How to Win ?

Explore the impact of behaviors in the game on winning or losing

112024502 黃品瑜
112024505 林峻瑋
112024509 吳振瑋
112024517 王品善
112024521 潘易承

Contribute of this report

黃品瑜	Decision Tree , Integrate the slide template , variable selection , edit ppt
林峻瑋	gboost , Slide design and formatting , variable selection , edit ppt
吳振瑋	EDA , KNN , Integrating the code , variable selection , edit report
王品善	Random Forest, Proofreading code , slides, and report , variable selection , edit ppt
潘易承	Logistic Regression , Compile the written report , variable selection , edit report

Motivation

Why do we want to explore victory or defeat in League of Legends?

purpose

1. How to predict the outcome of a game within ten minutes of data (determining the best model).

2. How to quickly determine the outcome of a game (determining the most important variable).

Introduction of data

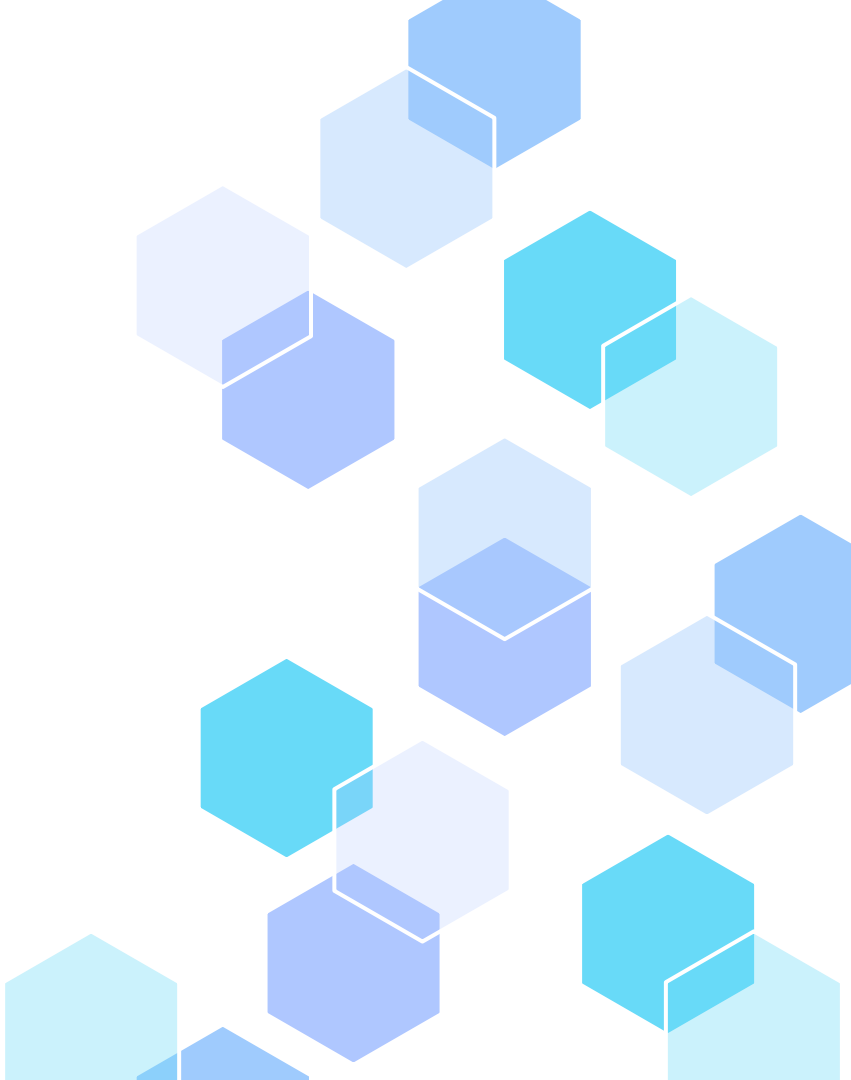
This dataset contains the first 10 minutes which come from the game "League of Legends (LoL)". Stats of approximately 10k ranked games from a high ELO (DIAMOND I to MASTER). There are 19 features per team (38 in total) collected after 10min in-game.

This includes kills, deaths, gold, experience, level, etc. The column blueWins is the value we are trying to predict. A value of 1 means the blue team has won. We want to predict which features are more correlated with winning.

