# CS: Go Professional Game Match Data Mining

**Milestone: Project Report**

**Group 1**

**Hao Wang**

**Zhongyu Zhang**

**617-774-7755 (Tel of Hao Wang)**

**405-837-4714 (Tel of Zhongyu Zhang)**

wang.hao15@northeastern.edu

zhang.zhongyu@northeastern.edu

**Percentage of Effort Contributed by Hao Wang: 50%**
**Percentage of Effort Contributed by Zhongyu Zhang: 50%**

**Signature of Hao Wang:** Hao Wang
**Signature of Zhongyu Zhang:** Zhongyu Zhang

**Submission Date: 12/9/2022**

**Contents**

## Part I. Project Proposal

### 1. Problem Setting

The video game presented by Valve Games named Counter-Strike: Global Offensive (CS: GO) is a worldwide popular FPS (first-person shooter) game. It has been an eSport for a long time, since 2013. The winner of the world championship could receive a pretty penny.

The rule of the championship games is easy to understand -- each team has five players, and they need to play as Counter-Terrorists and Terrorists in three games on three maps. Each game would be 30 rounds, and the team who won 16 rounds first would win a game. Seven maps in the map pool are available to be played, and each team could remove one map from the map pool before one game starts to get advantages for their sides.

The principle of this project is to use data mining techniques to analyze different professional gaming teams using the data set from kaggle.com named CS: GO Professional Matches.

### 2. Problem Definition

There would be many factors that affect the result of the world championship. This project aims to build a model that could predict the result of the winning team in the championship before the game starts. The prediction accuracy expected will be 60%. To achieve the goal, "linear regression" and "decision tree" will be used in this project.

### 3. Data Sources

CS: GO Professional Matches data set
https://www.kaggle.com/datasets/mateusdmachado/csgo-professional-matches?select=picks.csv

4. **Data Description**

The data set records are 43234 world championship game matches from 11/2015 to 03/2020. There are four tables in this data set, “Results.csv” contains map scores and team rankings; “Picks.csv” contains the order of map picks and vetoes in the map selection process; “Economy.csv” contains round start equipment value for all rounds played; “Players.csv” contains individual performances of players on each map. Within four tables, the “eventid” and “matchid” columns are unique for each game match and shared between tables. Thus, tables could be merged by these two identities.

## Part II. Data Collection, Data Visualization, Data Exploration, and Data Processing

1. **Data Collection & Data Processing**

As the description above, the data contains four .csv files. The data set needs to be cleaned, reduced and regrouped for analytics purposes. Since “economy.csv” and “results.csv” have similar sizes of data, a new data frame could be generated by merging them. However, “players.csv” is more independent, and it describes, for each match, every individual player's status, which could be used separately in the future. For picks.csv, since it missed 3/4 of the match record, it could be separated as well. After cleaning, reducing, and regrouping the data from “economy.csv” and “results.csv”, the following data shown below in figure 1 could be generated.

| | date | match_id | event_id | team | _map | start_as | result | ct_win | t_win | rank | team_eco | win_rate | won/lost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2020-03-01 | 2339402 | 4901 | Natus Vincere | Nuke | ct | 16 | 14 | 2 | 6 | 479750.0 | 0.80 | 1 |
| **1** | 2020-03-01 | 2339402 | 4901 | Natus Vincere | Dust2 | t | 16 | 8 | 8 | 6 | 578200.0 | 0.55 | 1 |
| **2** | 2020-03-01 | 2339402 | 4901 | Natus Vincere | Mirage | ct | 16 | 14 | 2 | 6 | 461650.0 | 0.89 | 1 |
| **3** | 2020-02-29 | 2339401 | 4901 | Astralis | Dust2 | ct | 5 | 1 | 4 | 1 | 376300.0 | 0.24 | 0 |
| **4** | 2020-02-29 | 2339401 | 4901 | Astralis | Nuke | t | 5 | 3 | 2 | 1 | 370550.0 | 0.24 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **31705** | 2017-04-04 | 2309263 | 2683 | HellRaisers | Cache | ct | 16 | 9 | 7 | 15 | 519450.0 | 0.62 | 1 |
| **31706** | 2017-04-04 | 2309262 | 2683 | CLG | Train | ct | 11 | 9 | 2 | 24 | 524650.0 | 0.41 | 0 |
| **31707** | 2017-04-04 | 2309261 | 2683 | TYLOO | Mirage | t | 12 | 4 | 8 | 33 | 513800.0 | 0.43 | 0 |
| **31708** | 2017-04-04 | 2309260 | 2683 | MVP Project | Mirage | ct | 4 | 4 | 0 | 98 | 310700.0 | 0.20 | 0 |
| **31709** | 2017-04-04 | 2309259 | 2683 | Immortals | Cobblestone | ct | 3 | 3 | 0 | 7 | 258600.0 | 0.16 | 0 |

