

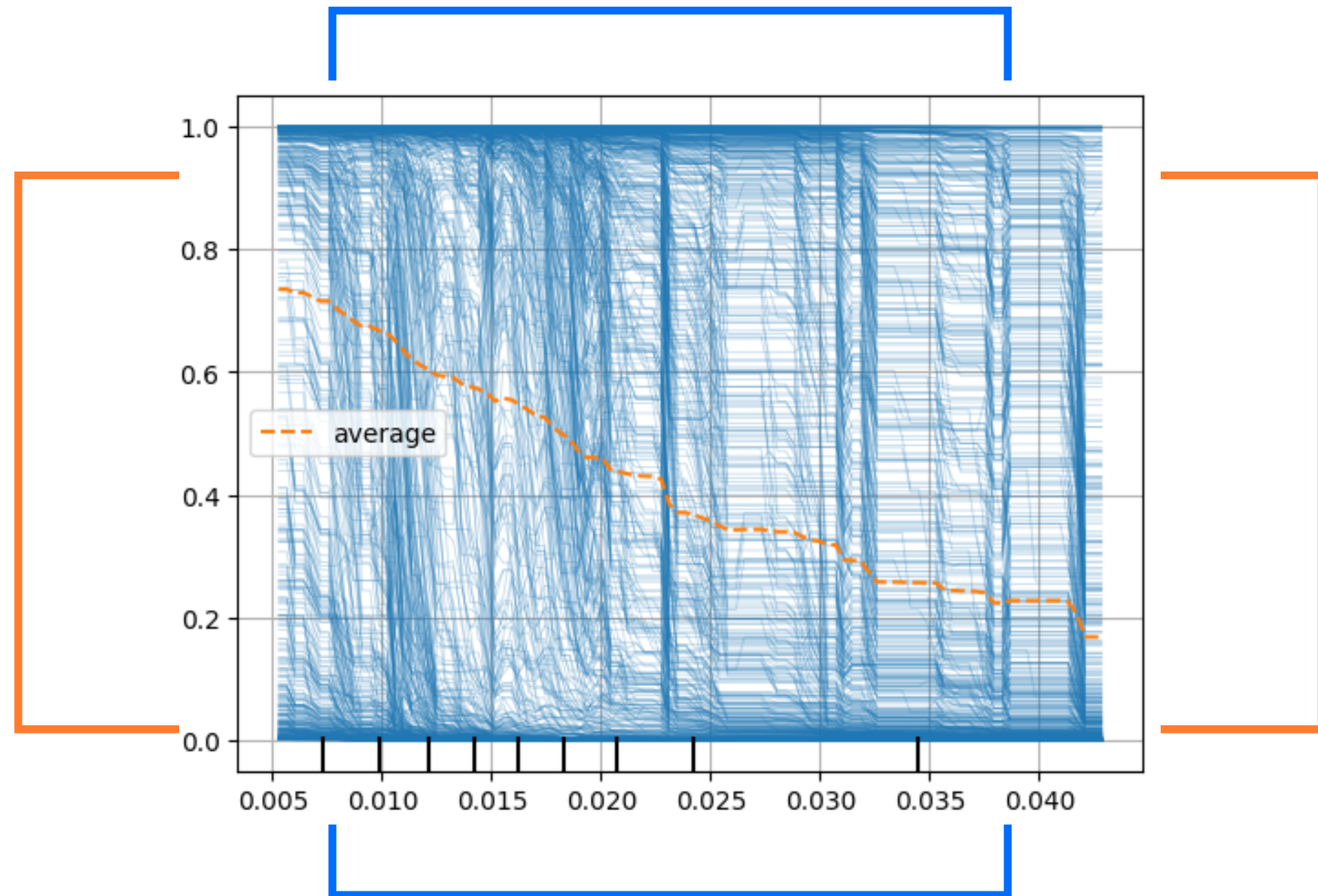
The background features a large, faint, stylized 'L' in the center, which is a common symbol for League of Legends. The 'L' is composed of a blue circle and a yellow rectangle. Surrounding the 'L' are several orange and blue rectangular outlines of varying sizes, some of which are partially cut off by the edges of the image. The text 'LEAGUE OF LEGEND' is in orange and 'LEVEL UP PLAN' is in blue, both in a bold, sans-serif font.

LEAGUE OF LEGEND LEVEL UP PLAN

REALTIME WINRATE PREDICTION OVERLAY

실시간 승률 예측 오버레이

- 리그 오브 레전드 클라이언트에서 제공하는 오버레이 형식 통계 제공 서비스
- 랭크 게임의 레거시 데이터를 토대로 학습시킨 모델을 기준으로 실시간 팀 스코어를 평가

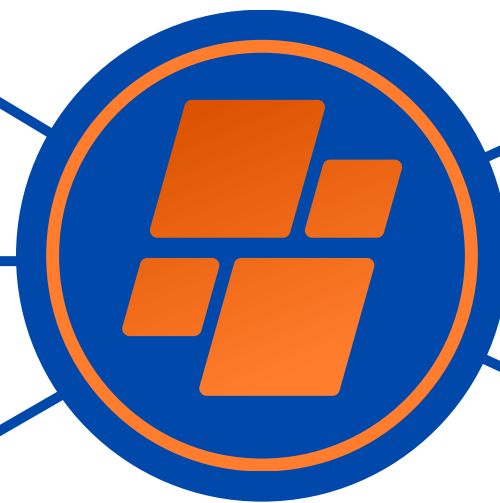


목차

프로그램 필요성

데이터셋 소개

가설 설정



모델 설정

모델 해석

한계점과 개선사항

프로그램 필요성

1

핵/헬퍼/치트 견제

모델이 학습한 데이터를 토대로
유저를 평가하여
비정상적인 플레이에 대한
합리적 의심을 끌어낼 수 있음

2

유저의 외부프로그램
사용 욕구 감소

롤 블리츠 등
교묘하게 규제를 벗어나는
프로그램들에 대한
유저의 필요 및 사용 욕구를
방지할 수 있음

3

외부 프로그램
규제 강화의 근거

롤 블리츠 등
교묘하게 규제를 벗어나는
프로그램들에 대한
제재를 확대할 경우,
유저들의 반발 감소 가능

데이터셋 소개

샘플 모델을 제작하기 위해
리그 오브 레전드의 상위 3개 티어(마스터, 그랜드마스터, 챌린저)의
2020년 랭크게임 데이터 191,000 게임을 토대로 데이터셋을 생성함
(챌린저 26000 게임, 그랜드마스터 65000 게임, 마스터 100000 게임)

