Predicting Counter-Strike Round Winners

Riley Gavigan
Department of Computer Science
University of Western Ontario
London, Ontario
251150776
rgavigan@uwo.ca

Benjamin Matthews
Department of Computer Science
University of Western Ontario
London, Ontario
251180881
bmatthe8@uwo.ca

Mathew Young
Department of Computer Science
University of Western Ontario
London, Ontario
251236525
myoun54@uwo.ca

Bruce Scott
Department of Computer Science
University of Western Ontario
London, Ontario
251241869
bscott97@uwo.ca

***Abstract*—This study works to accurately predict round winners in Counter-Strike (CS:GO) competitive matches. We identified the strongest classification algorithm for predicting Counter-Strike round winners from a selection of supervised learning algorithms. The study focuses on comparing both map-agnostic and map-specific results for six different classification algorithms on our provided dataset [7] with 122,411 snapshots taken every 20 seconds during live CS:GO rounds. Our dataset consists of approximately 700 demos from tournament matches in 2019 and 2020, curated to exclude warm-up rounds and restarts.**

Our methodology includes data processing methods to identify significant features and the target of our data; implementation specifics for each classification algorithm; and statistical analysis of the results for each algorithm through accuracy and confusion matrices. By comparing the accuracy between models, we determined that the Decision Tree algorithm worked best for our dataset, achieving an 83.7% accuracy when trained on map-specific data. We concluded that map-based accuracy differs significantly between maps and that we were able to implement a successful classification algorithm for predicting round winners. Our study leads to further implications in sports betting by generalizing these implementations to other sports. Furthermore, model improvement is noted as future work as more work can be done to find significant variables and emphasize variables that are important. This study outlined the importance of algorithm choice in machine learning tasks based on the data being provided.

***Index Terms*—Supervised Learning, Counter-Strike, Classification, Logistic Regression, Naive Bayes, KNN, SVM, Neural Network, Decision Tree**

I. Introduction

Counter-Strike: Global Offensive (CS:GO) is a competitive online tactical first-person shooter video-game. CS:GO is a pioneer in the e-sports industry, with yearly tournaments that garner millions of views being held, similar to other sports.

One significant problem in CS:GO and in sports betting is predicting the outcome of competitive competitions and matches. The unpredictable nature of CS:GO and in all sports is what drives the sports betting industry, with no algorithm being able to predict match results with certainty. This leads to the main problem of our study: predicting CS:GO round winners with a high level of accuracy. In CS:GO, a competitive match contains up to 30 rounds, and each round has a significant impact on the result of an entire match.

Our study investigates a dataset with over 122,000 snapshots taken place during rounds with features describing the conditions at random points in time. This dataset serves as a foundation for our investigation: comparing the performance of different supervised learning classification algorithms on the dataset to determine the strongest algorithm for the task.

We determine in this study that the Decision Tree algorithm is best at predicting round winners, both with a general model that is map-agnostic, and with models that are trained and tested on data for a single map. We were able to create a model that predicts the winner of a round accurately over 83% of the time, significantly outperforming random guessing.

This study demonstrates the importance of identifying important variables, such as maps in our instance, in training and testing models. Furthermore, it displays that the conditions of a round heavily determine the outcome of that round in a predictive manner to a significant degree. The study leads to further research opportunities to enhance the algorithms, and is highly extensible to applications in any other e-sport or sport with similar datasets.

The report is organized as follows: Section 2 provides an overview of related work in CS:GO supervised learning tasks alongside current state-of-the-art supervised learning approaches to classification. Section 3 enumerates the research objectives. Section 4 outlines the materials and methodology used to predict CS:GO round winners, such as data cleaning and models utilized. Section 5 displays the results of our implementation and analyzes its performance. Finally, Section 6 concludes the study and discusses future work to improve implementation.

II. Background & Related Work

A. Classification Algorithms

There are several supervised learning classification algorithms used today. These include: Logistic Regression, Naive Bayes, K-Nearest Neighbours, Decision Trees, Support Vector Machines, and Neural Networks [8]. Python implementations of these classification algorithms are available through the scikit-learn library [9].