31710 rows × 13 columns

Figure 1

Within the new dataset:

"date" describes the game matching date;

"match_id" describes the id of the matches;

"event_id" describes the id of the events;

"team" identifies the team who plays in this match;

"_map" shows the map of this match used;

"start_as" describes the side of the team when the match starts;

"result" shows how many turns the team won in that match (total turn is 30);

"ct_win" describes win times when the team on the ct side;

"t_win" describes win times when the team on the t side;

"rank" is the team rank of the world;

"team_eco" describes the total economy the team got in that match;

"win_rate" is the win rate in 30 turns;

"won/lost" shows if the team won the match or not (shown as binary).

## 2. First Phase of Data Analysis and Plots

By using the data that has been generated, below basic analytics could be produced. However, since this project aims to predict the world championship's winning rate, the top 20 and 20 ranked teams are focused on.

## 2.1 Top 20 Frequency Won Team (Figure 2)

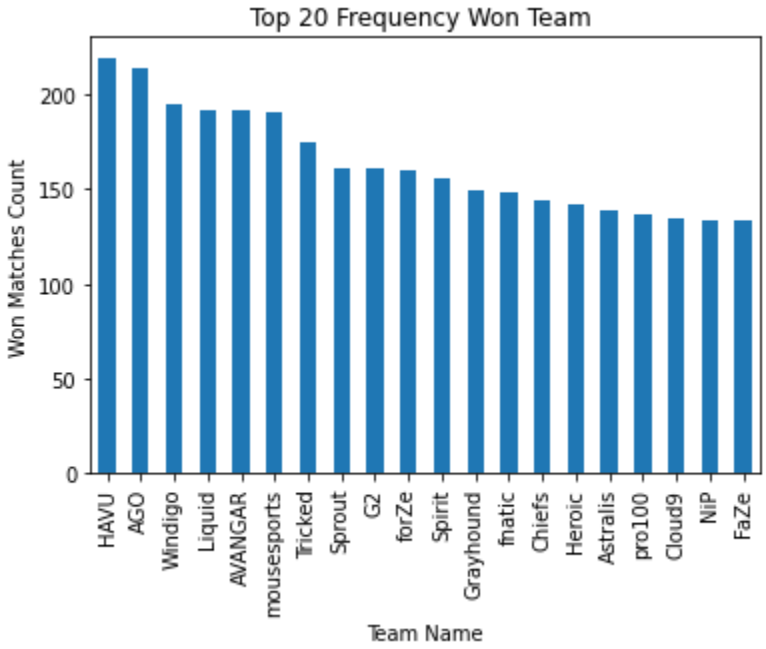


Figure 2

Figure 2 shows that within the top 20 frequency-won teams, AGO has the highest frequency of won matches, and NRG has the lowest frequency of won matches. However, all the top 20 teams have won at least 150 games.

