**Dota2 Match Outcomes Prediction**

**Based on Text Mining**

Zimeng Zhang

School of Information Studies

Syracuse University

Syracuse, NY, 13210

zzhang@syr.edu

**Abstract**

This project collected nearly 30,000 text chat of DOTA2 games. Apply LDA topic model to calculate the most frequently discussed topics in the game. Conduct trending analysis in a game and observe the trend of the topic changes. Utilize N- gram method to vectorize original text data. And establish classification models to predict the outcome of the game via SVM and MNB. Evaluate the performance of the model for a balanced dataset with accuracy, which was 68%, 18% better than random guess. Plot ROC curve and calculate AUC score to evaluate the performance of the model for the imbalanced data set. Finally, it is found that the performance of each classification model is similar. Analyze the informative features to interpret each model. As a result, features of the SVM model are more convincing.

1. **Introduction**

DotA2 is a multiplayer online battlefield (MOBA) video game developed and published by Valve. This video game was so popular that it developed into its own game, which inspired the entire MOBA video game genre, and continues to flourish today. Dota2 has a huge player base. Every day, as one of more than one hundred Dota2 heroes in 5v5 team competitions, millions of players around the world enter the battle. Dota2 is the deepest multiplayer action RTS (real-time strategy) game ever, and there are always new strategies to discover.

In each game, two teams of five players compete against each other to defeat the enemy and destroy the base. This is a multifaceted game that combines and requires players to have strategy (macro), mechanics (micro), leadership, communication, and other skills. Every detail of the game is crucial to the outcome of this match, starting from hero selection, item priority, skill, talent, strategy, and teamwork. Winning the game will increase the player's MMR, while losing the game will reduce the player's MMR. The prestige in DotA 2 depends almost entirely on the player's MMR, and therefore largely depends on the player's ability to win. This is no different from many other video games and sports.

Many studies and research projects have been conducted to try to predict the outcome of the game based on statistical data, which is not a strange concept. However, these projects have been rooted in a basic fact. That is, in any game, there is almost always a trend between the numbers (or at least some of them) and the outcome of the game. This is guaranteed. For example, there is no statutory law stating that the team with the most rebounds or rebounds will win the game, but it is almost certain that the team with the most numbers will usually win, and it does.

Statistics can predict the team’s victory without planning obviously, such as gold, death toll, death toll, etc. They can foresee victory. Except for the things that determine the outcome of the DotA 2 match, chatting in an RTS game is also a crucial aspect of victory. In people's traditional concept, correct, good, and positive communication leads to a relaxed and focused competitive state, which makes it easier to lead the team to victory. Relatively, putting pressure on teammates will cause the team atmosphere to become tense. In this environment, people are more likely to lose confidence due to mistakes, which leads to loss of the game. But in a tense game, players do not have time for in-depth communication. Instead of game strategies and encouragement, people prefer to use simple expressions or slang words or chat roulettes that have been pre-stored in the game to communicate. So, the chat is obviously a wildcard, but you can still predict the chat from the game. As players, we know that if we make a big mistake, we can expect the "report team" in the chat, or if we are fascinated, we can expect "gg ez" to appear eventually. Therefore, if we think that we can predict the chat based on the state or outcome of the game, can we predict the outcome of the game based on the chat? What topics players usually talk about when playing games, whether these topics can help to win the game, and whether they can predict the win or loss of the game through the chat records of a game, is the center of this article.

1. **Method**
   1. **Topic Modeling**

Topic modeling is a statistical modeling used to discover abstract "topics" that appear in document collections. Latent Dirichlet mapping (LDA) is an example of a topic model that is used to classify document text on a specific topic. A topic model is constructed for each document, and words are constructed for each topic model, which is modeled as a Dirichlet distribution. In natural language processing, LDA is a generative probability model that allows unobserved groups to interpret a set of observations that explain why certain parts of the data are similar. It is an algorithm that can help analyze the potential topic representations of a given corpus or data set. It assumes that each document is a combination of a small number of topics, and the creation of each word can be attributed to a topic in the document.

In order to realize the topics discussed in the DOTA2 competition, this article uses LDA for topic discovery, and annotates each document and its topic category for further trend analysis.

