Thong Sheng AngFinal Project ReportSTAT 5850

DESCRIPTION OF THE DATASET

The dataset that I will be using for this final project is the “csgo.csv” dataset. CS:GO (or Counter Strike: Global Offensive) is a popular multiplayer first-person shooting game that has been around since 2012. This dataset is in a csv format and it consists of 122411 total observations, along with 97 different variables, so it is a rather big dataset. Some examples of the variables are “time_left” (time left in the round), “ct_score” and “t_score” (scores for Counter-Terrorist and Terrorist teams), “round_winner” (winner for each round), and so on.

DESCRIPTION OF THE PROBLEM

The problem that I will be asking in this project is “Who will be the winner of each round, either CT (Counter-Terrorists) or T (Terrorists), judging from some of the variables in the dataset?”

Based on the description of problem above, this project will be a Prediction and Classification problem. I will be using some of the explanatory variables to predict the label of the response variable (in this case “round_winner”). Since the response variable is a categorical variable with only two outcomes (CT or T), this will be classified as a binary classification problem.

CODE

Refer to R script (STAT5850 Final Project.R) attached.

EXPLANATION OF ANALYSIS

Data preparation and cleansing

The csgo.csv dataset was imported into RStudio using “read.csv” function. As it is a relatively large dataset, it might take some time for RStudio to fully import the entire dataset.

This dataset also contains a lot of unnecessary information. I will not be using all of the 97 variables, therefore the variables that are not used will be removed. In the end, only 13 variables that are deemed useful for analysis were included.

```
> names(csgo)
 [1] "time_left"        "ct_score"         "t_score"          "bomb_planted"
"ct_health"
 [6] "t_health"         "ct_armor"         "t_armor"          "ct_money"
"t_money"
[11] "ct_players_alive" "t_players_alive"  "round_winner"
```

Next, I changed the data types of the variables accordingly. For example: the observations in “round_winner” will be changed from “character” type to “factor” type to facilitate the analysis and modeling processes. Using the “contrasts” function, a factor level of 0 was assigned to CT while T was assigned a value of 1.

```
> contrasts(csgo$round_winner)
   T
CT 0
T  1
```

The last phase of data preparation is to split the dataset is into training and testing sets. This will facilitate the analysis and modelling processes, especially when retrieving the test error rates of each classification models in the later part of this project.

Classifiers

There are a lot of statistical classification methods out there. However, I will only apply a couple classifiers that I deem fit for this dataset, namely:

1. Logistic regression
2. Linear discriminant analysis (LDA)
3. Quadratic discriminant analysis (QDA)
4. Classification tree
5. Pruned classification tree.

Multiple full models were fitted on the csgo dataset using all the classification model mentioned above. Then, a test error rate is calculated for every model fitted. This test error rate will be used later on to determine which classification method is the most effective in predicting the label for the responding "round_winner" variable.

R Packages

Only two packages were used in this project, the MASS package, and the tree package. MASS is required for the lda and qda classifiers, whereas the tree package was used to build classification tree.

RESULTS

Tables

A table was created for each classifier to show the number of observations that were labeled correctly and incorrectly. For example, we have a table created for logistic regression below:

```
> logit.table
          y.test
logit.pred    CT     T
        CT 21643  7358
        T   8454 23750
```

The table shows that 8454 observations were falsely labeled as T when they are supposed to be CT, and 7358 observations were falsely labeled as CT. So, we can obtain the test error rate by simply dividing the total number of incorrectly labeled observations by the total number of observations.

```
> logit.error = (logit.table[1,2] + logit.table[2,1])/(sum(logit.table))
```

Ultimately, a final table was constructed to compare the test error rate of all classifiers used above.

```
> comparison_table
            test error rate
logit.error       0.2583449
lda.error         0.2588187
qda.error         0.2654685
tree.error        0.2772976
prune.error       0.2772976
```

From this final table, we can conclude that logistic regression might be the most effective classification method to predict the label of “round_winner” for this dataset, as it achieved the lowest test error rate, with LDA at a very close second.

The method used to select the best classifier in this project is the validation set approach, where the dataset is randomly separated into training and testing sets. The training set is used to build the model, and the model with the lowest error rate when applied to the testing set will be chosen as the “best model”

Discussion

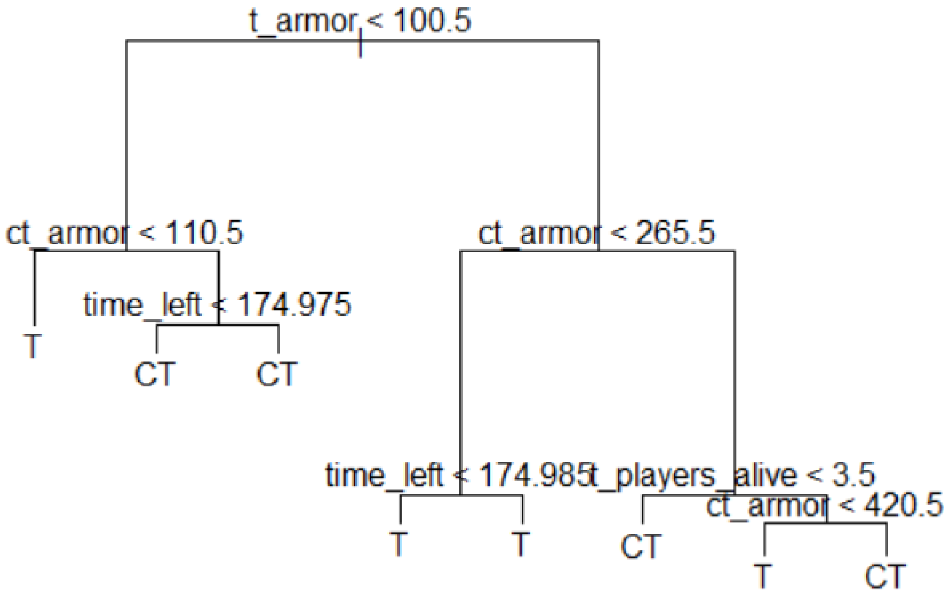
Logistic regression and LDA both produce linear boundaries to classify the label of observations, whereas QDA and classification trees produce a moderately non-linear boundary. LDA assumes that the observations are drawn from the normal distribution with common variance in each class, while logistic regression does not have this assumption.

The fact that logistic regression and LDA outperform QDA and classification tree proves that the true decision boundary for this dataset might be more linear. We cannot conclude whether the dataset holds the assumption of normality because the error rates of LDA and logistic regression are extremely close.

Besides that, noticed that the test error rate of classification tree and its pruned version are identical. This means pruning might not be necessary for this dataset as the number of terminal nodes might be optimum in this case. A different result might be achieved if both of these methods were applied to the original csgo dataset where all 97 variables are used.

Plot

The classification tree plot is printed below. The plot for the pruned classification tree is identical to this plot as well. From here, we can say that ct_armor might be the most important variable as it appeared the most frequently compared to other variables.



Reference

Lillelund, C. (2020, August 19). *CS:GO Round Winner Classification.* Kaggle.com. Retrieved from: https://www.kaggle.com/christianlillelund/csgo-round-winner-classification