## 2.2 Top 20 Rank Team Won Matches Count (Figure 3) & Top 20 Rank Team Average Win Rate (Figure 4)

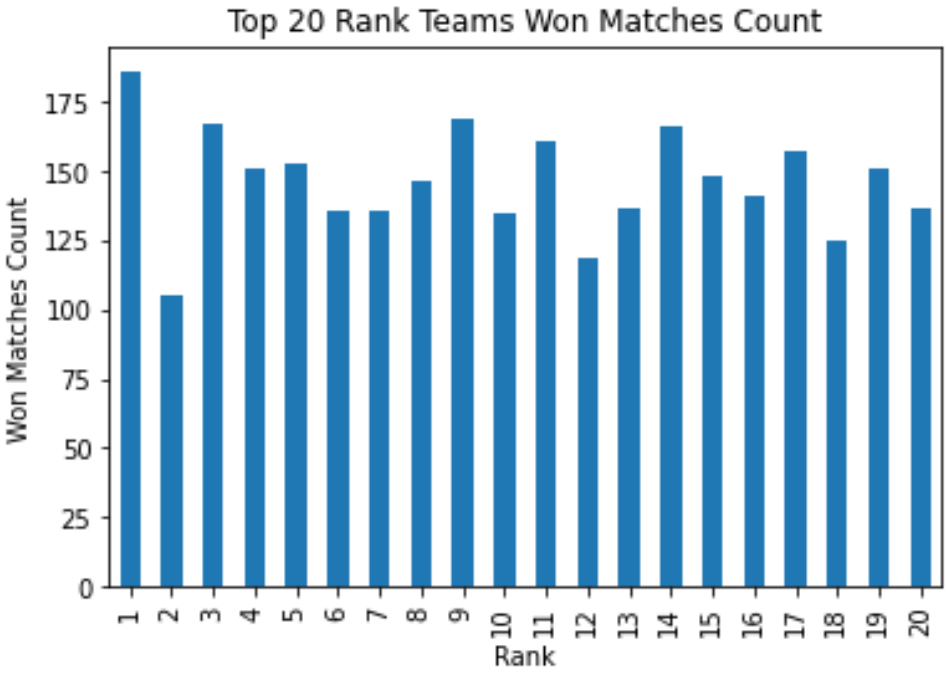


Figure 3

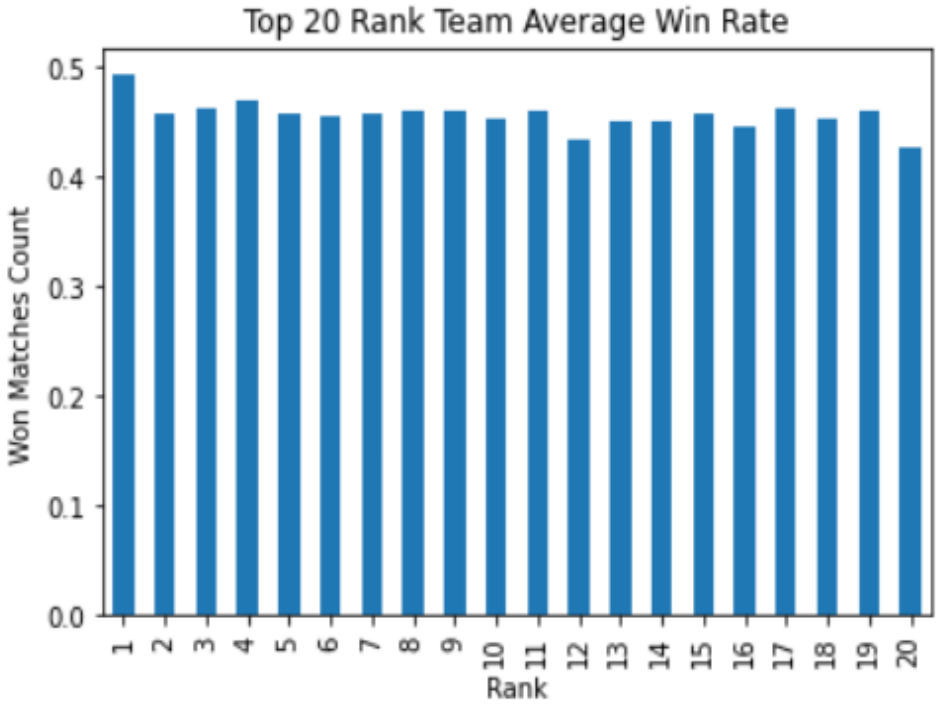


Figure 4

Figure 3 shows all top 20 ranked teams that won matches counted above 100, similar to figure 2. The difference between figure 2 and figure 3 might be because the world ranking changes yearly. The top 1 team won frequency is above 175.

Figure 4 shows that the win rate of the top 20 ranked teams is above 0.4, and the difference between different ranks is slight. Thus, for top-ranked teams, the gaming skill gaps are small.