* 1. **Classification**
     1. **Multinomial Naïve Bayes**

Naive Bayes is based on Bayes' theorem. The adjective Naïve says that the features of the data set are independent of each other. The appearance of one characteristic does not affect the possibility of the appearance of another characteristic. For smaller sample sizes, Naïve Bayes can outperform more powerful options. It is relatively robust, easy to implement, fast and accurate, and can be used in many different fields. Multinomial Naive Bayes considers feature vectors, where a given term represents the number of occurrences or very frequent, i.e., frequency. It is a classification model based on probability theory. Given the value of the variable in the training data, MNB can calculate the probability of the variable value in the test data and classify a certain document to a class based on the max probability.

* + 1. **Support Vector Machine**

Support vector machine (SVM) is a method used for data classification and regression analysis. Here, we use SVM to learn from the training data, match outcome and chat, and then build a model to predict the match outcomes based on chat data. SVM is a hyperplane classifier, which works by determining which side of the hyperplane is on. SVM maximizes the margin around the separating hyperplane. The decision-making function is completely specified by a subset of the training samples. This subset of the vector is called the support vector.

Compared with the MNB method, the SVM training algorithm is a binary linear classifier with no probability. This method divides the data into categories separated by hyper lanes, and the marginal width of the hyper lanes becomes as wide as possible. The new data will be allocated to the classified space according to which side of the space it is located. Both MNB and SVM are used to build classification models.

* 1. **Data Set**

Original data set can be obtained from OPENDOTA. <https://www.opendota.com/>.

There is also match chat data available on Kaggle. <https://www.kaggle.com/devinanzelmo/dota-2-matches> In this report, I collected around 30,000 matches data from source, including chat text, chat time, match outcomes, and numeric statistic for a game. And split the chat data of each match into 2 teams as winning team and losing team. Then label the match outcomes 1 as win, 0 as lose. So, the data will be perfectly balanced with half winning and half losing. Here, we will furtherly make an explanation of the data storage method. In the original data set, players’ chat text and match attributes are stored separately in 2 tables. In the match table, each row demonstrates attributes of a certain match. For instance, we can extract match outcome, duration, and region. In the chat table, each row demonstrates that chat content of a player at a certain time point of a specified game.

* 1. **Experimental Procedures**
     1. **Data Preprocessing**

For the classification problem, the first thing is to join different data table in order that each record in data frame is stored in the pattern like (chat, …, match outcomes) which fits classification model. We load all data into pandas data frame and operate on it to select the part we need. For each game, the speeches of each player from different teams are concatenated together to form chat text data for that team, and then re-identify team id. For each row in the match outcome table, we split it into 2 outcomes for each team and re-identify team id. Then join above 2 resulting tables on team id to extract text, region, and match outcome. After which, we have already created for prediction problem.

We create other text data frames for topic modelling analysis problems. Duration of a regular is supposed to be 40 minutes to 50 minutes, so we select matches in that duration. We extract chat data from U.S. servers and concatenate them together for topic modelling. As a result, we get well-prepared data for topic modelling.

* + 1. **Vectorization Selection**

People often use short expressions in an intense game. In theory, converting single words into N-grams will be more suitable for the model. So, N-grams method is used to detect if the model would have a better performance by taking word combinations as input instead of taking a single word as input. We try to perform different gram conversions on training data to build a regression model and determine which vectorization to use based on its prediction accuracy. Table 1 demonstrates that holding all other parameters unchanged, trigram transformation has the best performance for each classification model. For the vectorization we utilize trigram transform and set minimum term frequency equals to 5 to avoid very rare spelling errors and remove stops words

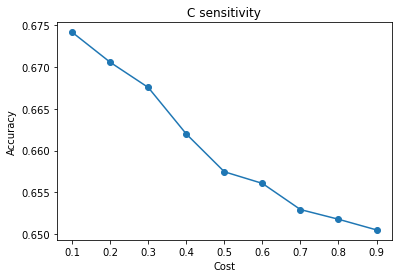
**Table 1** Accuracy of N-grams

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | | |
| Vectorization | Unigram | Bigram | Trigram |
| MNB | 0.6659 | 0.6696 | 0.6732 |
| SVM | 0.6483 | 0.6564 | 0.6572 |

* + 1. **SVM hyperparameter tuning**

SVM model has a crucial hyperparameter called C, which would have an influence on model performance significantly. To achieve the best performance, we run several SVM models with different C in range 0.1 to 1and consider accuracy as the evaluation criteria. Select C that the model has the best accuracy score. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger margin separating hyperplane, even if that hyperplane misclassified more points. For very tiny values of C, you should get misclassified examples, often even if your training data is linearly separable. Fig demonstrates the accuracy of SVM vs. cost. To optimize the model, we set c = 0.1.