◆ 중복치 제거

◆ blue팀과 red팀 데이터를 분리

◆ 오타가 있는 column명 변경

◆ 결측치 없음

◆ duration과 비례하는 데이터 처리

◆ Shape : (145740, 50)

데이터셋 소개

binary categorical data

Wins
FirstBlood
FirstTower
FirstBaron
FirstDragon
FirstInhibitor

Wins를 타겟으로 설정

non-binary categorical data

gameId

전부 삭제

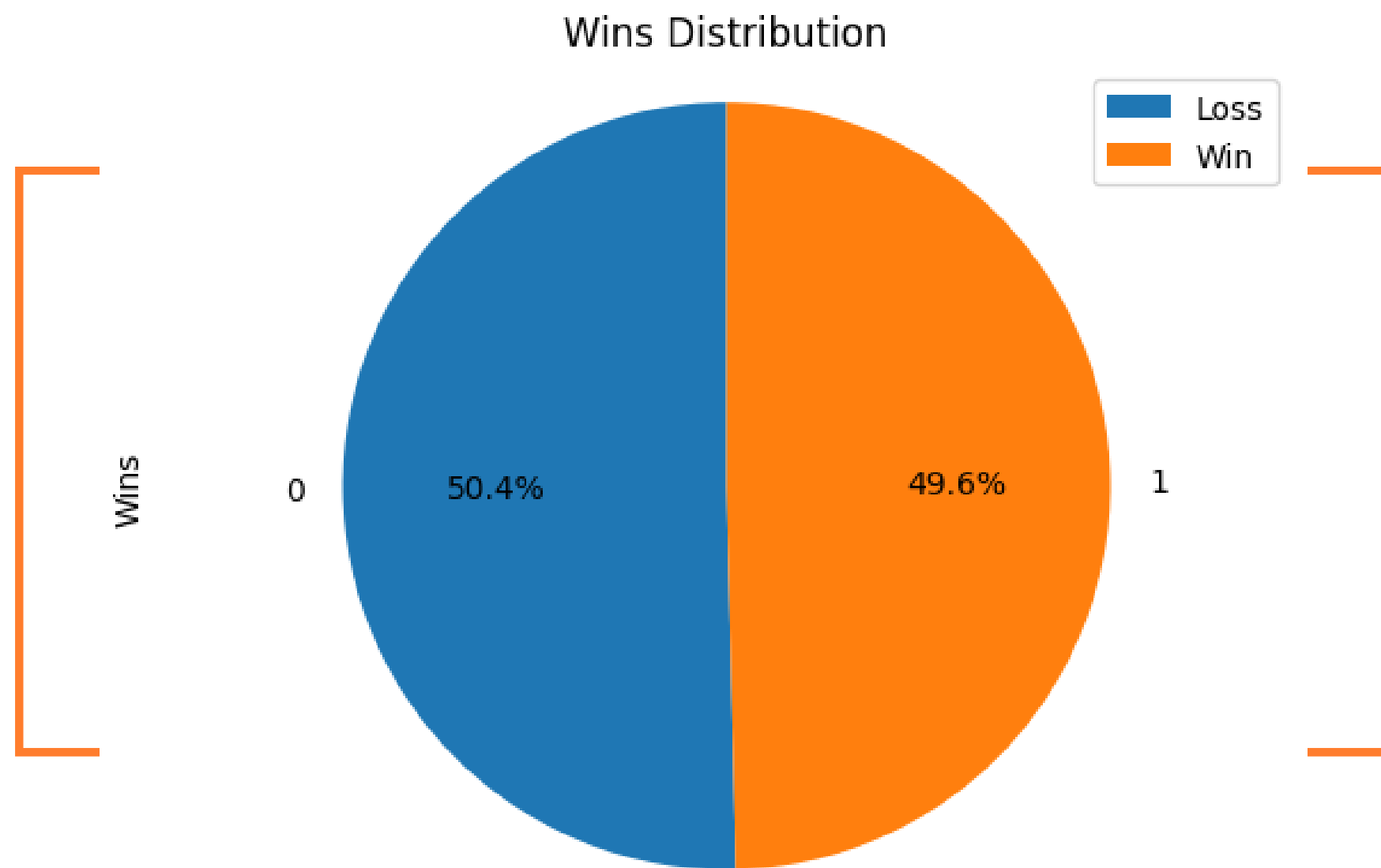
numerical data

gameDuration WardPlaced
Wardkills Kills Death Assist
ChampionDamageDealt
TotalGold TotalMinionKills
TotalLevel AvgLevel
JungleMinionKills
KillingSpree TotalHeal
ObjectDamageDealt

