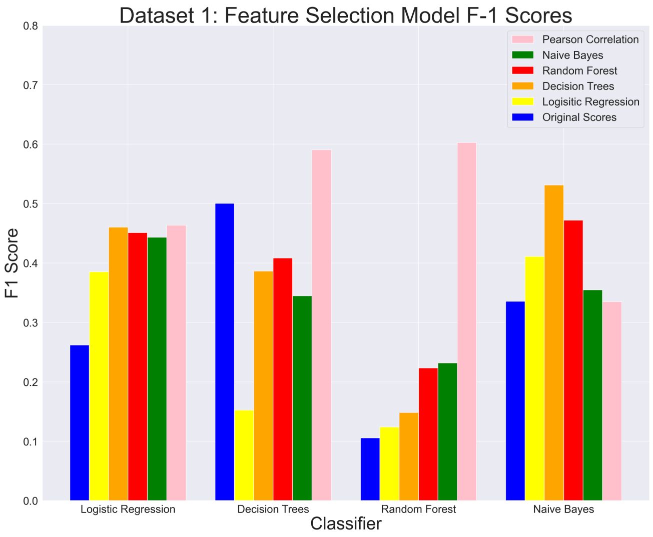


Figure 9

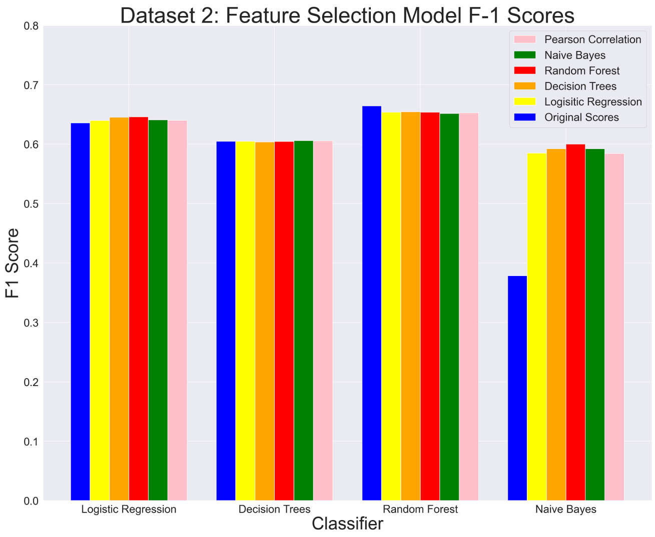


Figure 10

It is clear that feature selection has entirely different impacts on dataset 1 than dataset 2. For every classifier in dataset 2, other than Naive Bayes, Figure 10 negligible improvements regardless of the feature selection method. However, for Naive Bayes it is evident that every feature selection method significantly improves the model, they all jump from an original f1 score of .379 to approximately .6. [Appendix C] The behavior is unusual since Naive Bayes models are usually resilient to irrelevant features. It is possible the immense amount of unnecessary features or the correlation between features (defeating the Naive Bayes assumption) is affecting the performance of the model. With these improvements, Naive Bayes now scores similarly to the other features, which are generally still poor.

Unlike dataset 2, nearly all feature selection methods provided an improvement of the original classification models, with 3 selection methods being the exception on decision trees, as can be seen in Figure 9. The f1 score of the logistic regression model jumped from a low of .262 to a high of .464 when Pearson correlation was applied. The other methods provided similar improvements as well [Appendix C], x with the exception of logistic regression which did slightly worse at .385. Regardless of the improvement, every method improved upon the original model. Similarly, the Naive Bayes model was optimized by all the feature selection methods, other than Pearson correlation. The f1 scores improved from the original of .334 to a high of .531 from the decision tree selection method. This is most likely due to the removal of significant irrelevant attributes, rather than redundancy since the Pearson correlation showed no improvement at an f1 score of .335. While the previous two classifiers were improved by most of the feature selection methods, the decision tree classifier was only improved by Pearson correlation; the score improved from the original of .500 to .590. Although not as significant as the others, the removal of redundant features by Pearson correlation reduced any overfitting of the original model, as the original was trained without a specific depth. As expected, decision trees are resilient to irrelevant attributes, so the other feature selection methods made no improvements. For the previous classifiers, there was at most an improvement of 45%, but Pearson correlation improved the Random forest model by approximately 140%, from an original score of .1057 to .6028 with Pearson. The classifier feature selection models improved scores, but at most to an f1 score of .232 by Naive Bayes. As with decision trees, we suspect it is due to the removal of redundant features.