**Figure 1** SVM cost sensitivity



* + 1. **Trending analysis**

To see how the topics change through the whole game, we conduct trending analysis. Call the LDA algorithm to fit a topic model, and transform all documents to their topic distributions, set a threshold for document’s topic probability. If the probability of a certain document is greater than threshold, annotate each document to their topic distribution. Chat time is already in our data. Then conduct trend analysis by plotting topic numbers vs. timeline.

1. **Results**
   1. **Topics trending**

Table 2 demonstrates top representative words for each topic, we can summarize the topics as 4 categories, which are respectively taunt, interactions in the game, comments of the entire game and unknown topic.

**Table 2** Topics distribution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Topic: | **Taunt** |  |  |  |
| ez mid, xd, ez, mid, haha, lmao, wtf, mmr, like, game | | | | |
| Topic: | **Interactions in the game** | | |  |
| lol, game, nice, team, just end, good thing, ty | | | | |
| Topic: | **Comments of entire game** | | |  |
| gg, report, ggwp, noob, commend pls | | | |  |
| Topic: | **Unknown Topic** | |  |  |
| que, es, la, el, xd, jaja, usa, peru, mierda, se | | | | |

Topic 0, taunt, is a battle cry, sarcastic remark, or insult intended to demoralize the recipient, or to anger them and encourage reactionary behaviors without thinking. If a player is over performing than enemies, he probably will say some slang words to mentally attack my enemies, which is also a communication strategy. Topic 1 interactions in game, players often discuss and praise the actions that take place in the game. Players often discuss and praise the actions that take place in the game. Interactions in game are closely related to the game situation. Obviously, the dominant player’s words are more confident. When the advantage is gradually lost and the situation begins to be intense, the player, people may lose confidence and say, “just end”. Topics 2 comments of the entire game. This topic is more like a tradition at the end of each game, for instance, commend good players, report bad ones. And often when it comes to the end of the game.

**Figure 2** Trending analysis

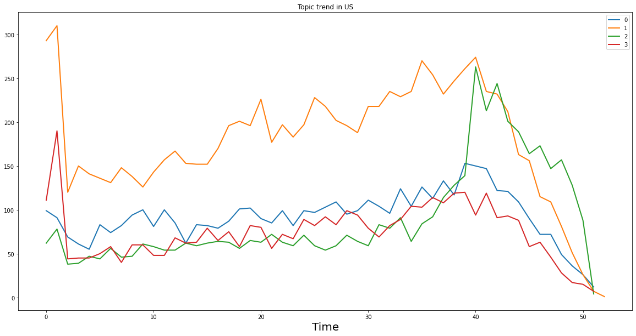


Figure 2 demonstrates that topics change over the timeline. Every topic gradually increases as the game is going. At the beginning of the game, that is during drafting time, players are selecting heroes and discussing strategy. Taunt and interactions are the main topic, as the game starts, all topics decrease sharply means that players must focus on the game. And we can see from the timeline that topic1: interactions in game are the most frequently discussed topic through the whole game, while when coming to the end of the game, comments of the entire game increases sharply from the least frequently discussed topic to the most frequently discussed topic.

* 1. **Classification Report**
     1. **Multinomial Naïve Bayes**

We compiled a classification report to observe the classification results. For this binary classification problems, the classification ability of the MNB classifier for the two categories is different, which can be reflected by the difference between recall and precision score. The failure category has a higher precision, and the success category has a higher recall. The average accuracy of the classifier is 67%, which is 17% higher than random guess.

**Table 3** MNB classification report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MNB | precision | recall | f1-score | support |
| defeat | 0.73 | 0.54 | 0.62 | 4274 |
| victory | 0.64 | 0.80 | 0.71 | 4353 |
| accuracy |  |  | 0.67 | 8627 |
| macro avg | 0.68 | 0.67 | 0.67 | 8627 |
| weighted avg | 0.68 | 0.67 | 0.67 | 8627 |

In order to better explain the model, we extracted the top 5 informative words for each class from the model. When these words appear in a paragraph of text, they have a greater impact on the model's classification judgment. The following table shows the top five informative features for each class in the MNB model.

**Table 4** MNB Informative features

|  |  |
| --- | --- |
| Defeat | Victory |
| 100 MMR | report |
| 1 vs 5 | mid |
| acc buyer | gg gg |
| 4v6 | game |
| 22 minutes | gg wp |

According to table 4, The most informative features for defeat class makes more sense. Because when people are losing, they are more likely to blame MMR, blaming the skills of team mates even being suspicious that “You are playing so bad, if the account really belongs to you?” and blaming that they lost the game in such a short time. However, features for victory class contain ‘report’ which mostly appear in team losing.