Table of contents for the methodology

01

Exploratory Data Analysis

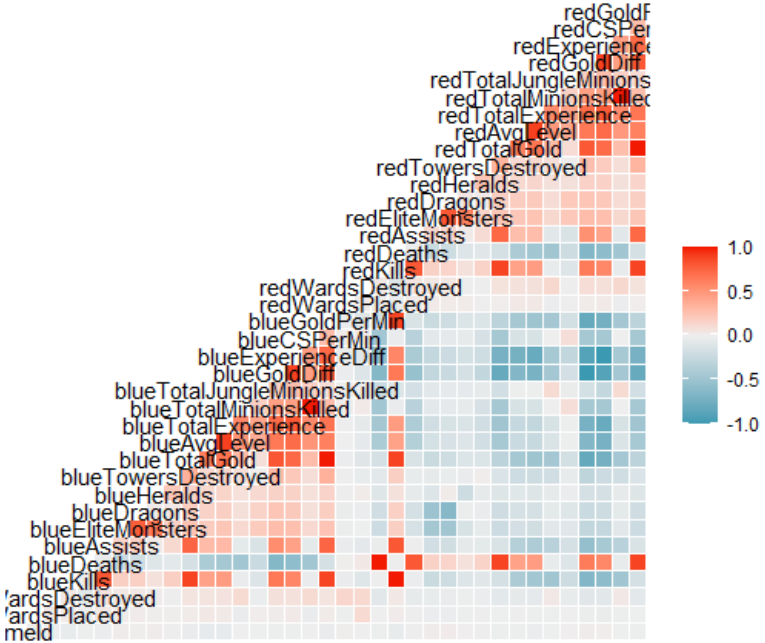


Introduction of Variable

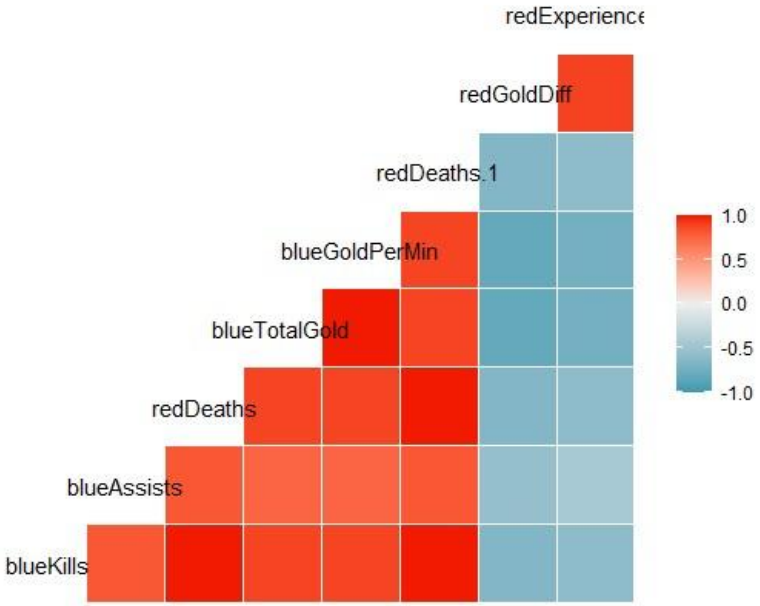
gameId (遊戲場次編號)
blueWins (藍方獲勝)
blue/redWardsPlaced (藍/紅方插眼數)
blue/redWardsDestoryed (藍/紅方拆眼數)
blue/redFirstBlood (藍/紅方首殺)
blue/redKills (藍/紅方擊殺數)
blue/redDeaths (藍/紅方死亡數)
blue/redAssists (藍/紅方助攻數)
blue/redEliteMonsters
(藍/紅方擊殺小龍和諭示者數)
blue/redDargons (藍/紅方擊殺小龍數)

blue/redHeralds (藍/紅方擊殺諭示者數)
blue/redTowerDestoryed (藍/紅方防禦塔破壞數)
blue/redTotalGold (藍/紅方總經濟)
blue/redAveLevel (藍/紅方平均等級)
blue/redTotalExperience (藍/紅方總經驗)
blue/redTotalMinionsKilled (藍/紅方總小兵擊殺數)
blue/redTotalJungleMinionsKilled (藍/紅方總野怪擊殺數)
blue/redGoldDiff (藍/紅方經濟差)
blue/redExperienceDiff (藍/紅方經驗差)
blue/redCSPerMin (藍/紅方每分鐘平均小兵擊殺數)
blue/redGoldPerMin (藍/紅方每分鐘平均經濟)

Exploratory Data Analysis



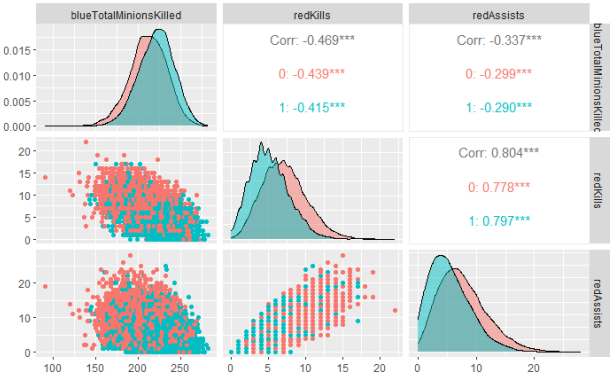
Exploratory Data Analysis

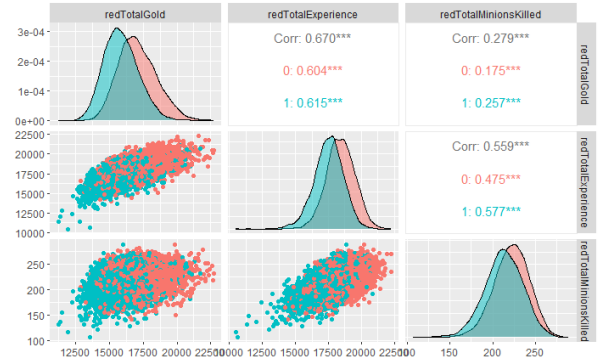


Variable Selection

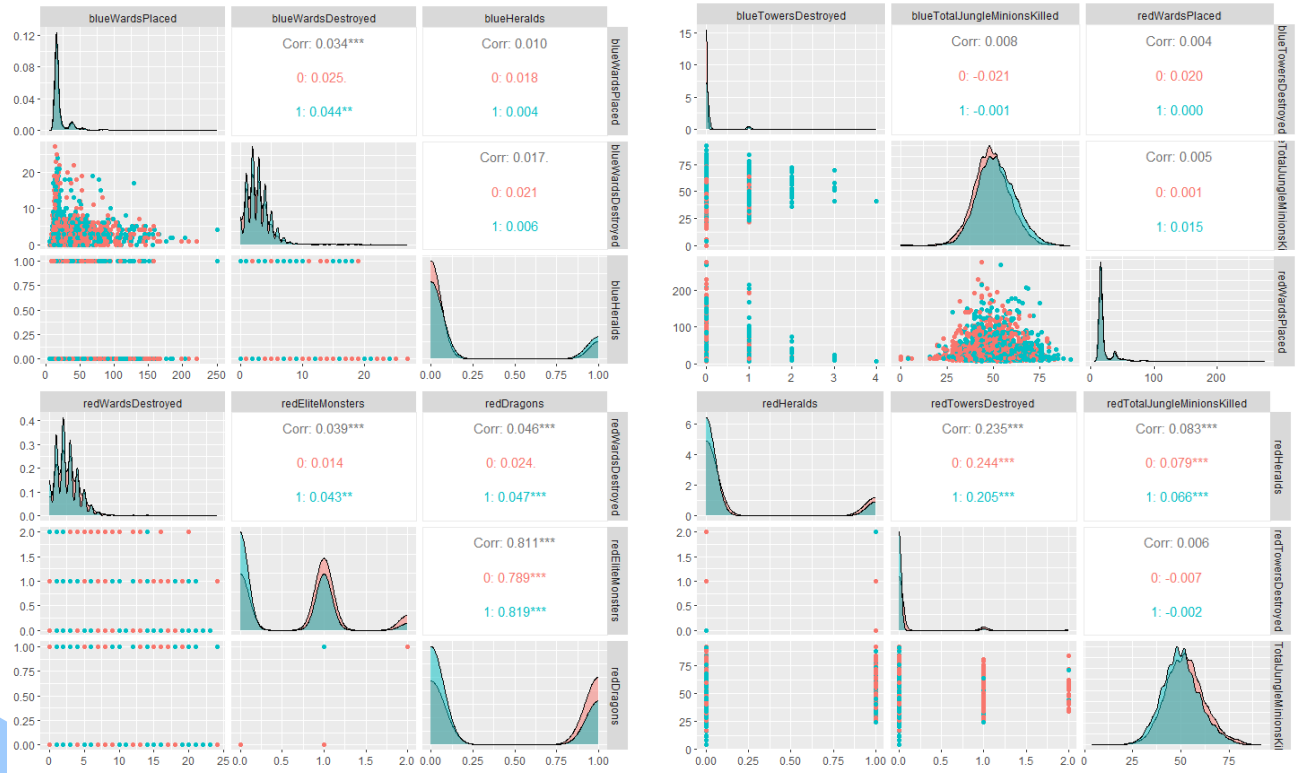
blueKills	redKills
blueAssists	redAssists
blueDragon	redTotalGold
blueTotalGold	redTotalExperience
blueTotalExperience	redTotalMinionsKilled
blueTotalMinionsKilled	blueFirstBlood