Conclusion

It is evident that, in every part of the experimentation, that the results for dataset 2 are much more stable than that of dataset 1, which is in large part due to the abundance of instances and concentrated attributes. Even with feature selection, the models trained on dataset 1 don't score higher than those trained on dataset 2, although close with Pearson correlation. This does not invalidate the results of dataset 1, but emphasizes that it is more important to have a larger set of instances with fewer meaningful features rather than a smaller set with a large number of meaningful features. Ideally, a combination of the takeaways from both dataset analyses would be most useful to stakeholders.

Further, we believe that our modeling results from Dota 2 match predictions can be used by Esports teams and management teams to improve their team performances and increase their winning records. More specifically, they can use the models to help them predict their chances of winning with a specific team of players with their chosen heroes against another team. This way, they can find the best combination of heroes and players against a certain opponent team in a match. Similarly, professional players could utilize this ML model to run variations of team member prospects, with their publicly available game statistics, in order to build the most effective team. In either use case, the ML model wouldn't improve how an individual player performs, rather it will help optimize team builds and provide them with the best chance at victory, which in the ESports industry is worth millions of dollars.

Future Extension

One of the papers we reviewed analyzed several models for Dota 2 game outcome prediction, and the model that resulted in the highest test accuracy used an ensemble of **Genetic Algorithm and Logistic Regression**.[7] This modeling approach achieved a high 74.10% test accuracy, however, due to the small dataset being used, the model's reliability and consistency are questioned in the review.[7] The paper didn't mention if the Genetic Algorithm and Logistic Regression modeling approach has been done on a bigger dataset, so this could be a future modeling approach that our team uses for future extensions of our data analysis on Dota 2 game prediction.[7]

Another extension for future extension can be to overcome the limitations faced in this report, primarily the construction of a larger version of dataset 1. Given more time and computing resources, a dataset with the features of dataset 1 and dataset 2 can be curated with hundreds of thousands of instances. The Dota2 API endpoints utilized for this report provides access to that data. The combined dataset would give the classification models the advantage of both meaningful features and a large number of instances.

References

[1] D. Berrar, P. Lopes, and W. Dubitzky, “Incorporating domain knowledge in machine learning for soccer outcome prediction,” Machine Learning, vol. 108, no. 1, pp. 97–126, 2018.

[2] M. Goswami, “Dream11 team predictor with python and machine learning,” Medium, 22-Jan-2021. [Online]. Available: https://medium.com/analytics-vidhya/dream11-team-predictor-with-python-and-machine-learning-f0dfce1489eb. [Accessed: 29-Sep-2021].

[3] A. Sapienza, P. Goyal, and E. Ferrara, “Deep neural networks for optimal team composition,” Frontiers in Big Data, vol. 2, 2019.

[4] B. Yang, “Predicting e-sports winners with machine learning,” *Medium*, 17-Dec-2019. [Online]. Available: https://blog.insightdatascience.com/hero2vec-d42d6838c941. [Accessed: 29-Sep-2021].

[5] R. Wolff, “5 types of classification algorithms in machine learning,” MonkeyLearn Blog, 26-Aug-2020. [Online]. Available: https://monkeylearn.com/blog/classification-algorithms/. [Accessed: 29-Sep-2021].

[6] A. Ohri, “8 popular regression algorithms in machine learning of 2021,” *Jigsaw Academy*, 05-Apr-2021. [Online]. Available: https://www.jigsawacademy.com/popular-regression-algorithms-ml/. [Accessed: 29-Sep-2021].