* + 1. **Support Vector Machine**

After tuning the hyperparameter of the SVM model, we set c=0.1 for the best model performance accuracy. We compiled a classification report to observe the classification results. For these binary classification problems, the classification ability of the SVM classifier for the two categories is almost the same. The average accuracy of the classifier is 68%, which is 18% higher than random guess.

**Table 5** MNB classification report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SVM | precision | recall | f1-score | support |
| defeat | 0.69 | 0.66 | 0.68 | 4274 |
| victory | 0.68 | 0.70 | 0.69 | 4353 |
| accuracy |  |  | 0.68 | 8627 |
| macro avg | 0.68 | 0.67 | 0.68 | 8627 |
| weighted avg | 0.68 | 0.67 | 0.68 | 8627 |

In order to better explain the model, we extracted the top 5 informative words for each class from the model. When these words appear in a paragraph of text, they have a greater impact on the model's classification judgment. The following table shows the top five informative features for each class in SVM model.

Compared to the word given by MNB, features of SVM make more sense. In the victory category, each short sentence can show the advantage the player has achieved. And the words for the defeat category reflects the player’s loss of confidence in the game.

**Table 6** MNB classification report

|  |  |
| --- | --- |
| Victory | Defeat |
| commend | end |
| ez | report |
| let end | just end |
| recommend | mid |
| voltis | finish |

1. **Discussion**

From the former section we know that SVM has a better average accuracy dealing with balanced data sets. While MNB has a better f1-score for predicting victory class. Does this mean that MNB has a better ability predicting victory and SVM has better ability predicting victory?

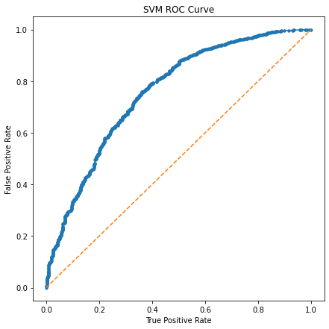
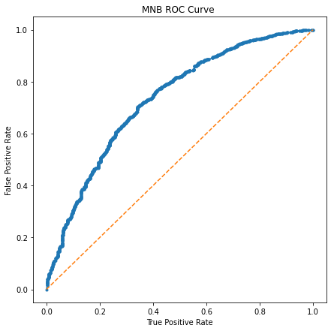
To answer this question, I sample from original data to create an imbalanced data set with 90% victory and 10% defeat to see if there is an obvious difference in model performance.

When our goals are unbalanced, for example, we are more likely to look at the classifier’s ability to correctly predict the victory of the game. Precision or recall would be a better evaluation.

Here we use true positive rate and false positive rate to plot the receiver operating characteristic curve (ROC curve). ROC curve is a graph showing the performance of a classification model at all classification thresholds. The AUC score shows the 2 classifiers have almost the same overall performance.

MNB:0.7320, SVM: 0.7571

**Figure 3** ROC curve for MNB (left), SVM (right)



1. **Conclusion and Limitation**

Through the work done in this article, we have learned about the topics that people talk about in a game, and the topics change as the game progresses. For example, people always talk a lot at the beginning of the game and when the game officially starts, Players will be more inclined to focus on the game. As the game progresses, there will be more and more interactions between players.

We have obtained a relatively good model for predicting the outcome of the game, up to 18% higher than random guessing, the accuracy of the two models is not much different, and the informative feature of SVM is more convincing, and both predict the capabilities of imbalanced data sets are similar.

Finally, I must think about the causal relationship between game winning and the player's speech, whether it is because the speech led to the victory or because of some specific game processes such as gradually establishing an advantage, expanding the advantage, and turning it into a victory. In this process, the player will make such a statement when the winning ticket is held.

**Reference**

K. Wang and W. Shang, "Outcome prediction of DOTA2 based on Naïve Bayes classifier," 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), 2017, pp. 591-593, doi: 10.1109/ICIS.2017.7960061.

M. Anshori, F. Mar’i, M. W. Alauddin and F. Abdurrahman Bachtiar, "Prediction Result of Dota 2 Games Using Improved SVM Classifier Based on Particle Swarm Optimization," 2018 International Conference on Sustainable Information Engineering and Technology (SIET), 2018, pp. 121-126, doi: 10.1109/SIET.2018.8693204.