Each classification algorithm has unique pros and cons, leading to their power being dependent on the data available and the problem being solved.

1) Logistic Regression: Previous studies analyzing Logistic Regression in medical research has revealed a clear idea about what the method excels at and where it falls short. The regression technique has many benefits and succeeds at measuring associations, controlling confounded variables, and predicting outcomes. It can efficiently examine multiple independent variables on a binary outcome and the impact these variables have on the output. This can provide great insight into each variable and what it affects. Pitfalls of Logistic Regression must be analyzed to ensure proper use of the method [11]. Choosing the right predictor is essential to achieving a meaningful result. Picking too many variables will lead to large errors and broad confidence intervals. Therefore, Logistic Regression should be used with a minimal amount of variables to achieve the result. This technique thrives in classification with a binary outcome and a minimal amount of high-impact variables [10].

2) Naive Bayes: Naive Bayes has been analyzed logically with probability and known algorithms to determine its strengths and weaknesses in research. Naive Bayes succeeds in its efficiency. Training and classification can be completed with only one run through the data. Also, the method requires a small amount of training data to produce estimations about the classification. The performance of Naive Bayes drops when the features involved are highly correlated. The algorithm assumes that the features are not functionally correlated, a property that the real-world often contradicts, producing incorrect data. Therefore, Naive Bayes is best used with a small amount of data with features that are not correlated, calculating simple probabilities for classification [5].

3) K-Nearest Neighbours: The K-Nearest Neighbour (KNN) algorithm excels in its simplicity and effectiveness for classification problems. KNN takes in any data and groups similar data points together based on their proximity to each other. This simple calculation makes it easier to understand where values came from, providing trivial confirmation of the output and error detecting. The shortcoming of KNN is observed through its efficiency. KNN is inherently a lazy learner, removing many of its applications in the web mining field. Also, KNN requires a good value of 'k' to perform well. A wrong value could lead to big groups that show no meaningful correlations. The KNN algorithm is best used for simple classification problems requiring analysis of patterns and similar data points [2].

4) Decision Trees: The decision tree method is a popular one for machine learning because of its application in many fields. The primary advantage is that Decision Trees are easy to understand. The result is simple for others to comprehend and make conclusions about the data analyzed. There are 2 types of decision trees: classification and regression; allowing for different computations and uses depending on the data. The pitfall of the decision tree is that it can only be used over a single table and over a single attribute at a time. This reduces the speed and scalability of the method, making it weak for large, complex datasets. The decision tree excels at analyzing single sections of data and should be used in simple relationships for peak performance [6].

5) Neural Networks: Neural Networks are used in many machine learning studies due to their complexity and effectiveness. They learn by adjusting weights of their neurons to classify training data, working to predict unknown data. Neural Networks are very successful at taking in a complex system and simplifying it down into simple elements. The largest problem with Neural Networks is related to their greatest strength: complexity. With so many processors and calculations comes very complex implementation and an overall confusion of how predictions are calculated. Overall, Neural Networks perform very well and should be used with a large, complex dataset [6].

6) Support Vector Machines: Support Vector Machines (SVM) are commonly used in classification problems. The main strength of SVMs is the training required for successful prediction is relatively minimal. It scales very well to more complex data with minimal errors. SVMs have been used extensively with categorization of text documents and has been successful with great results [4]. The main pitfall of SVMs is that they require a good Kernel function to perform well. A standard Gaussian Kernel can be used; however if the inputs are non-continuous, a more complex kernel is required. Therefore, SVMs work best for classic classification problems and data as the SVM and its kernel can effectively be trained and produce results with great accuracy [4].