[7] A. Semenov, P. Romov, K. Neklyudov, D. Yashkov, and D. Kireev, “Application of Machine Learning in Dota 2: Literature Review and Practical Knowledge Sharing,” [Online]. Available: http://ceur-ws.org/Vol-1842/paper_05.pdf. [Accessed: 24-Nov-2021]

[8] Willingham, AJ. “What Is Esports? A Look at an Explosive Billion-Dollar Industry.” *CNN*, Cable News Network, 27 Aug. 2018, https://www.cnn.com/2018/08/27/us/esports-what-is-video-game-professional-league-madden-trnd/index.html.

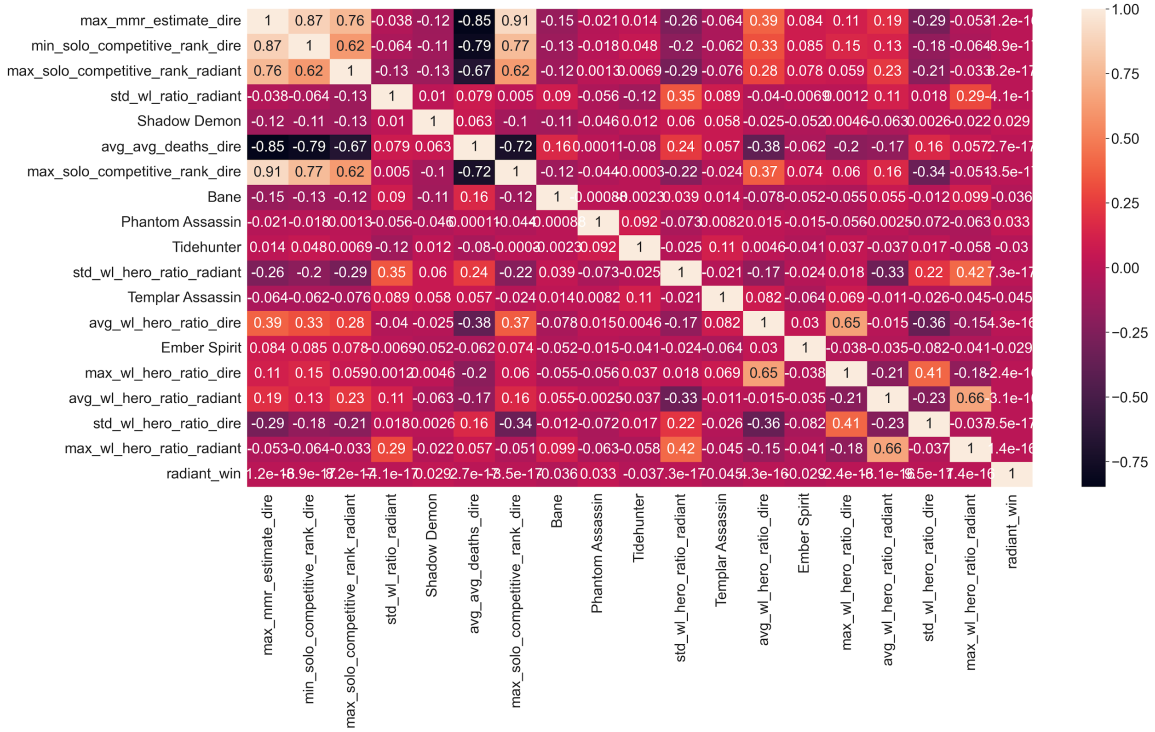
[9] Savov, Vlad. “Is Dota 2 a Game or an Industry?” Bloomberg.com, Bloomberg, 27 Aug. 2021, https://www.bloomberg.com/news/newsletters/2021-08-27/world-s-most-lucrative-esports-event-dota-2-returns-to-live-play.

[10] Brownlee, Jason. “How to Choose a Feature Selection Method for Machine Learning.” How to Choose a Feature Selection Method For Machine Learning, Machine Learning Mastery, 20 Aug. 2020, https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/.