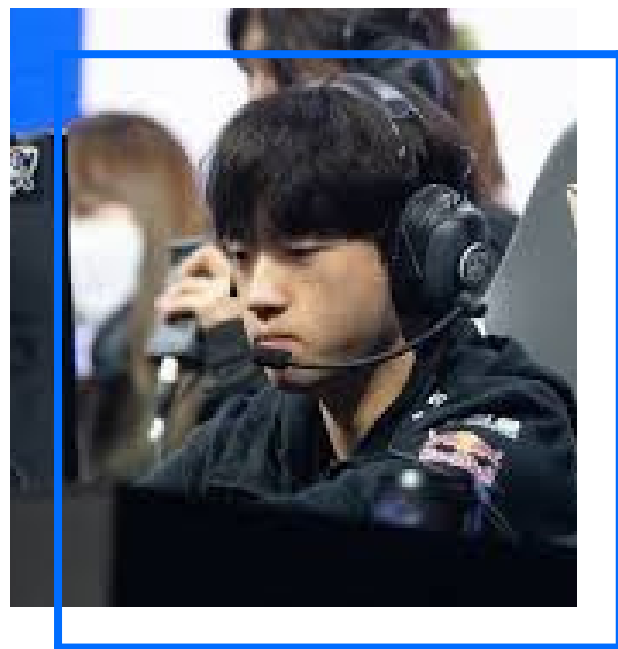
level 관련 데이터 통합

데이터셋 소개

Target으로 사용한 Columns인
Wins의 분포를 확인한 결과
분포가 거의 5:5로
불균형 문제가 거의 존재하지 않음을
확인했음

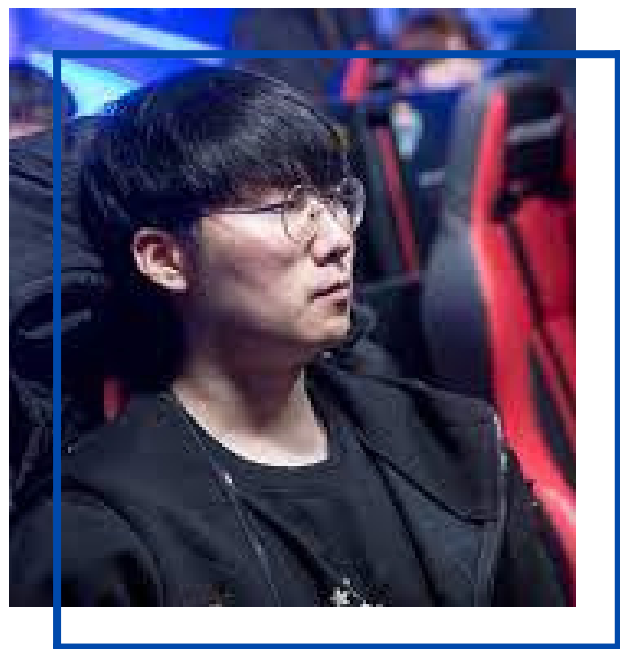


가설 설정



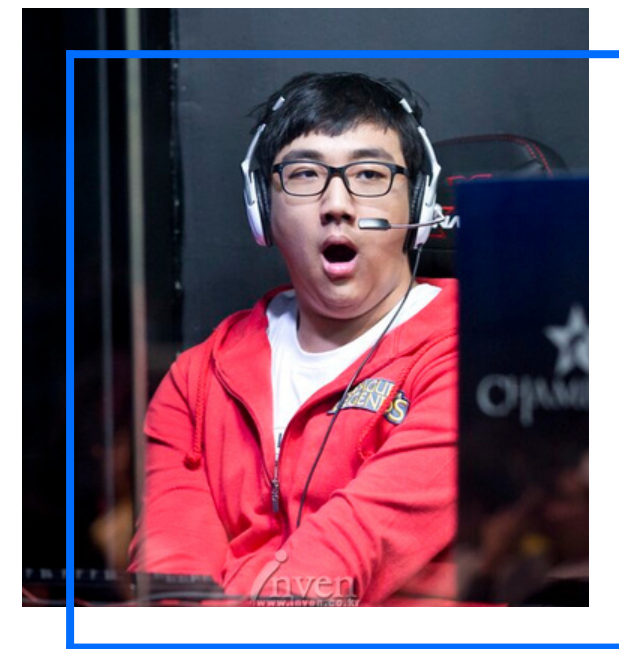
가설 1

팀원 데스 수가 적을수록
승리할 확률이 높을 것이다.



가설 2

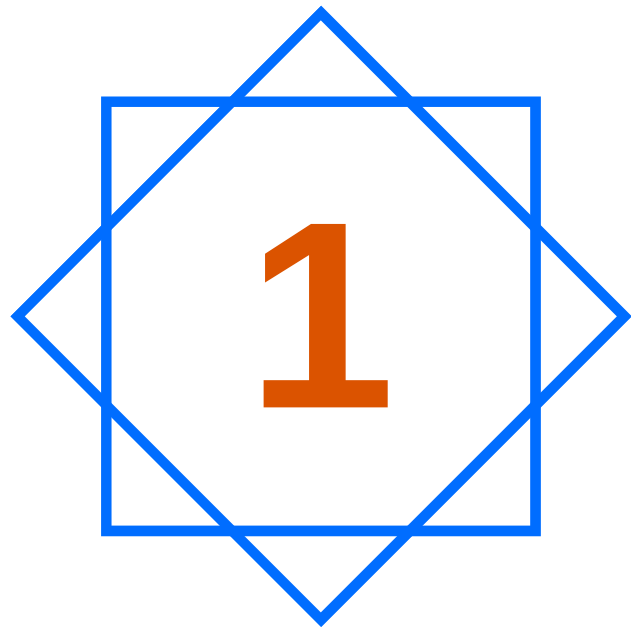
팀원 킬 수가 높을수록
승리할 확률이 높을 것이다.



가설 3

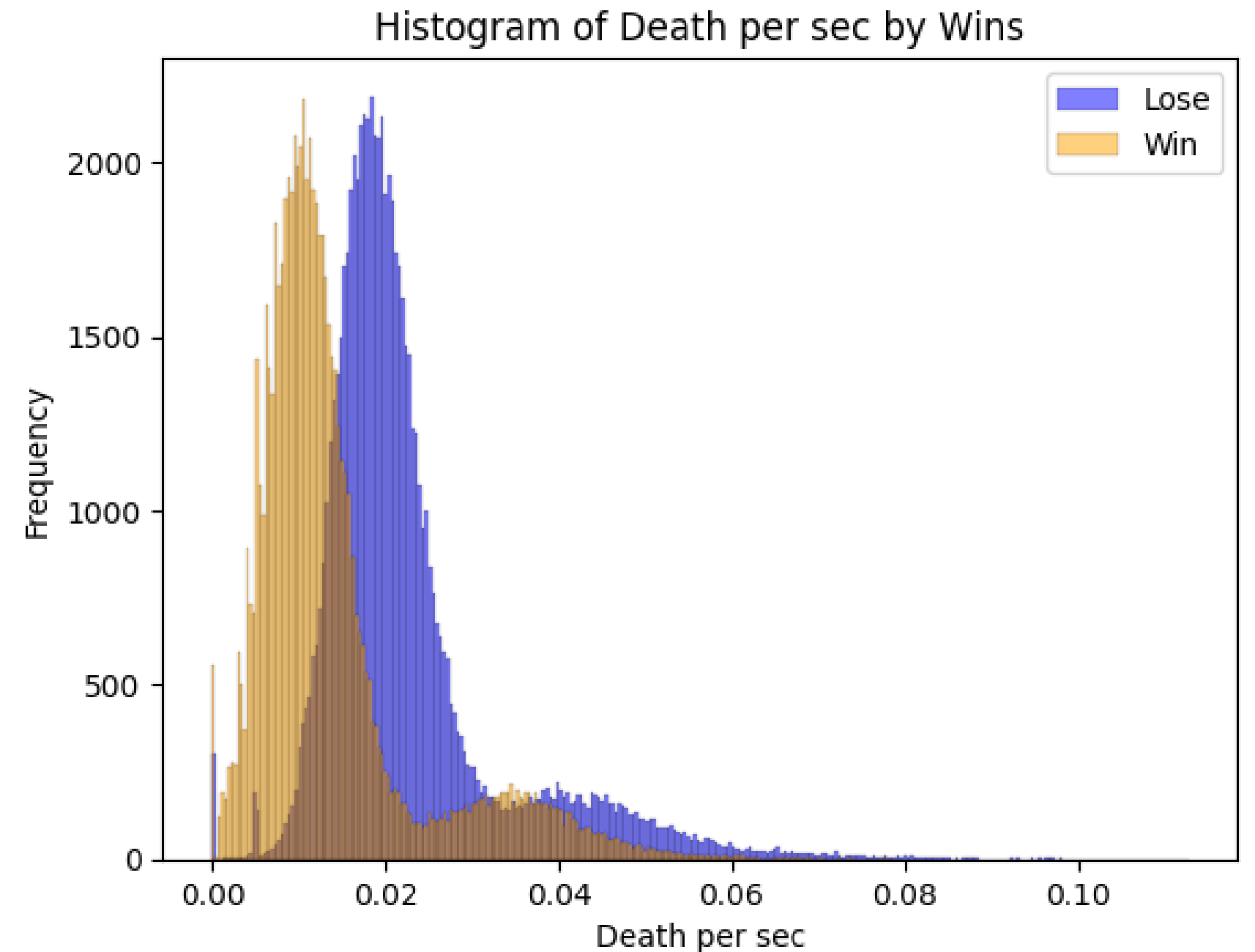
팀의 총 골드량이 많을수록
승리할 확률이 높을 것이다.