B. CS:GO Machine Learning Case Studies

Previous research in the field of CS:GO machine learning has showcased the growing interest in leveraging data-driven approaches to enhance gameplay understanding and performance prediction. Initiatives such as the study conducted by Peter Xenopoulos, Bruno Coelho, and Claudio Silva [12] have contributed significantly to this field, presenting great insights into optimal team economic decisions in Counter-Strike. However, despite these efforts, a significant research gap persists in accurately predicting round outcomes in competitive CS:GO matches. Existing studies often focus on broader aspects of gameplay analysis or personal player performance metrics, leaving a notable void in accurately predicting the specific round-by-round winner. Our study seeks to address this gap by specifically targeting the prediction of round winners, utilizing a large dataset and advanced classification algorithms to achieve high accuracy levels.

C. Classification For Competitive Sports

Prior research has suggested that larger datasets with player-level statistics, combined with more complex learning models, are the best way to improve the accuracy of predictions. Some success has been found using play-by-play data to train models, similar to how the data for CS:GO matches has been classified. Unfortunately, the vast majority of prior research on this topic surrounds the topic of physical sports rather than E-sports. However, there is a bridge that crosses the gap from the physical to virtual sports world with the idea of player-level stats and strategy. This is because the abstract idea of sports and E-sports are relatively the same [3].

III. Research Objectives

- O1: Implement a strong classification algorithm that accurately predicts CS:GO round outcomes.
- O2: Compare the effectiveness of different classification algorithms on predicting CS:GO round outcomes.
- O3: Identify relevant information for predicting CS:GO round outcomes with support from intuition about data.
- O4: Establish a correlation between algorithmic efficiency and the pros and cons of the classification algorithm.

IV. Research Methodology

The primary research question this study addressed is: are we able to accurately predict the round winner of a CS:GO match using classification? We hypothesize that we will be able to implement a classification algorithm that does this with greater accuracy than random guessing. Our methodologies include data processing prior to implementation of six different classification algorithms: Logistic Regression, Naive Bayes, K-Nearest Neighbours, Decision Trees, Support Vector Machines, and Neural Networks [9]. We compare models directly based on their accuracy and confusion matrices on the test set [9].

A. Data Processing

Our dataset [7] contains 96 features and 1 target column. We transformed the target feature "round_winner" into a binary target called "ct_win", which is directly mapped from "round_winner" by setting "ct" to 1 and "t" to 0. All feature variables are numerical except for "map", which we remove from the dataset in our original model comparisons. The "map" variable is later used to divide our data into specific train and test sets for each map to train map-based models.

The dataset was created by taking snapshots from around 700 demos of high-level tournament matches in CS:GO throughout 2019 and 2020 [7]. There are a total of 122,411 i.i.d. snapshots that have been pre-processed, filtered, and curated to exclude irrelevant snapshots such as those from warm-up rounds [7].

B. Study Implementation

Six classification models were used to implement our study. Our datasets were split 80%/20% for our train and test sets, respectively. All models were implemented using scikit-learn models [9]. First, we implemented all 6 models on the entirety of our dataset: Logistic Regression, Naive Bayes, K-Nearest Neighbours, Decision Trees, Support Vector Machines, and Neural Networks. Afterwards, each model was then re-implemented to focus on predicting outcomes for each map rather than being map agnostic. [9]

Details about the six classification models:

- Logistic Regression: 1000 maximum iterations, l2 penalty
- Naive Bayes: Gaussian Naive Bayes, var_smoothing of 1e-09
- K-Nearest Neighbours: n-neighbours value of 5, uniform weights, auto algorithm decided by sklearn, 30 leaf size, Minkowski power parameter of 2
- Decision Trees: Gini impurity, best split choice, no maximum depth, minimum samples to split of 2, and minimum samples to be a leaf of 1
- Support Vector Machines: Regularization parameter of 1, rbf kernel, degree of 3, $1/(n\_features * X.var())$ gamma value
- Neural Networks: multi-layer perceptron classifier, lbfgs solver, alpha value of 0.00001, and hidden layer sizes of 5 and 2