Exploratory Data Analysis





Exploratory Data Analysis

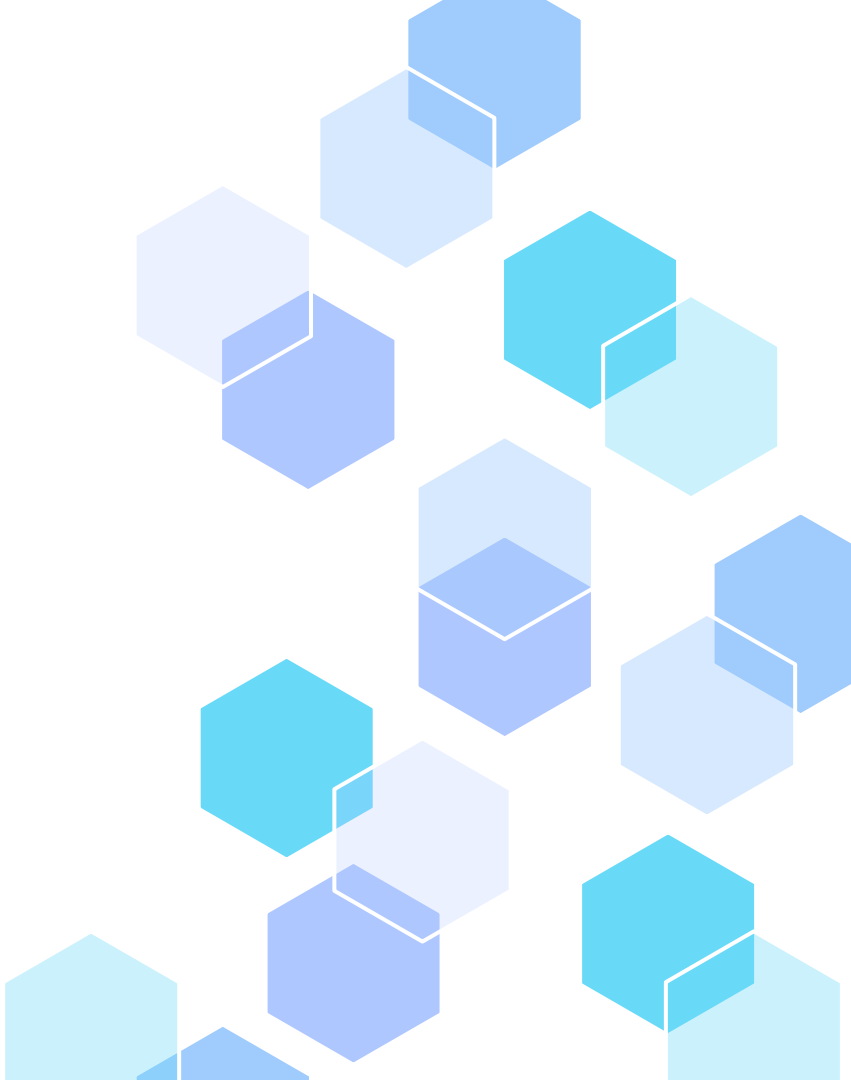


Model Selection (LASSO)

(Intercept)	-0.002186338
xblueKills	.
xblueAssists	-0.053897359
xblueDragons	0.275331303
xblueTotalGold	0.665907476
xblueTotalExperience	0.299188663
xblueTotalMinionsKilled	-0.052336086
xredKills	.
xredAssists	.
xredTotalGold	-0.582962011
xredTotalExperience	-0.321849353
xredTotalMinionsKilled	0.088450130
sel_train$blueFirstBlood1	0.008555861