### 2.3 Match Count by Year

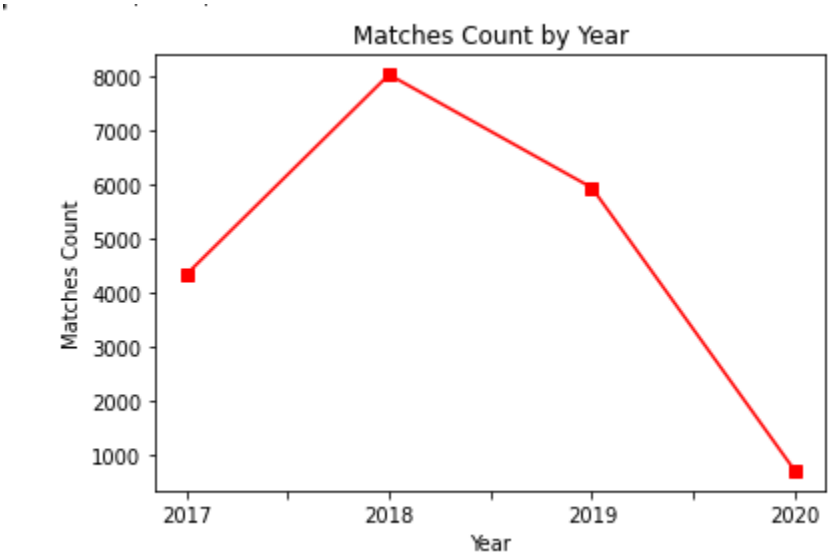


Figure 5

We also generated a line chart showing the Matches Count by Year (Figure5), and as the figure shows, in 2018 there are 8000 matches total and it is the most matches counted in 4 years. After 2018, the matches are reduced significantly. That might be because the pandemic started at the end of 2019, and the world championship was influenced by it.

### 2.4 Correlation Numerical Data

For analyzing the relationship between different efforts, we can generate a heat map (Figure 6) for showing the correlation between different efforts.