Appendix

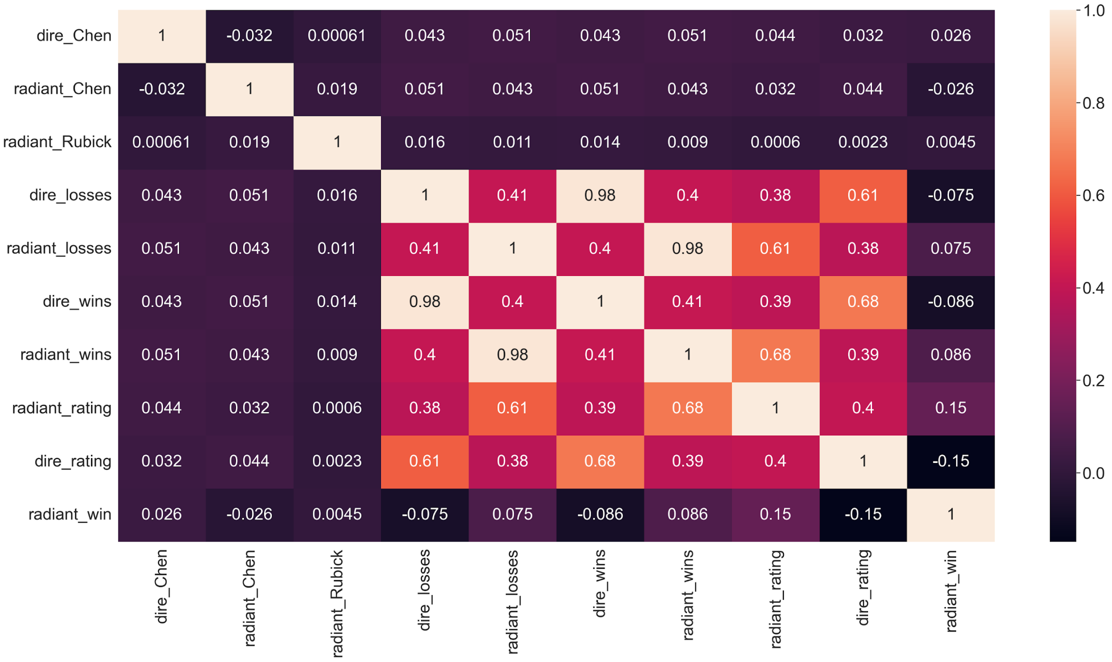
Appendix A - Pearson Correlation

Dataset 1



	max_mmr_estimate_dire	min_solo_competitive_rank_dire	max_solo_competitive_rank_radiant	std_wl_ratio_radiant	Shadow Demon	avg_avg_deaths_dire	max_solo_competitive_rank_dire	Bane	Phantom Assassin	Tidehunter	std_wl_hero_ratio_radiant	Templar Assassin	avg_wl_hero_ratio_dire	Ember Spirit	max_wl_hero_ratio_dire	avg_wl_hero_ratio_radiant	std_wl_hero_ratio_dire	max_wl_hero_ratio_radiant	radiant_win
max_mmr_estimate_dire	1	0.87	0.76	-0.038	-0.12	-0.85	0.91	-0.15	-0.021	0.014	-0.26	-0.064	0.39	0.084	0.11	0.19	-0.29	-0.053	-1.2e-16
min_solo_competitive_rank_dire	0.87	1	0.62	-0.064	-0.11	-0.79	0.77	-0.13	-0.018	0.048	-0.2	-0.062	0.33	0.085	0.15	0.13	-0.18	-0.064	-8.9e-17
max_solo_competitive_rank_radiant	0.76	0.62	1	-0.13	-0.13	-0.67	0.62	-0.12	0.0013	0.0069	-0.29	-0.076	0.28	0.078	0.059	0.23	-0.21	-0.033	8.2e-17
std_wl_ratio_radiant	-0.038	-0.064	-0.13	1	0.01	0.079	0.005	0.09	-0.056	-0.12	0.35	0.089	-0.04	-0.0069	0.0012	0.11	0.018	0.29	-4.1e-17
Shadow Demon	-0.12	-0.11	-0.13	0.01	1	0.063	-0.1	-0.11	-0.046	0.012	0.06	0.058	-0.025	-0.052	0.0046	-0.063	0.0026	-0.022	0.029
avg_avg_deaths_dire	-0.85	-0.79	-0.67	0.079	0.063	1	-0.72	0.16	0.00011	-0.08	0.24	0.057	-0.38	-0.062	-0.2	-0.17	0.16	0.057	2.7e-17
max_solo_competitive_rank_dire	0.91	0.77	0.62	0.005	-0.1	-0.72	1	-0.12	-0.044	-0.0003	-0.22	-0.024	0.37	0.074	0.06	0.16	-0.34	-0.051	-3.5e-17
Bane	-0.15	-0.13	-0.12	0.09	-0.11	0.16	-0.12	1	-0.00088	-0.0023	0.039	0.014	-0.078	-0.052	-0.055	0.055	-0.012	0.099	-0.036