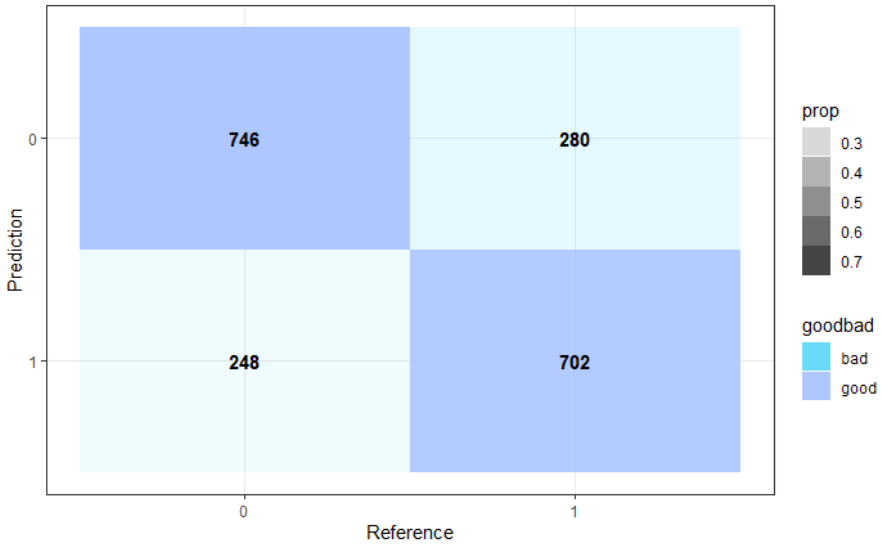
02

K Nearest Neighborhood



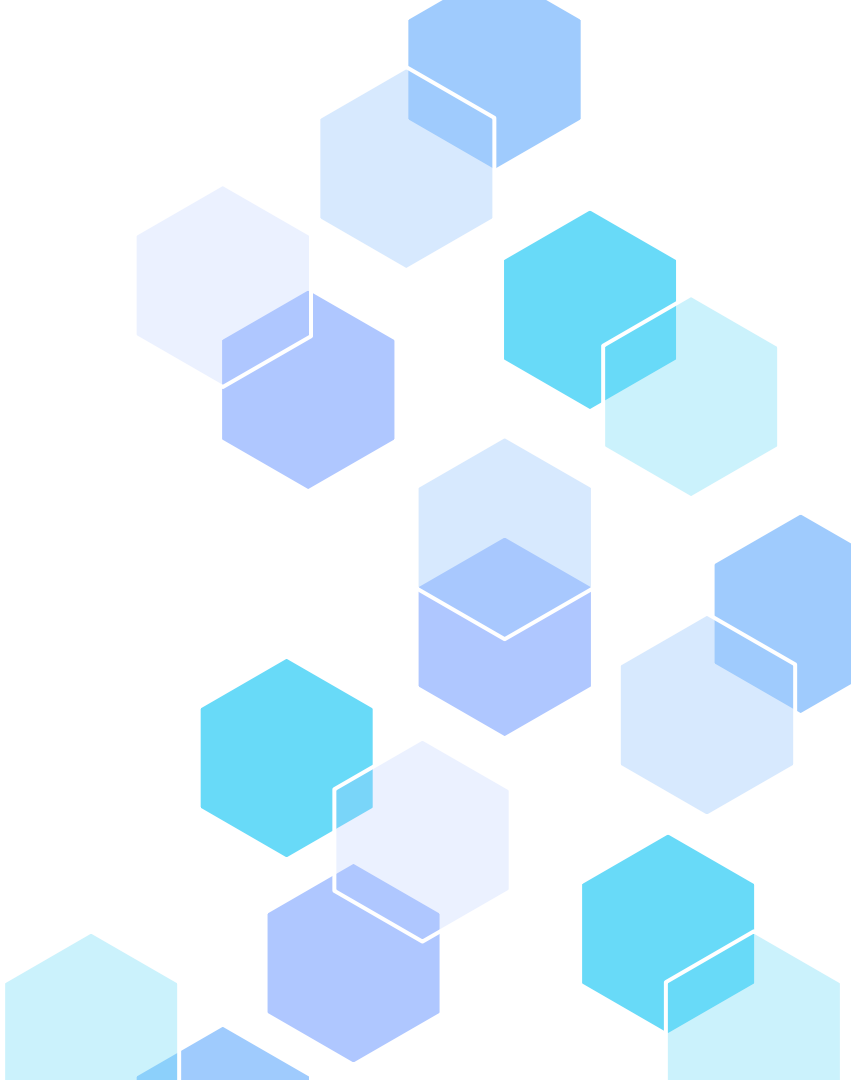
K Nearest Neighborhood

```
Accuracy : 0.7328
Sensitivity : 0.7505
Specificity : 0.7149
```



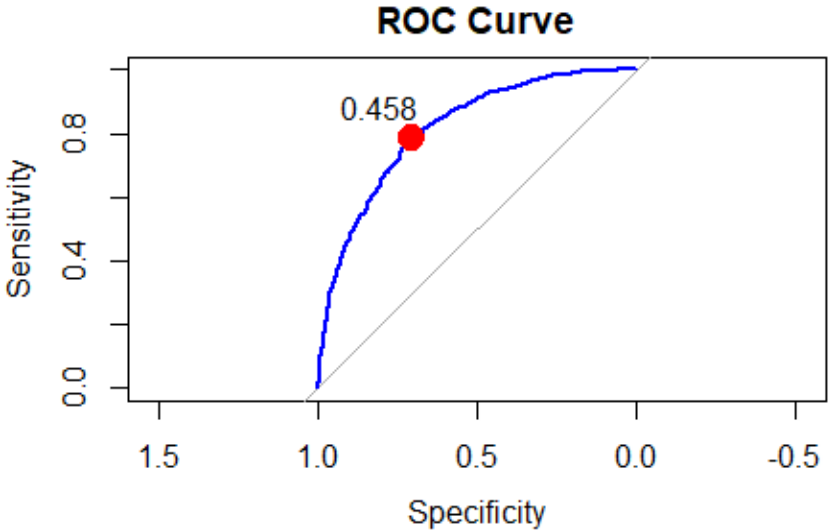
03

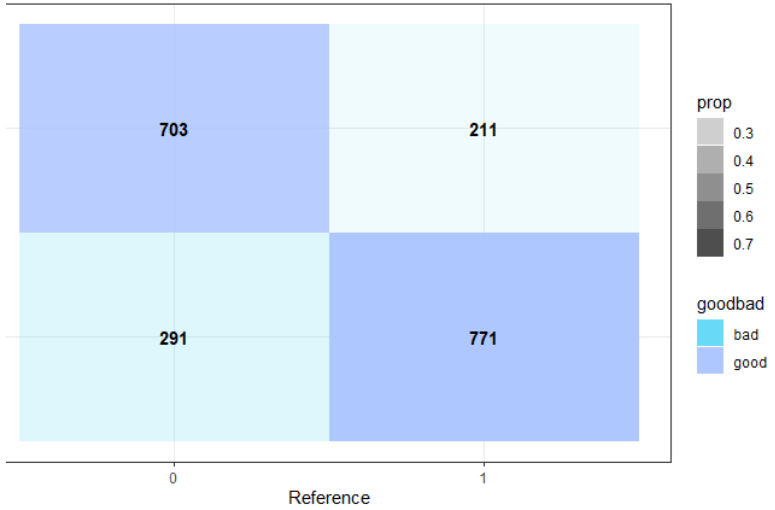
Logistic Regression



Logistic Regression

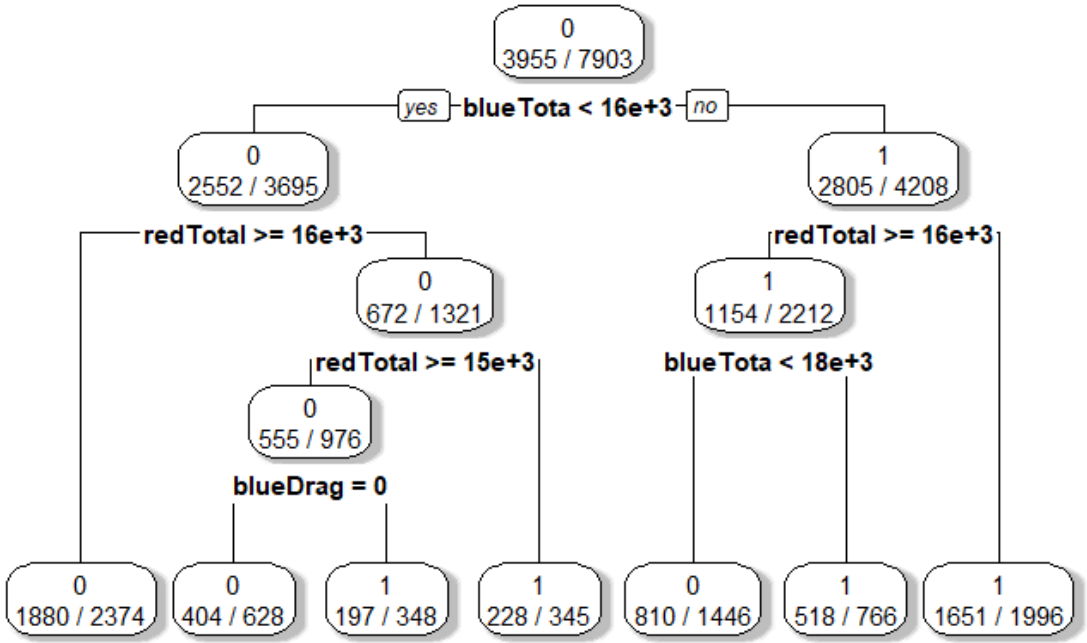
```
Accuracy : 0.746
Sensitivity : 0.7072
Specificity : 0.7851
```