가설 설정



팀원 데스 수가 적을수록
승리할 확률이 높을 것이다.

히스토그램으로 확인한 결과
패배시의 그래프가 승리시보다
초당 데스 횟수가 앞선 것을 확인할 수 있었다.
가설 1의 내용이 일부 맞는 것으로 확인

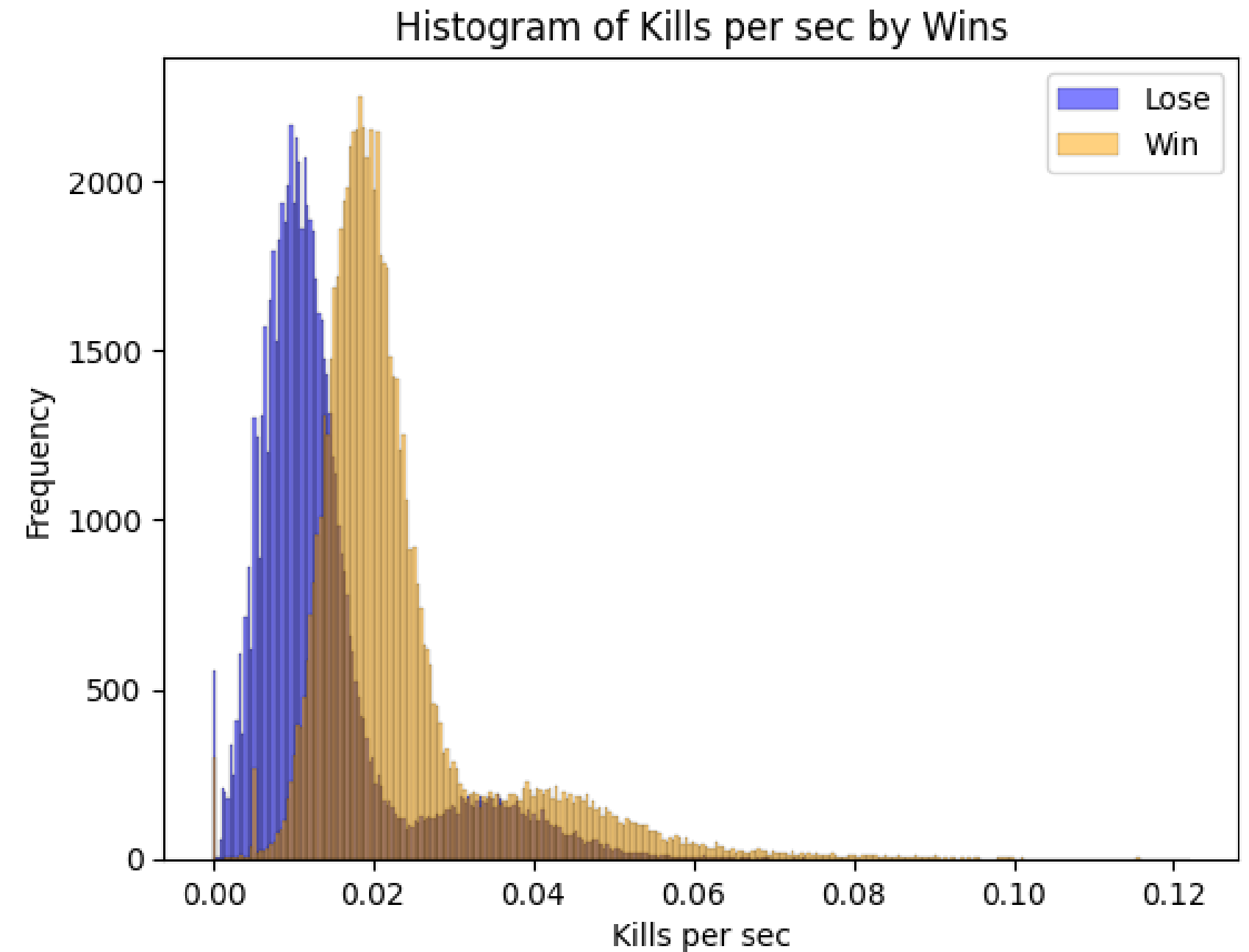


가설 설정



팀원 킬 수가 높을수록
승리할 확률이 높을 것이다.