Phantom Assassin	-0.021	-0.018	0.0013	-0.056	-0.046	0.00011	-0.044	-0.00088	1	0.092	-0.073	0.0082	0.015	-0.015	-0.056	-0.0025	-0.072	-0.063	0.033
Tidehunter	0.014	0.048	0.0069	-0.12	0.012	-0.08	-0.0003	-0.0023	0.092	1	-0.025	0.11	0.0046	-0.041	0.037	-0.037	0.017	-0.058	-0.03
std_wl_hero_ratio_radiant	-0.26	-0.2	-0.29	0.35	0.06	0.24	-0.22	0.039	-0.073	-0.025	1	-0.021	-0.17	-0.024	0.018	-0.33	0.22	0.42	7.3e-17
Templar Assassin	-0.064	-0.062	-0.076	0.089	0.058	0.057	-0.024	0.014	0.0082	0.11	-0.021	1	0.082	-0.064	0.069	-0.011	-0.026	-0.045	-0.045
avg_wl_hero_ratio_dire	0.39	0.33	0.28	-0.04	-0.025	-0.38	0.37	-0.078	0.015	0.0046	-0.17	0.082	1	0.03	0.65	-0.015	-0.36	-0.15	4.3e-16
Ember Spirit	0.084	0.085	0.078	-0.0069	-0.052	-0.062	0.074	-0.052	-0.015	-0.041	-0.024	-0.064	0.03	1	-0.038	-0.035	-0.082	-0.041	-0.029
max_wl_hero_ratio_dire	0.11	0.15	0.059	0.0012	0.0046	-0.2	0.06	-0.055	-0.056	0.037	0.018	0.069	0.65	-0.038	1	-0.21	0.41	-0.18	-2.4e-16
avg_wl_hero_ratio_radiant	0.19	0.13	0.23	0.11	-0.063	-0.17	0.16	0.055	-0.0025	-0.037	-0.33	-0.011	-0.015	-0.035	-0.21	1	-0.23	0.66	-3.1e-16
std_wl_hero_ratio_dire	-0.29	-0.18	-0.21	0.018	0.0026	0.16	-0.34	-0.012	-0.072	0.017	0.22	-0.026	-0.36	-0.082	0.41	-0.23	1	-0.037	9.5e-17
max_wl_hero_ratio_radiant	-0.053	-0.064	-0.033	0.29	-0.022	0.057	-0.051	0.099	-0.063	-0.058	0.42	-0.045	-0.15	-0.041	-0.18	0.66	-0.037	1	1.4e-16
radiant_win	-1.2e-16	-8.9e-17	8.2e-17	-4.1e-17	0.029	2.7e-17	-3.5e-17	-0.036	0.033	-0.03	7.3e-17	-0.045	4.3e-16	-0.029	-2.4e-16	-3.1e-16	9.5e-17	1.4e-16	1