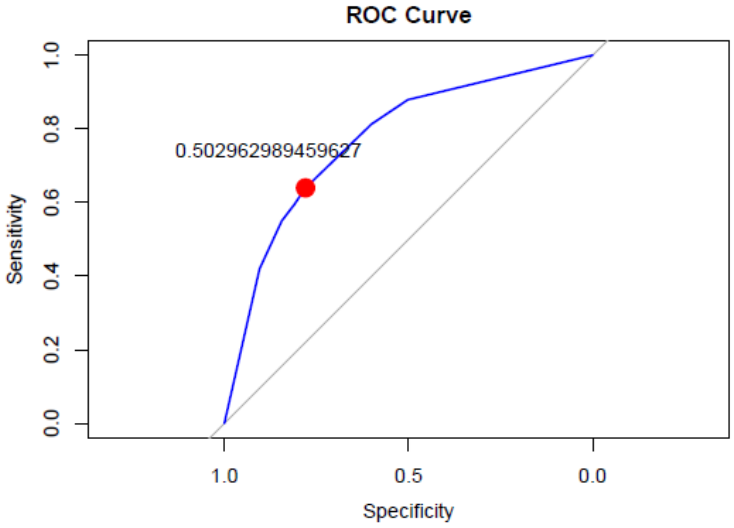


04 Decision Tree

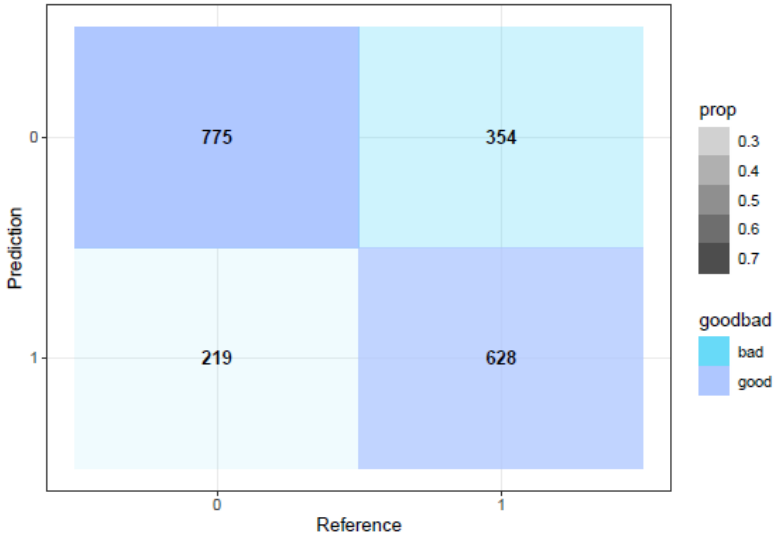
Decision Tree (LASSO)



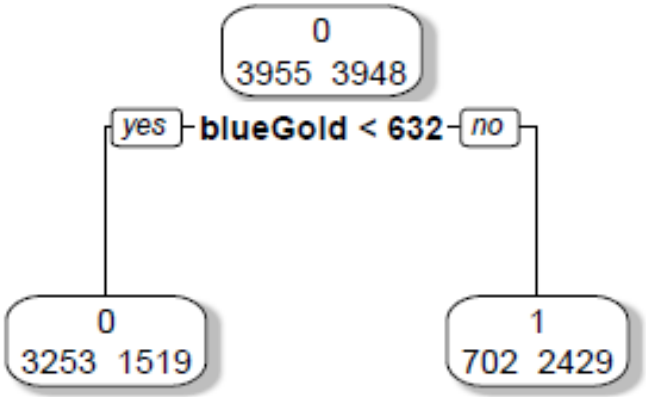
Decision Tree (LASSO)




```
Accuracy : 0.71
Sensitivity : 0.7797
Specificity : 0.6395
```

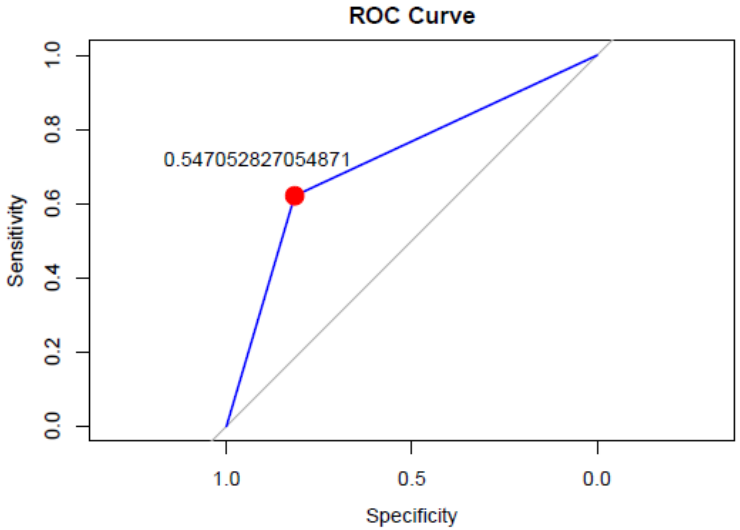


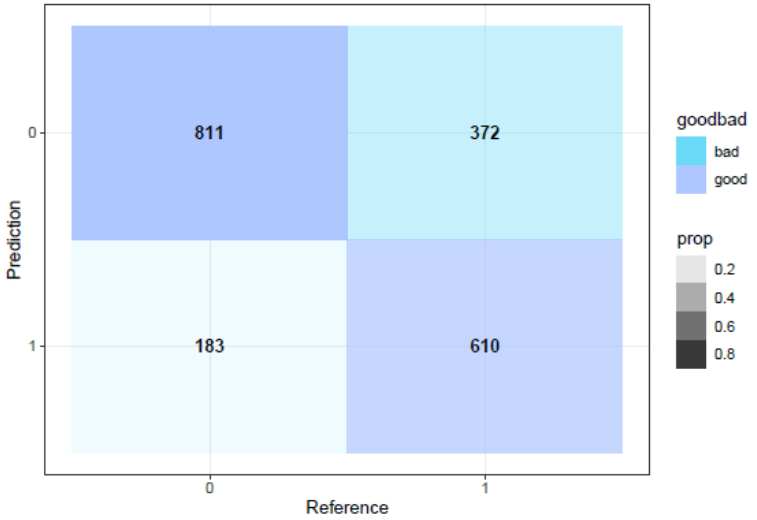
Decision Tree(All Variable)



Decision Tree(All Variable)