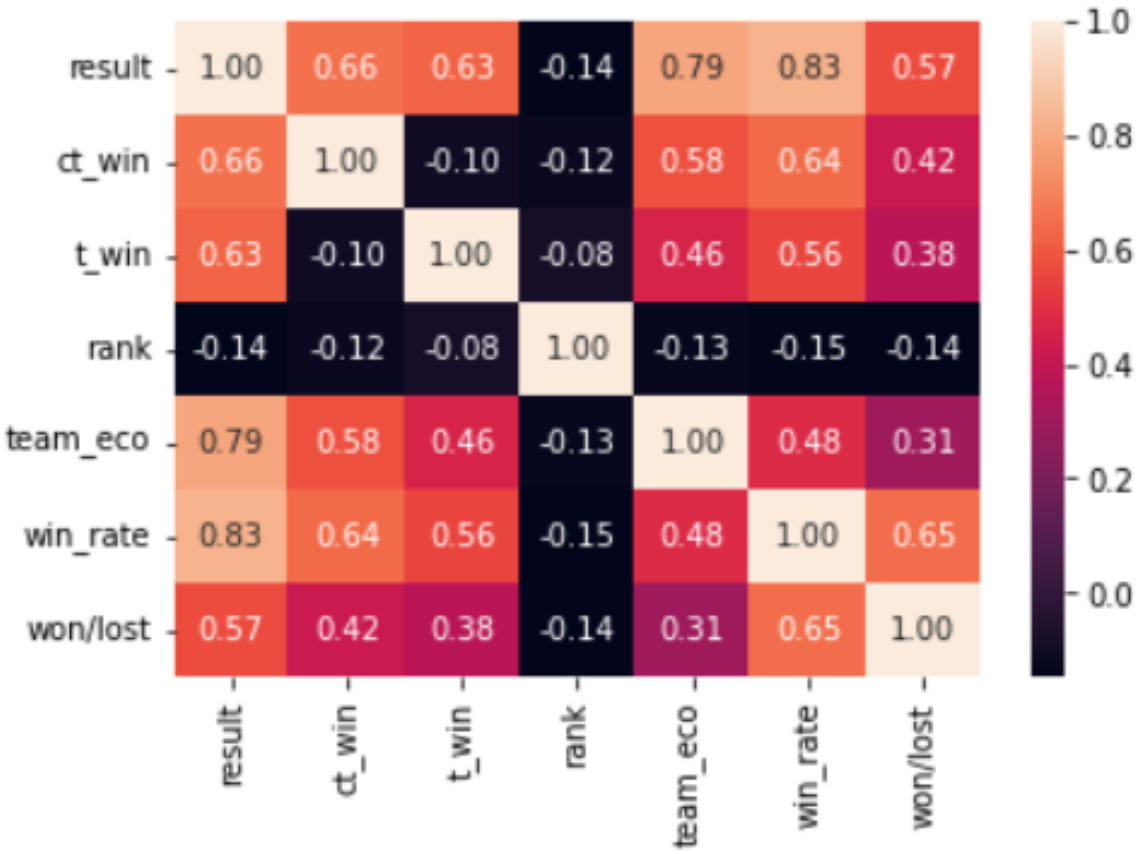


Figure 6

The heat map shows all numerical data are weakly correlated. Since shown in the heat map, match win/loss is weakly correlated to all of the other values, we decide to use PCA to reduce the dimension to speed up the model fit for k-NN model in the following step.

For value ‘rank’, even if it shows it is weakly correlated to match win/lost, it still has a strong correlation compared with other values. So we decide ‘rank’ is one of these dimensions as well.

By using plot, we can also generate a relationship scatter chart table (Figure 6) for showing all relationships between different numerical data.