Dataset 2



	dire_Chen	radiant_Chen	radiant_Rubick	dire_losses	radiant_losses	dire_wins	radiant_wins	radiant_rating	dire_rating	radiant_win
dire_Chen	1	-0.032	0.00061	0.043	0.051	0.043	0.051	0.044	0.032	0.026
radiant_Chen	-0.032	1	0.019	0.051	0.043	0.051	0.043	0.032	0.044	-0.026
radiant_Rubick	0.00061	0.019	1	0.016	0.011	0.014	0.009	0.0006	0.0023	0.0045
dire_losses	0.043	0.051	0.016	1	0.41	0.98	0.4	0.38	0.61	-0.075
radiant_losses	0.051	0.043	0.011	0.41	1	0.4	0.98	0.61	0.38	0.075
dire_wins	0.043	0.051	0.014	0.98	0.4	1	0.41	0.39	0.68	-0.086
radiant_wins	0.051	0.043	0.009	0.4	0.98	0.41	1	0.68	0.39	0.086
radiant_rating	0.044	0.032	0.0006	0.38	0.61	0.39	0.68	1	0.4	0.15
dire_rating	0.032	0.044	0.0023	0.61	0.38	0.68	0.39	0.4	1	-0.15
radiant_win	0.026	-0.026	0.0045	-0.075	0.075	-0.086	0.086	0.15	-0.15	1

Appendix B - Top 5 important features for each classification model

Dataset 1

Logistic Regression	Decision Trees	Random Forest	Naive Bayes
min_solo_competitive_rank_dire	max_wl_hero_ratio_radiant	std_wl_hero_ratio_radiant	Templar Assassin
min_solo_competitive_rank_radiant	std_wl_hero_ratio_dire	avg_wl_hero_ratio_dire	Phantom Assassin
avg_avg_deaths_dire	avg_wl_hero_ratio_radiant	avg_wl_hero_ratio_radiant	Bane
max_solo_competitive_rank_dire	max_wl_hero_ratio_dire	std_wl_hero_ratio_dire	Shadow Demon
max_mmr_estimate_dire	Ember Spirit	Tidehunter	std_wl_ratio_radiant

Dataset 2

Logistic Regression	Decision Trees	Random Forest	Naive Bayes
min_solo_competitive_rank_dire	max_wl_hero_ratio_radiant	std_wl_hero_ratio_radiant	Templar Assassin
min_solo_competitive_rank_radiant	std_wl_hero_ratio_dire	avg_wl_hero_ratio_dire	Phantom Assassin
avg_avg_deaths_dire	avg_wl_hero_ratio_radiant	avg_wl_hero_ratio_radiant	Bane
max_solo_competitive_rank_dire	max_wl_hero_ratio_dire	std_wl_hero_ratio_dire	Shadow Demon
max_mmr_estimate_dire	Ember Spirit	Tidehunter	std_wl_ratio_radiant

Appendix C - Feature selection impact on classification model scores:

Dataset 1	Logistic Regression	Decision Trees	Random Forest	Naive Bayes
Original Scores	0.261907	0.500298	0.105753	0.335592
Logistic Regression	0.385068	0.152192	0.123939	0.410919
Decision Trees	0.460251	0.386196	0.148348	0.531221
Random Forest	0.450996	0.408449	0.223409	0.472022
Naive Bayes	0.443235	0.344899	0.231753	0.354787
Ridge Regression	0.459023	0.452148	0.304360	0.444454
Pearson Correlation	0.463529	0.590448	0.602843	0.334770

Dataset 2	Logistic Regression	Decision Trees	Random Forest	Naive Bayes
Original Scores	0.635751	0.604735	0.664609	0.378651
Logistic Regression	0.639889	0.604794	0.653685	0.585079
Decision Trees	0.645506	0.603890	0.654608	0.592208
Random Forest	0.645920	0.604391	0.653670	0.600085
Naive Bayes	0.640865	0.606035	0.651480	0.592266
Ridge Regression	0.639889	0.605080	0.654284	0.585079
Pearson Correlation	0.640267	0.605577	0.652830	0.584402

Muhammad Ali Qadri	Neural Networks, and feature exploration	PCA analysis and Feature analysis
Member 2 name	Data creation (from API), preprocessing, and exploration	Feature Importance, Feature Selection
Xiaolu Dedman	Models: Decision Tree, Random Forest, and Logistic Regression	Feature Distribution