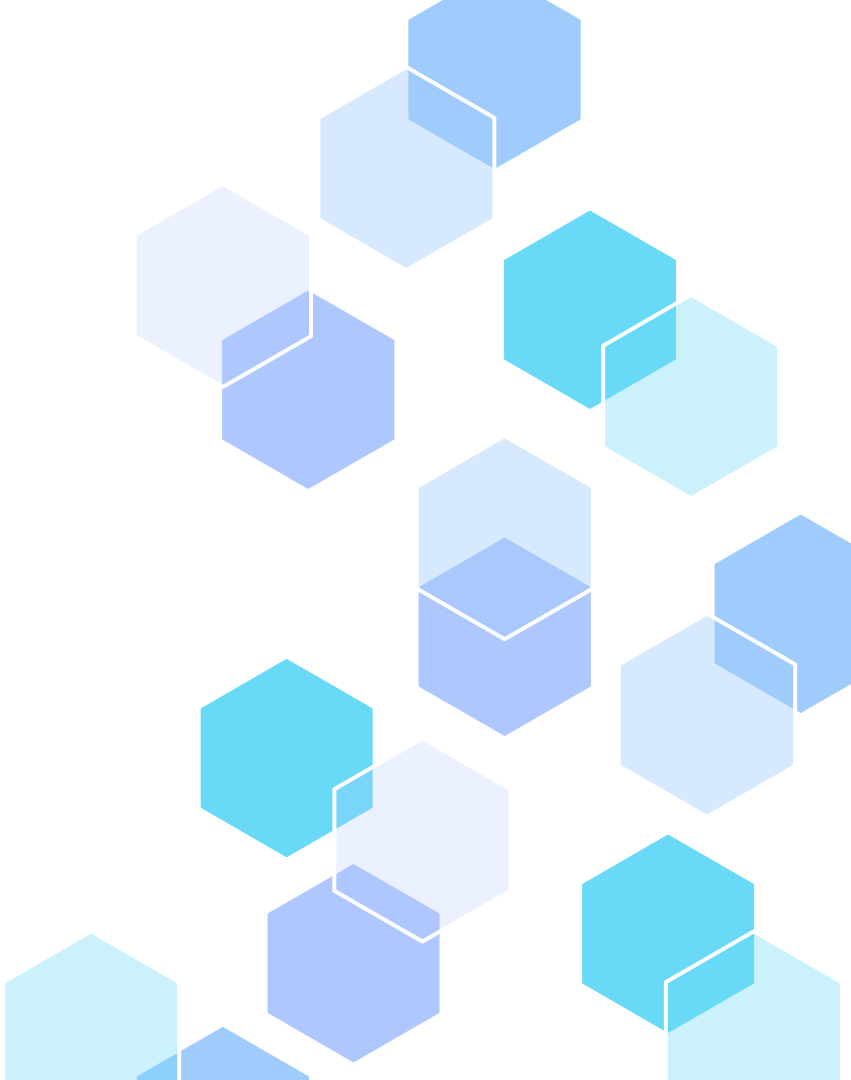
```
Accuracy : 0.7191
Sensitivity : 0.8159
Specificity : 0.6212
```





05

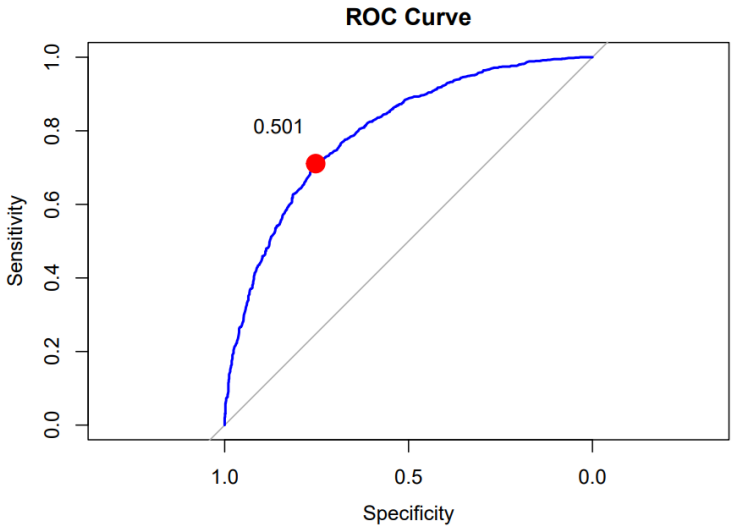
Random Forest

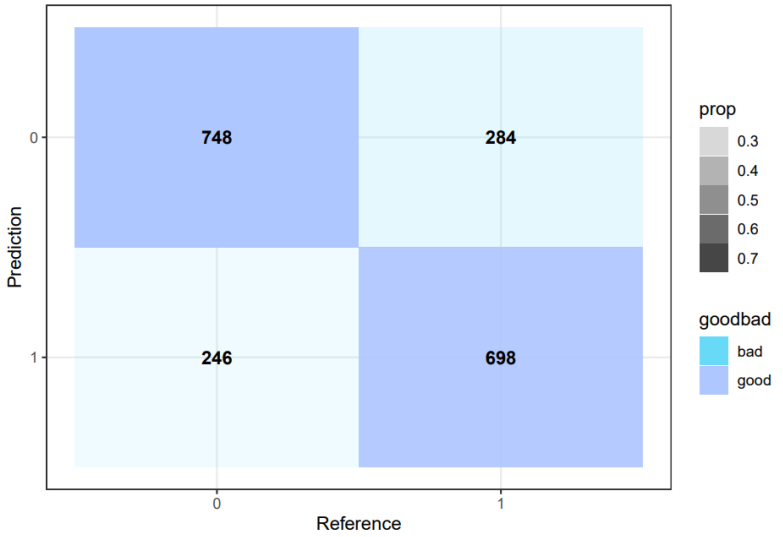


Random Forest(LASSO)

```
mtry ntree
   3   500
```

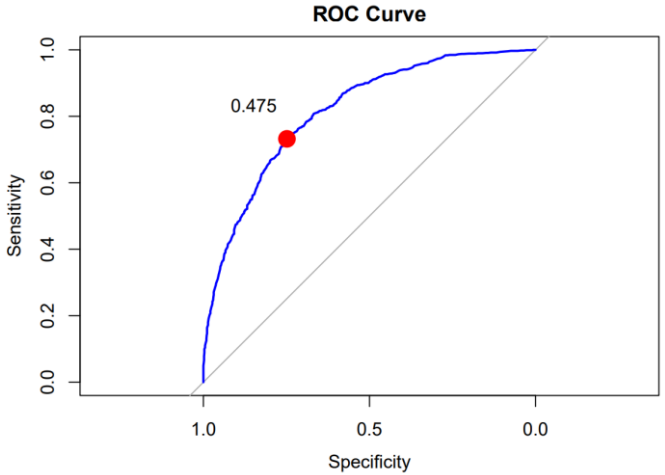
```
Accuracy : 0.7318
Sensitivity : 0.7525
Specificity : 0.7108
```