C. Result Analysis

In our study, the effectiveness of various classification algorithms is assessed primarily through two critical metrics: model accuracy and confusion matrices. These tools allow us to quantitatively analyze and compare the performance of each algorithm in predicting the winners of Counter-Strike rounds. By leveraging these two analytical methods, we thoroughly evaluated the predictive power of each algorithm. The accuracy gave us a direct measure of overall performance, while the confusion matrices provided deeper insights into the nature of the predictions and the potential biases in the models. Together, these tools form the cornerstone of our result analysis, ensuring a comprehensive understanding of each algorithm's effectiveness in predicting CS:GO round winners. In addition to the primary metrics of model accuracy and confusion matrices, the differential performance across various models sheds light on the unique characteristics and suitability of each algorithm for round winner prediction. For instance, the superior performance of the Decision Tree model (Table I) suggests its ability to capture complex patterns in the CS:GO data. In contrast, the lower accuracy of models like Support Vector Machines and Neural Networks as seen in (Table I) could point to challenges such as overfitting or underfitting, or perhaps the need for more extensive parameter tuning to adapt more accurately to the dataset.

D. Threats to Validity

Applicability of results to CS:GO matches that are not high-level is difficult to describe, since all snapshots gathered in our dataset are taken directly from high-level matches. For greater generalizability, retrieving snapshots from low-level and mid-level games could support results. However, it is likely that adding these data points would reduce model accuracy.

Furthermore, snapshots were collected for CS:GO in 2019 and 2020, which may be less applicable in present-day high-level matches. As time progresses, player skill level and skill ceilings tend to increase. By gathering identical snapshot data from more recent high-level tournament matches, we could determine how similar predictions are over 4-5 years. The first tournament for CS:GO's successor, Counter-Strike 2, took place in Copenhagen during March 2024, making present-day comparison possible.

V. Results

Selection of the most accurate classification algorithm for predicting CS:GO round winners was done through comparing the accuracy and confusion matrices for the six algorithms considered. Accuracy results from these comparisons are shown in Table I. Results were gathered by training and testing the model against all data points available. Moreover, results were not separated based on the CS:GO map they were collected for.

TABLE I
ACCURACY OF EACH MODEL ON ALL MAPS

Model	Accuracy
Decision Tree	0.821
K-Nearest Neighbours	0.762
Logistic Regression	0.743
Naive Bayes	0.728
Neural Network	0.724
Support Vector Machine	0.512

By not separating based on map, results are weaker than they would have been if trained on the same amount of map-specific data. This is because the CT or T win rate differs depending on the map considered [1]. For example: de_cache has a CT win rate of 54.6% whereas de_dust2 has a CT win rate of 49.6%.

After collecting results on accuracy for all maps, we excluded Support Vector Machines from further consideration because of how inaccurate it was in initial results. We then trained and tested each remaining model on data for each map, and averaged the results to obtain a map-based accuracy for each model. Results from these comparisons are shown in Table II.

TABLE II
AVERAGED MAP-BASED ACCURACY OF ALL MODELS, EXCLUDING SVM AFTER POOR PERFORMANCE ON OVERALL ACCURACY.

Model	Accuracy
Decision Tree	0.837
Logistic Regression	0.761
Naive Bayes	0.748
K-Nearest Neighbours	0.739
Neural Network	0.650

From map-based comparisons, it was evident that the Decision Tree model was significantly outperforming all other models at classifying round winners in CS:GO. We display the confusion matrix for the Decision Tree model for all combined maps in Figure 1. Furthermore, the accuracy of the Decision Tree model on each specific map is shown in Table III.