히스토그램으로 확인한 결과
패배시의 그래프가 승리시보다
초당 킬 횟수가 뒤쳐진 것을 확인할 수 있었다.
가설 2의 내용이 일부 맞는 것으로 확인

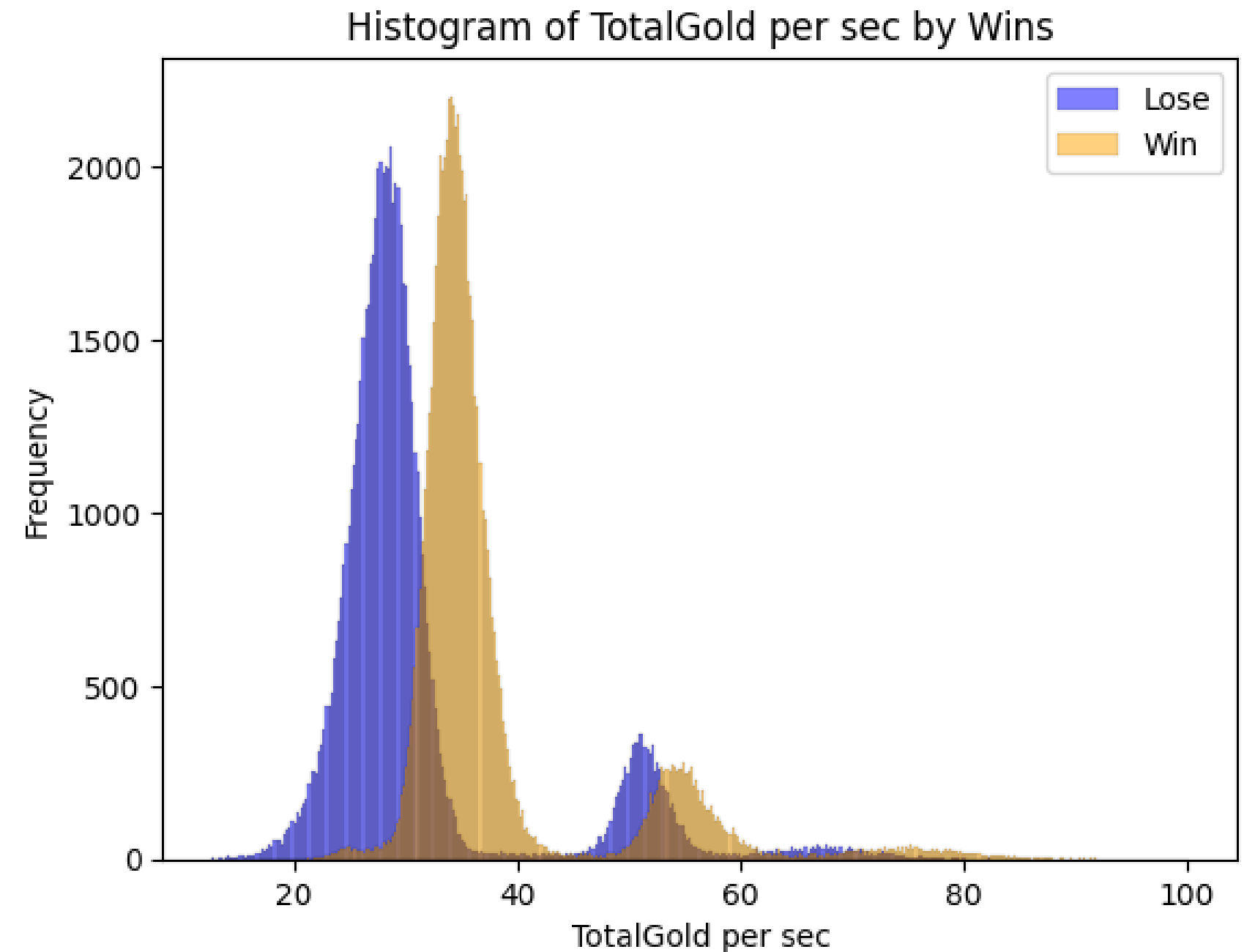


가설 설정



팀의 총 골드량이 많을수록
승리할 확률이 높을 것이다.

히스토그램으로 확인한 결과
패배시의 그래프가 승리시보다
초당 골드 획득수가 뒤쳐진 것을 확인할 수 있었다.
가설 3의 내용이 일부 맞는 것으로 확인



모델 설정

TRAIN TEST SPLIT

```
y_train = bluegame['Wins']  
x_train = bluegame.drop('Wins', axis = 1)  
y_test = redgame['Wins']  
x_test = redgame.drop('Wins', axis = 1)  
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, random_state=42, stratify=y_train)
```

blue팀 데이터를 훈련과 검정에 활용
red팀 데이터는 통째로 테스트에 활용

모델 선택

```
dtc = DecisionTreeClassifier(random_state=42)  
rfc = RandomForestClassifier(random_state=42)  
xgb = XGBClassifier(random_state=42)  
dtc.fit(x_train, y_train)  
rfc.fit(x_train, y_train)  
xgb.fit(x_train, y_train)
```