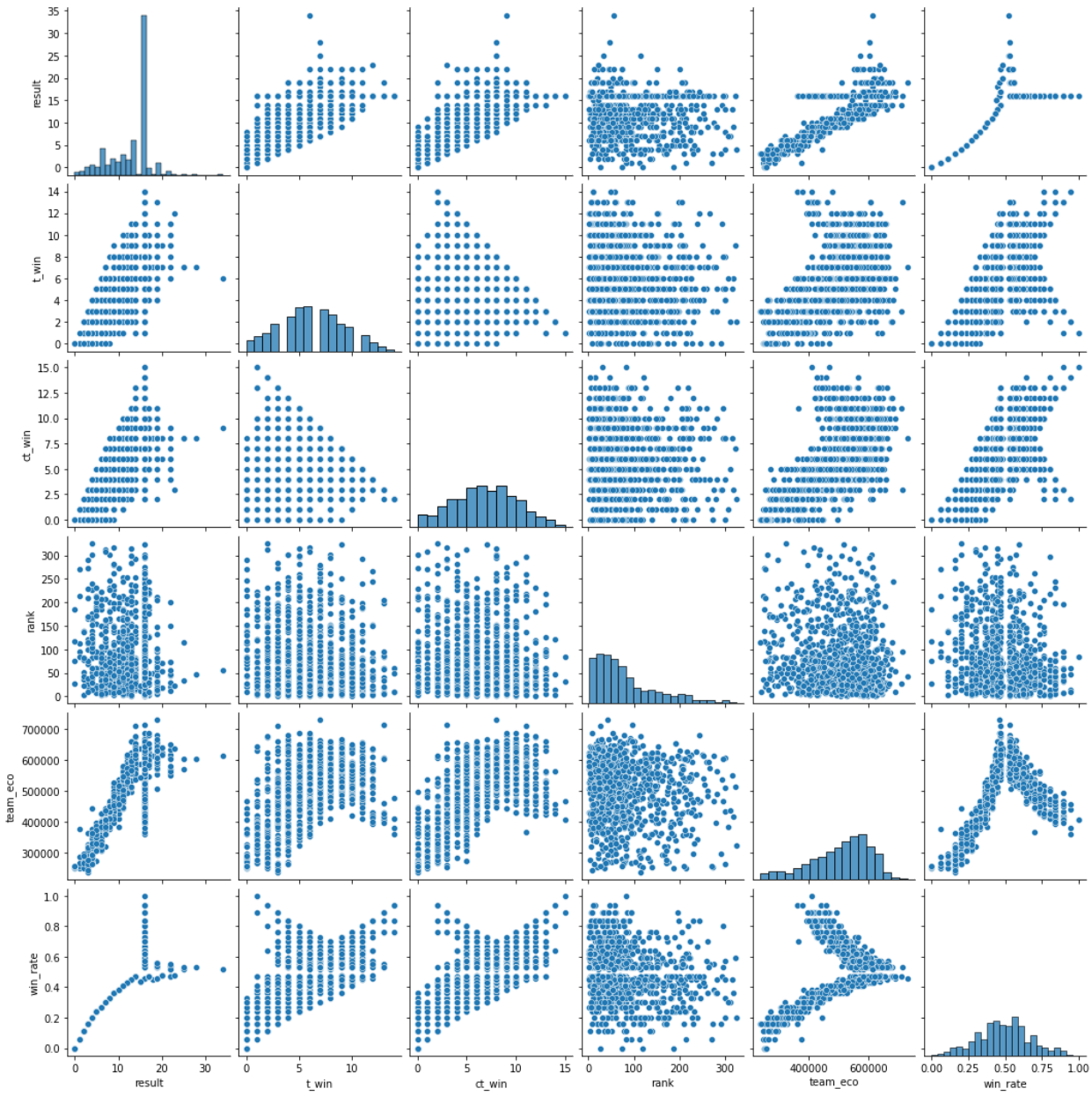


Figure 7

## 2.5 Dimensional Reduction

For speeding up the fitting of certain machine learning algorithms like k-NN, we decide to use PCA to reduce the dimension of our data. The dimension was reduced into 2-dimensional after we use the PCA algorithm, and the table(Figure 8) is shown below

| | principal component 1 | principal component 2 | won/lost |
|---|---|---|---|
| **0** | -1.724609 | -2.555199 | 1 |
| **1** | -1.420329 | 0.022618 | 1 |
| **2** | -1.887755 | -2.542620 | 1 |
| **3** | 2.848699 | 0.385657 | 0 |
| **4** | 2.851548 | -0.488612 | 0 |
| **...** | ... | ... | ... |
| **31705** | -1.339754 | -0.375054 | 1 |
| **31706** | 0.332325 | -1.620384 | 0 |
| **31707** | 0.180115 | 0.839892 | 0 |
| **31708** | 3.617945 | -0.985549 | 0 |
| **31709** | 4.031411 | -0.945100 | 0 |

31710 rows × 3 columns

Figure 8

In this table, won/lost is the target we would like to analyze with. By using this table, we could plot a scatter chart(Figure 9) for showing the difference between winning and losing matches affected by principal components 1 and 2, and it would help us speed up the fitting of models.

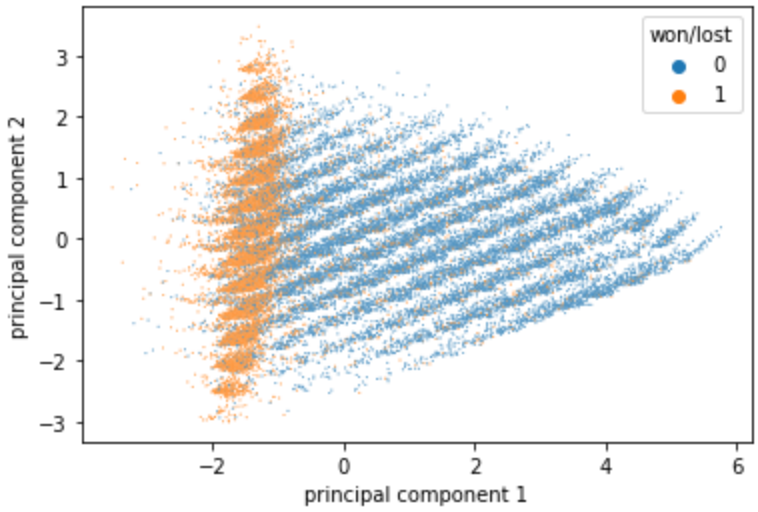


Figure 9

**Part III. Exploration of candidate data mining models, and select the models**

1. **Partitioning data for training and evaluation**

Because our data set contains both numerical data and categorical data, we decided to use multiple models to test which models fit our data best. The models are including but not limited to k-NN, the Naive Bayes Classifier, Logistic Regression and Regression Trees.

We partitioned the dataset into two parts with 75% training data and 25% test data. Training data is for training the models, and test data is used at the end for doing the final unbiased evaluation of the model.