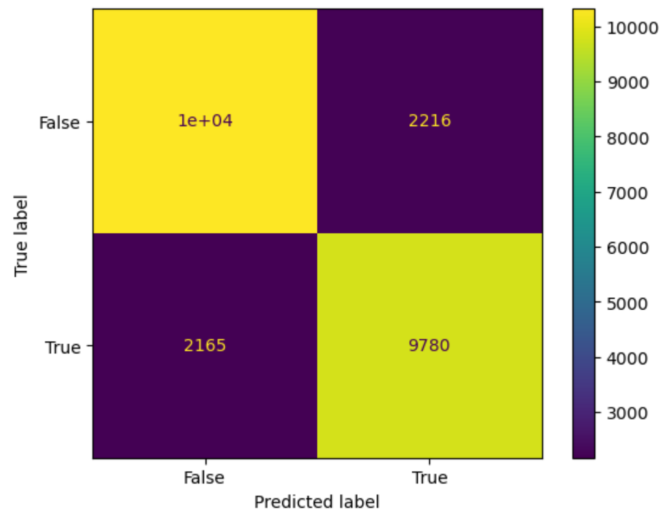

Fig. 1. Confusion matrix for Decision Tree model.

TABLE III
MAP-BASED ACCURACY OF DECISION TREE MODEL

Map	Accuracy
de_cache	0.897
de_dust2	0.847
de_vertigo	0.838
de_inferno	0.837
de_nuke	0.827
de_overpass	0.823
de_mirage	0.822
de_train	0.813

These results highlight the differences that exist between different maps. Some maps in CS:GO are less predictable than others based on this data, with our accuracy results ranging from 0.81 to 0.90. All results are highly accurate and better than expected prior to conducting the study. Improvements could be seen through collecting more features and emphasizing significant features, leading to further investigation.
Summary of results:

- O2: Decision Tree model significantly outperforms all other classification algorithms (KNN, Logistic Regression, Naive Bayes, Neural Network, SVM).
- O3: Map-based accuracy differs significantly for each of the 8 maps available in CS:GO, ranging from 0.813 to 0.897.
- O1: We successfully implemented a classification algorithm to accurately predict CS:GO round winners around 83% of the time.

VI. Conclusions & Future Work

In this study, we investigated the classification of round winners in the popular tactical video-game Counter-Strike: Global Offensive. We addressed O1 through implementation of a successful Decision Tree classification algorithm with 83% accuracy in predictions. O2 was addressed through the comparison of the Decision Tree model with 5 other algorithms that is outperformed significantly. O1 and O2 results are shown in Table I and Figure 1. We addressed O3 by determining that map choice had a significant impact on model results, observed in Table III. We were able to successfully answer the question about whether accurate CS:GO round winner prediction was possible, leading to implications in sports betting.

Future work based on this study includes identifying features that have the greatest impact on prediction results, leading to model improvement through emphasizing these features (ex: squaring features). Furthermore, generalizing results to other e-sports and sports such as Formula 1 is possible through re-use of the GitHub repository[1] with slight modification of target variables and features (ex: Formula 1 has track names instead of driver names, and some tracks allow for less overtaking than other tracks). This study demonstrates the importance of model and algorithm choice in any machine learning task, since different algorithms perform better at different tasks.

REFERENCES

[1] Ct or t sided map stats for counter-strike: Global offensive, 2015.

[2] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification, 2003.

[3] O. Hubáček, G. Šír, and F. Železný. Exploiting sports-betting market using machine learning, 2019.

[4] V. Jakkula. Tutorial on support vector machine (svm).

[5] Neli Kalcheva, Maya Todorova, and Ginka Marinova. Naive bayes classifier, decision tree and adaboost ensemble algorithm – advantages and disadvantages, 01 2020.

[6] A. Kaushal and Manshi Shukla. Comparative analysis to highlight pros and cons of data mining techniques-clustering, neural network and decision tree, 2014.

[7] Christian Lillelund. Cs:go round winner classification, 2023.

[8] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] P. Ranganathan, C. S. Pramesh, and R. Aggarwal. Common pitfalls in statistical analysis: Logistic regression, 2017.

[11] Jill C. Stoltzfus. Logistic regression: A brief primer, 2011.

[12] Peter Xenopoulos, Bruno Coelho, and Claudio Silva. Optimal team economic decisions in counter-strike, 2021.

[1] https://github.com/rgavigan/csgo-predicting