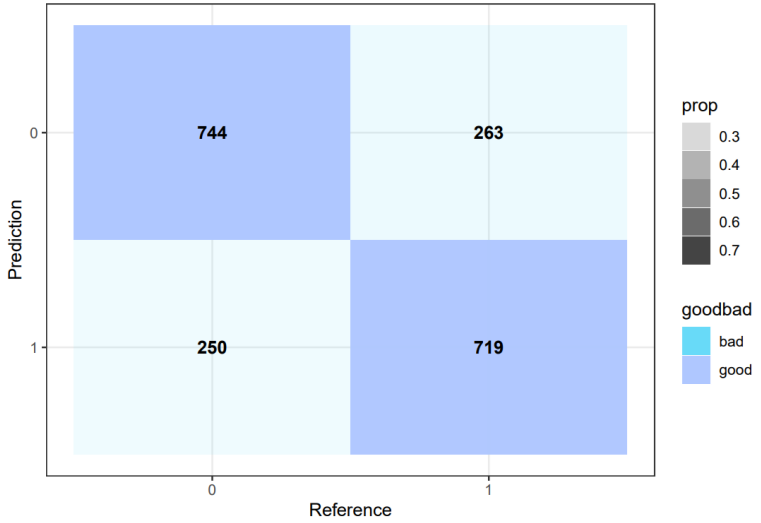


Random Forest(All Variable)

```
mtry ntree
   1   300
```

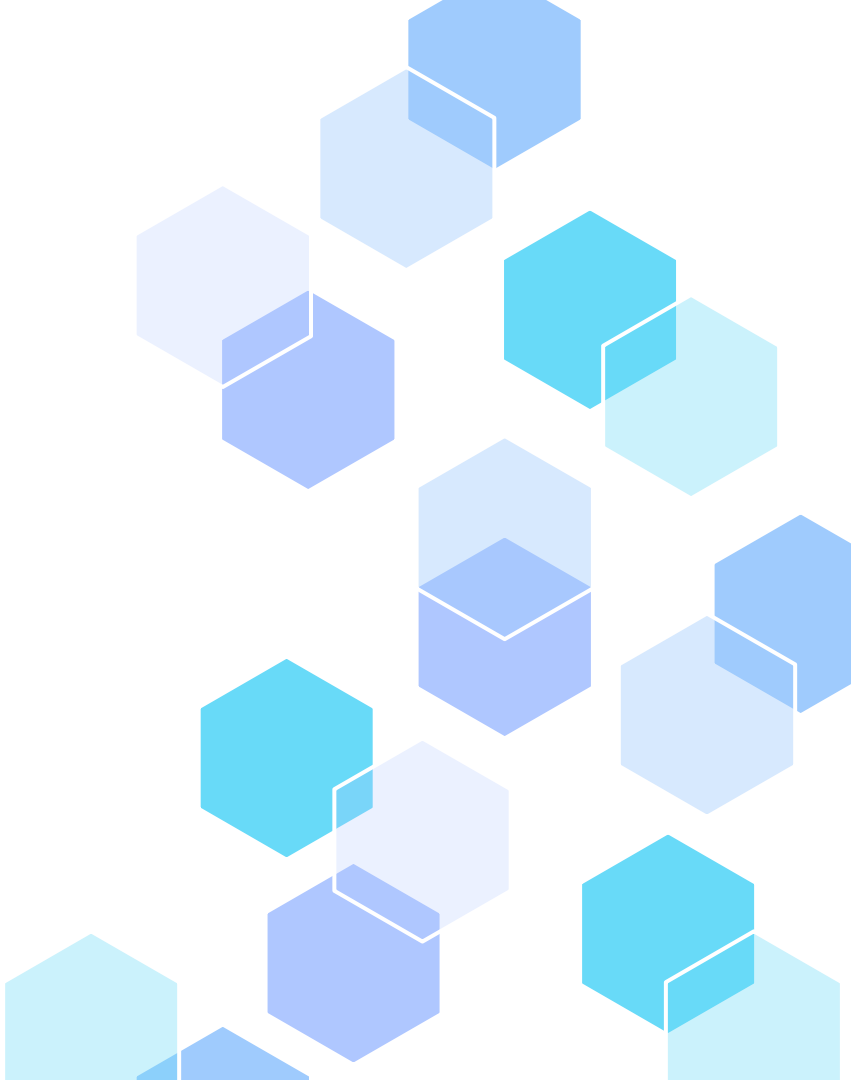



```
Accuracy : 0.7404
Sensitivity : 0.7485
Specificity : 0.7322
```



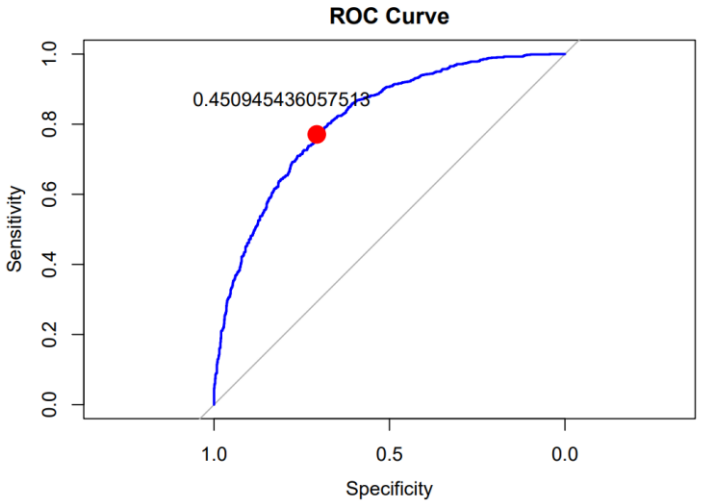
06

gboost

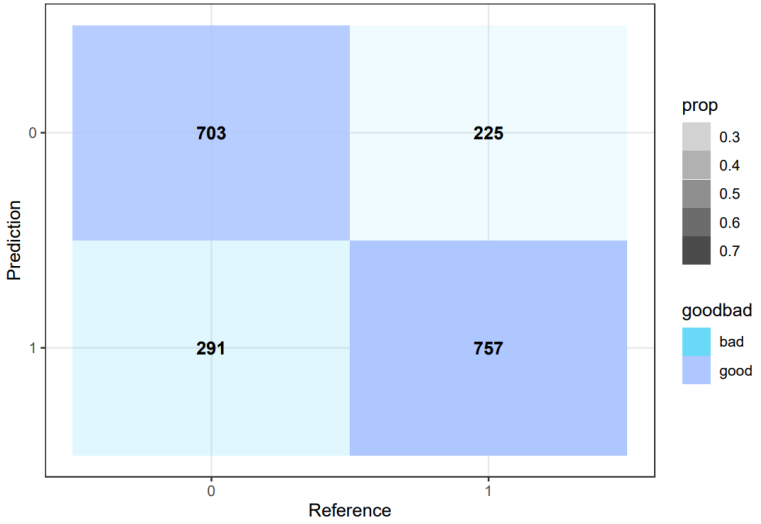


gboost(LASSO)

```
n.trees interaction.depth shrinkage n.minobsinnode
    500                 4      0.01             10
```