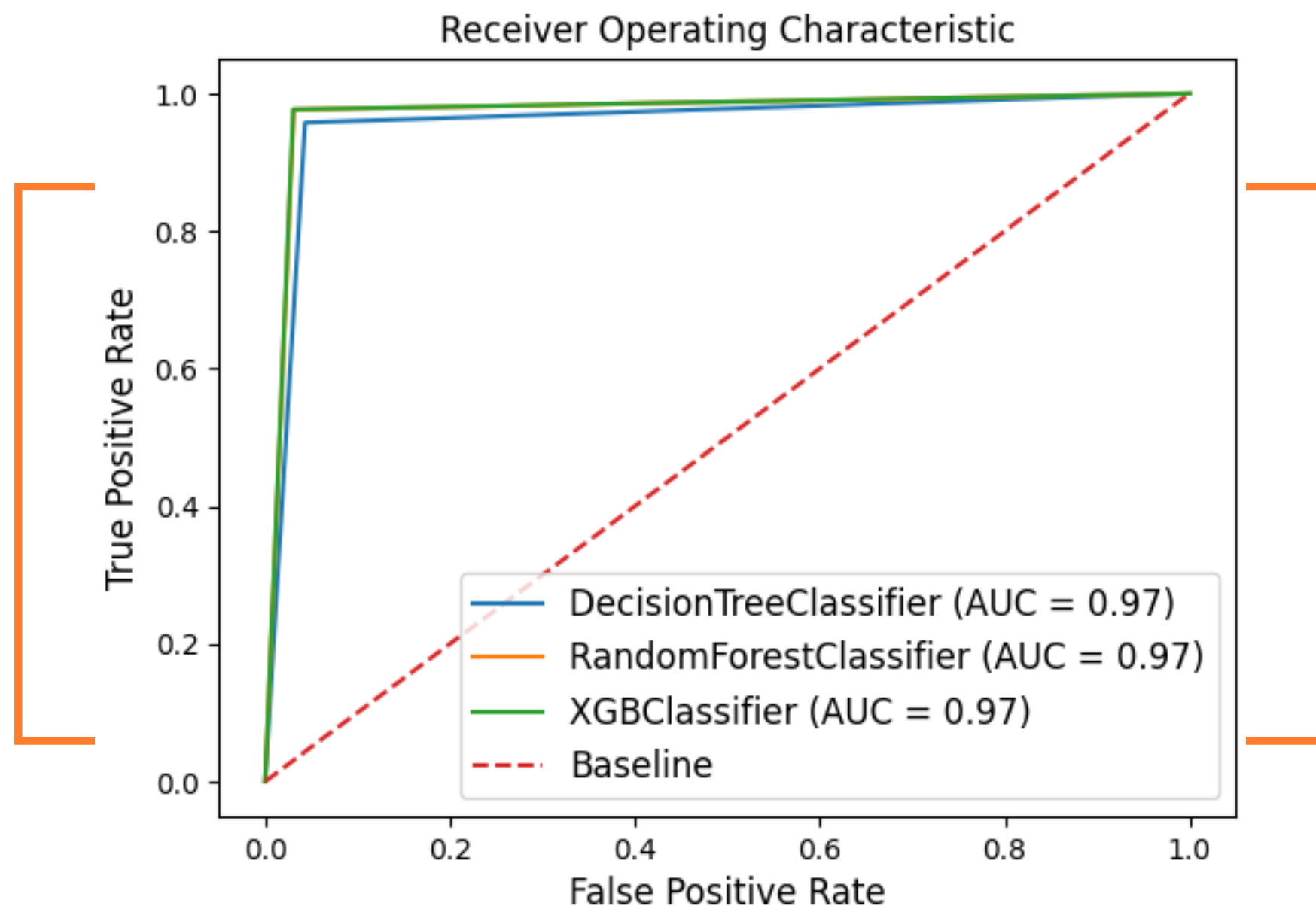
세 가지 분류 모델을 모두 적용 후 비교
(DecisionTreeClassifier, RandomForestClassifier,
XGBClassifier)

모델 설정

선행 연구나 모델이 없었기 때문에
모든 예측 확률을 0으로 설정한
베이스라인을 설정함.

ROC-Curve로 살펴보았을 때,
세 모델 모두 성능에
큰 차이는 없었음.

모델 성능이 과도하게 높아
다른 검증도 병행함.



모델 설정

앞서 분리해두었던 검정셋으로
모델 검정을 실행한 결과
세 모델 모두
큰 오류 없이 적합하게 학습되었으며
과적합도 의심되지 않음

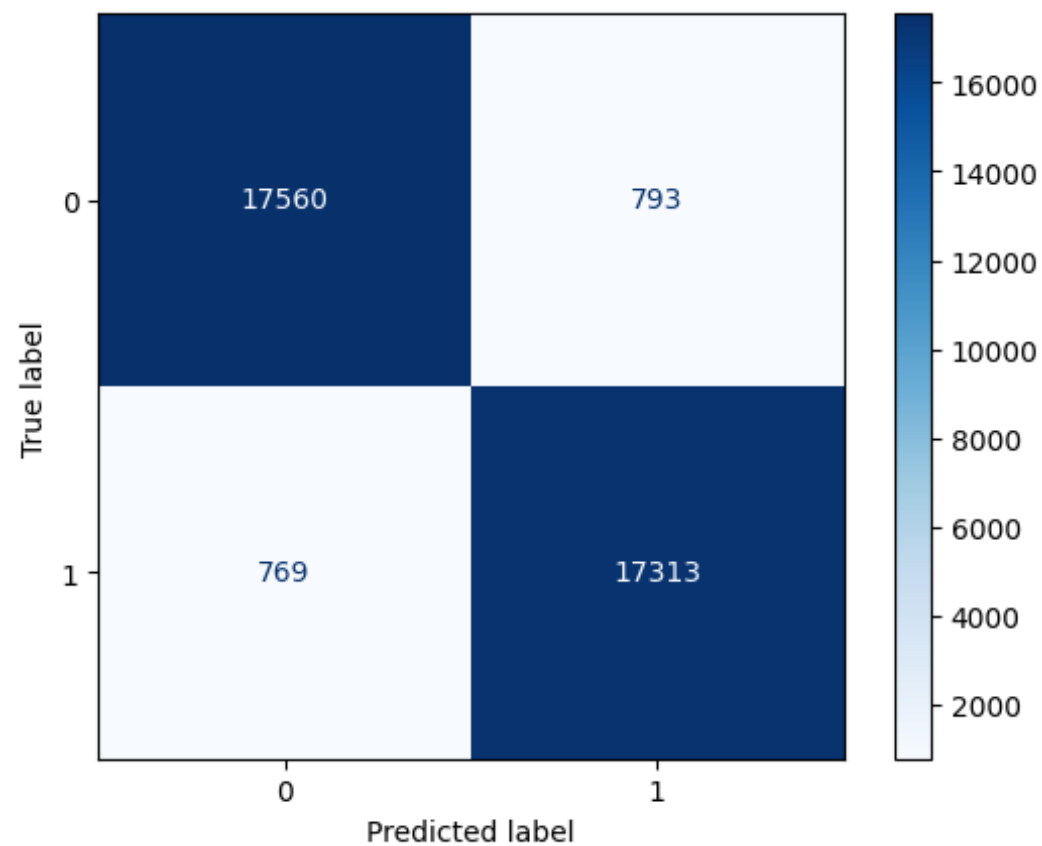
DecisionTreeClassifier	precision	recall	f1-score	support
0	0.96	0.96	0.96	18353
1	0.96	0.96	0.96	18082
accuracy			0.96	36435
macro avg	0.96	0.96	0.96	36435
weighted avg	0.96	0.96	0.96	36435

RandomForestClassifier	precision	recall	f1-score	support
0	0.98	0.97	0.97	18353
1	0.97	0.98	0.97	18082
accuracy			0.97	36435
macro avg	0.97	0.97	0.97	36435
weighted avg	0.97	0.97	0.97	36435