For evaluating which model that fits our data best, we did model evaluation by calculating prediction accuracy for each model. The table below (Figure 10) shows prediction accuracy of each model.

| Model | Prediction Accuracy | F1 Score |
|---|---|---|
| **k-NN** | 0.77 | 0.75 |
| **Naive Bayes Classifier** | 0.81 | 0.79 |
| **Regression Trees** | 0.78 | 0.78 |
| **Logistic Regression** | 0.84 | 0.84 |

Figure 10

As the table shows, the prediction accuracies are about the same, but the Logistic Regression has the highest accuracy and F1 score. Thus, we will select the Logistic Regression as our model.

## Part IV. Performance Evaluation

The logistic regression model that performed best on the validation data was selected for evaluation on the test partition of the dataset (25% Test Partition).

1. **ROC Curve analysis**

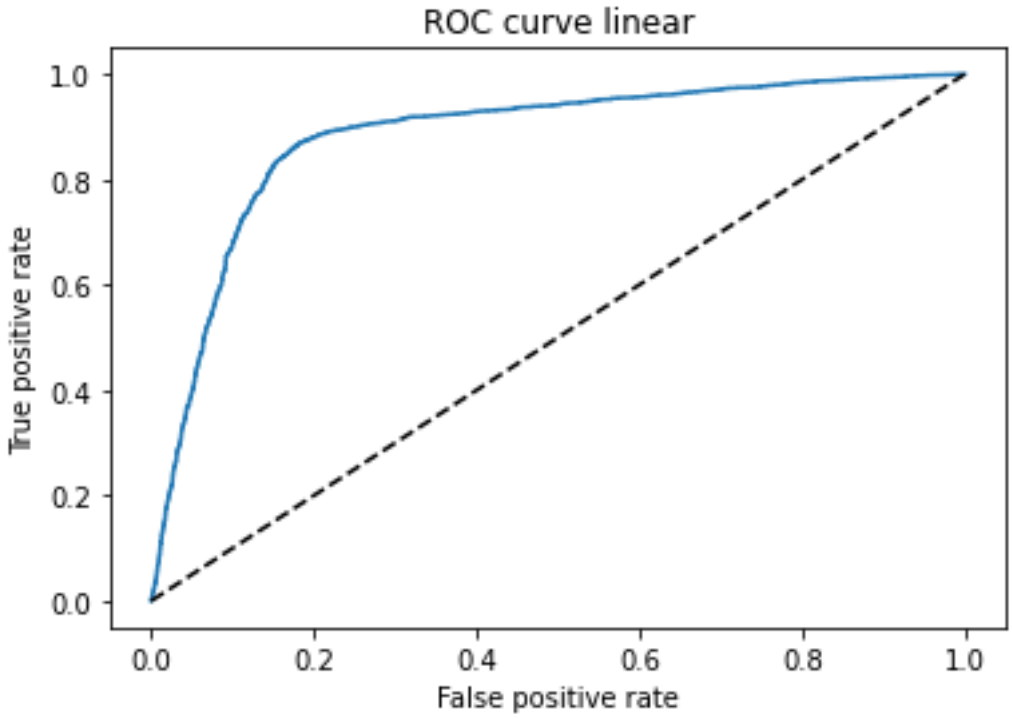


Figure 11

In the chart (Figure 11) above, the blue line is the ROC curve. Its curve tends to the upper left corner, it has a better roc curve, In other words, this means that the model can have a high percentage to detect which team will win or not.