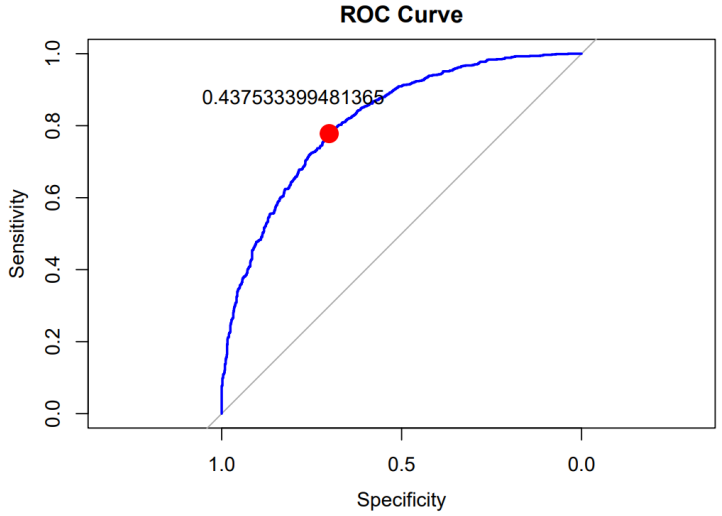



```
Accuracy : 0.7389
Sensitivity : 0.7072
Specificity : 0.7709
```

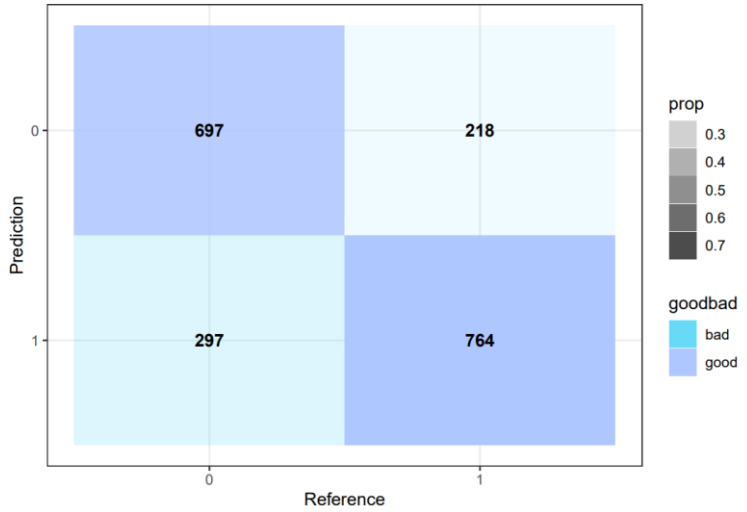


gboost(All Variable)

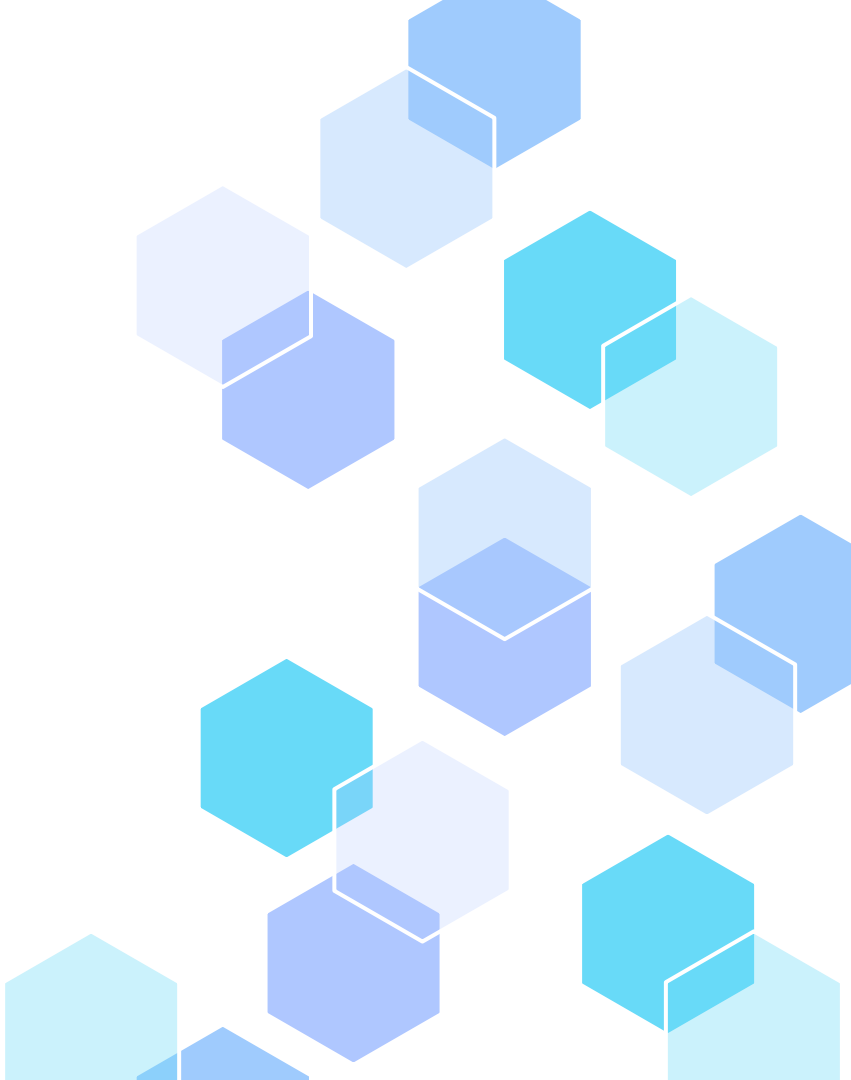
```
n.trees interaction.depth shrinkage n.minobsinnode
    500                 3      0.01             10
```




```
Accuracy : 0.7394
Sensitivity : 0.7012
Specificity : 0.7780
```



Conclusion



Conclusion

1. How to predict the outcome of a game within ten minutes of data (determining the best model).
2. How to quickly determine the outcome of a game (determining the most important variable).

	auc	sen	spe	most_important_variable
knn_sel	0.7328	0.7505	0.7149	NA
knn_all	0.5602	0.5704	0.5499	NA
logistic	0.7460	0.7072	0.7851	blueTotalGold, redTotalGold
tree_sel	0.7100	0.7797	0.6395	redTotalGold, blueTotalGold
tree_all	0.7191	0.8159	0.6212	blueGoldDiff, redGoldDiff
rf_sel	0.7318	0.7525	0.7108	redTotalGold, blueTotalGold
rf_all	0.7404	0.7485	0.7322	blueGoldDiff, redGoldDiff
gboost_sel	0.7389	0.7072	0.7709	redTotalGold, blueTotalGold
gboost_all	0.7373	0.7565	0.7179	redGoldDiff, blueGoldDiff

Reference

- https://www.kaggle.com/code/xiyuewang/lol-how-to-win/input

- https://www.rdocumentation.org/packages/caret/versions/4.47/topics/train

Thanks!