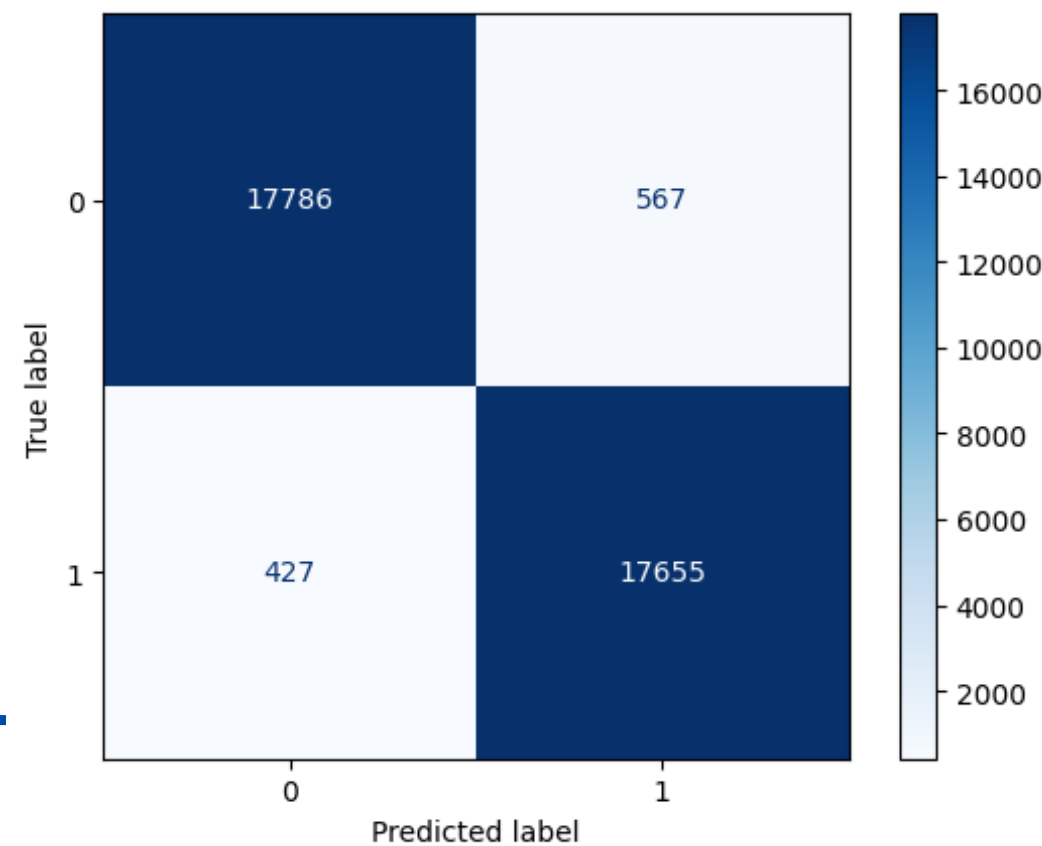
XGBClassifier	precision	recall	f1-score	support
0	0.98	0.97	0.97	18353
1	0.97	0.98	0.97	18082
accuracy			0.97	36435
macro avg	0.97	0.97	0.97	36435
weighted avg	0.97	0.97	0.97	36435