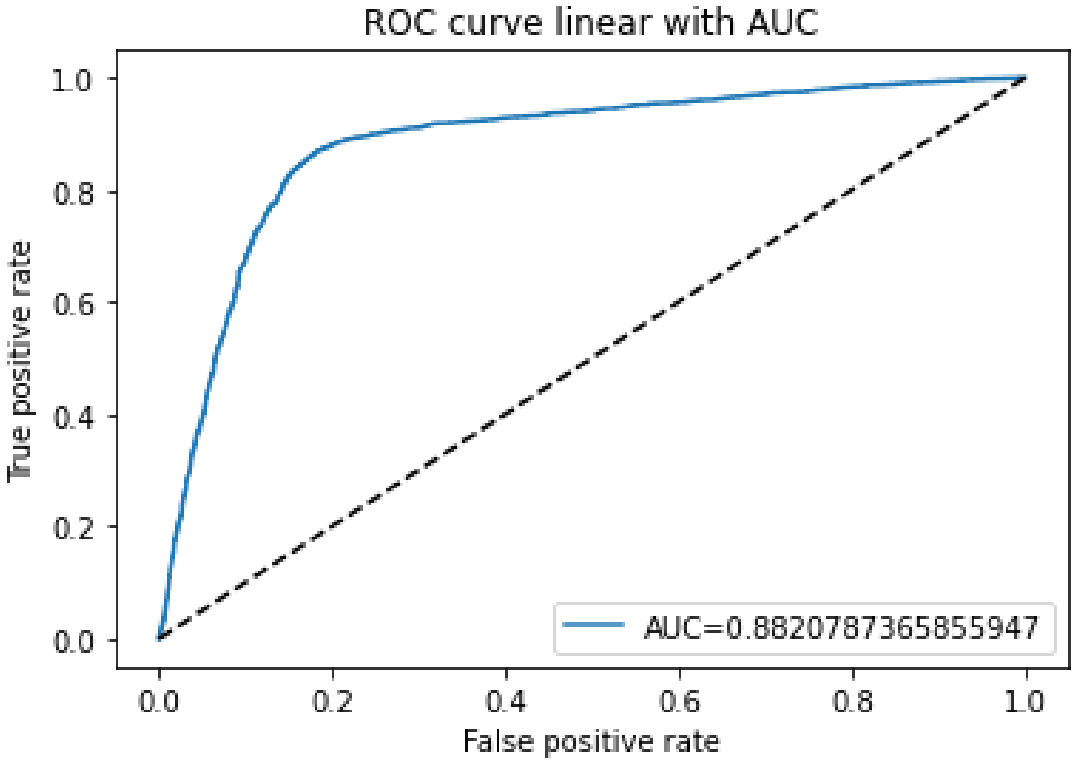


Figure 12

At the same time, we can quantify this curve, using AUC, the area under the curve. It will tell us how many parts are below this curve. The closer the AUC is to 1, the better the model is. By calculation, the value of AUC is 0.88. It shows that this model does a good job in data classification. It also proved to be highly accurate in predicting victory and failure.

**2. F1 Score, Recall and Precision**

```
[120] print(f1_score(y_test, y_pred))

      0.8298989898989898
```

Figure 13

From the above graph (Figure 13) the score of F1 is 0.847, which is a relatively high value, it means that the precision and recall are still relatively similar, and this proves that the model is very accurate.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.86 | 0.84 | 0.85 | 4261 |
| 1 | 0.82 | 0.84 | 0.83 | 3667 |
| accuracy | | | 0.84 | 7928 |
| macro avg | 0.84 | 0.84 | 0.84 | 7928 |
| weighted avg | 0.84 | 0.84 | 0.84 | 7928 |

Figure 14

Now looking at the precision and recall from the graph (Figure 14) above, when the model predicts that the first team will win, it is correct 82% of the time. The recall value indicates that the model correctly detects 84% of the cases where the first team wins.

**Part V. Project Results**

The average person tends to focus on their favorite team or the team they are from locally. People will not judge very accurately for other regions or teams that are not very well known. Then it is helpful to use our model to simulate wins and losses. The model simulations are 84% correct so that gambling companies, game officials, and spectators can better understand the game.

At the same time, people have personal emotional factors that influence which team wins. As a result, they often cannot do a fair and impartial analysis. But with our model simulation, it does not receive the influence of external factors and can achieve an 84% accuracy rate.

## Part VI. Impact of Project Outcomes

We used data mining, data analysis, and machine learning to gain insight into the popular game Counter-Strike Global Offensive.

Initially, we collected data on win/loss results, professional players, professional teams, weapon selection, and economic status in this game. These things have an essential impact on the game, winning or losing.

We finally chose logistic regression for model selection because we found that only it had the highest correctness rate compared to the other three models. It has the highest correct rate of 84%.

Our model is 84% correct, and we think it is a tool that a game official company can use to predict tournament results in advance. They can be better prepared this way for some game data analysis before the tournament. Commercially, as more and more people are willing to gamble on eSports. For some gambling companies, predicting the tournament results in advance helps them make more money because they can also use the model to calculate the odds.

## Citation

Machado, Mateus Dauernheimer. “CS: GO Professional Matches.” *Kaggle*, 26 Mar. 2020, https://www.kaggle.com/datasets/mateusdmachado/csgo-professional-matches.