모델 설정

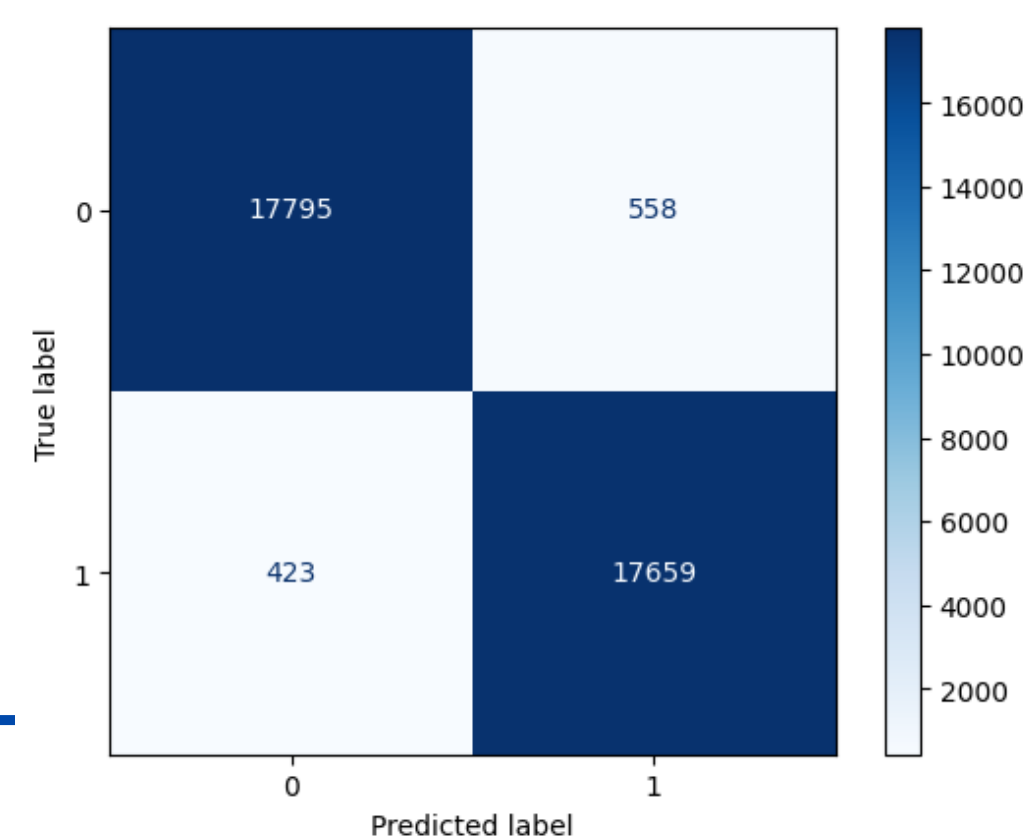
DecisionTreeClassifier



RandomForestClassifier



XGBClassifier



세 모델 모두 좋은 학습 결과를 보여주었지만,
가장 낮은 1종 및 2종 오류를 보이는
XGBClassifier를 최종적으로 채택

모델 설정

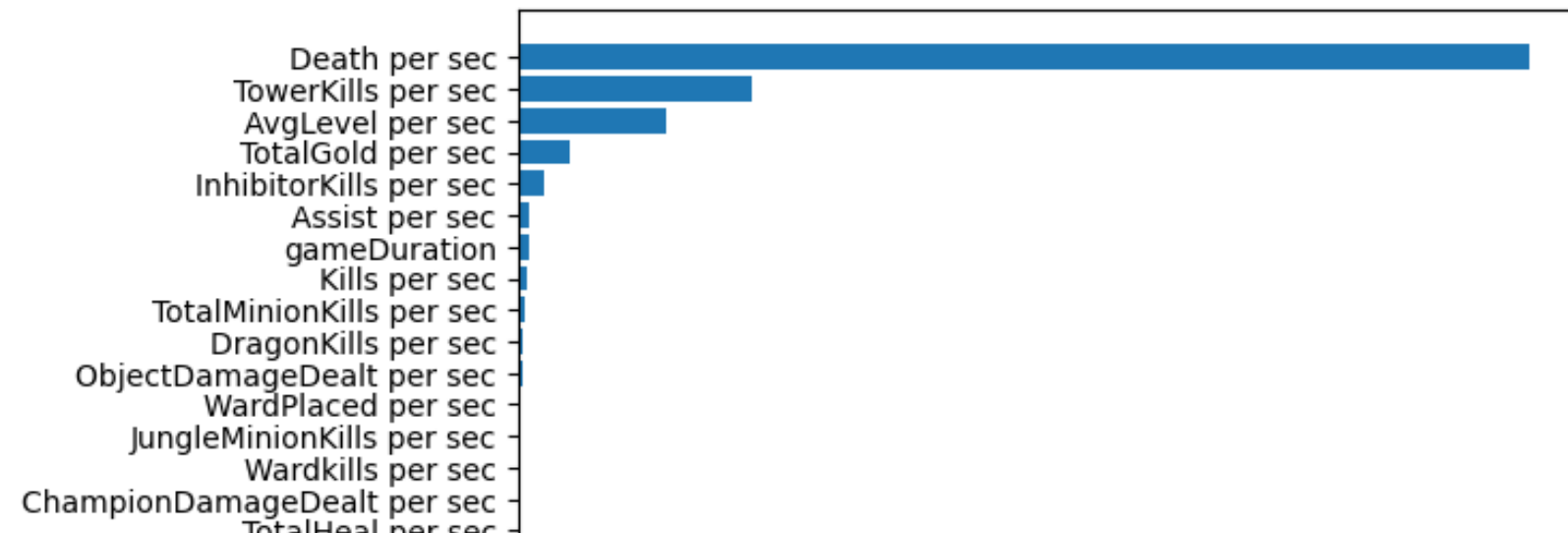
테스트셋으로
모델의 최종 성능을 확인한 결과
모델은 충분히 일반화되었음을
알수 있음.

```
y_pred = xgb.predict(x_test)
print('XGBClassifier', classification_report(y_test, y_pred))
```

XGBClassifier		precision	recall	f1-score	support
	0	0.98	0.97	0.97	72327
	1	0.97	0.98	0.98	73413
	accuracy			0.98	145740
	macro avg	0.98	0.98	0.98	145740
	weighted avg	0.98	0.98	0.98	145740

모델 해석

Feature Importance Graph



원본 데이터의 크기가 거대해서
Permutation Importance로 측정.
학습에 있어 영향력이 미미한 feature들도
남겨두더라도 정확도가 높아
별도로 drop하지는 않음.

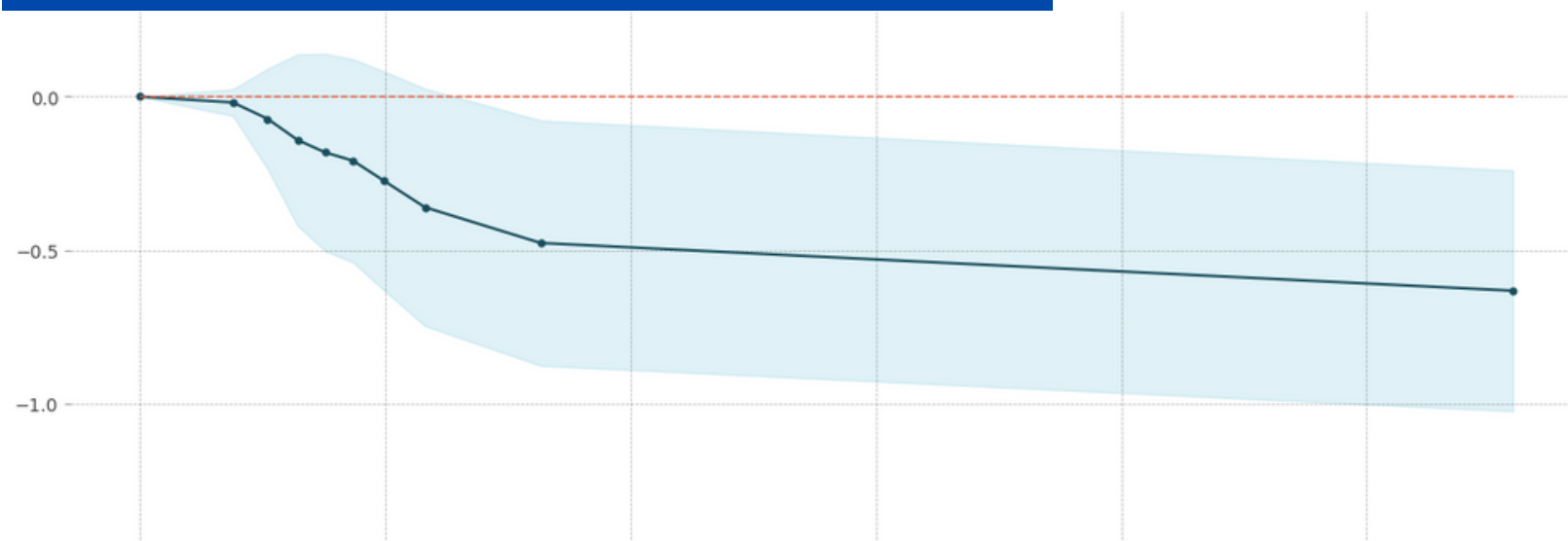
Table of Feature Importance

Weight	Feature
0.0774 ± 0.0013	Death per sec
0.0180 ± 0.0009	TowerKills per sec
0.0114 ± 0.0002	AvgLevel per sec
0.0040 ± 0.0003	TotalGold per sec
0.0020 ± 0.0001	InhibitorKills per sec
0.0009 ± 0.0001	Assist per sec

팀 플레이어가 초당 평균 몇회 사망했는지를 나타내는
Death per sec와
게임 내내 초당 평균 몇회 타워를 파괴했는지를 나타내는
TowerKills per sec가
가장 높게 나타났음

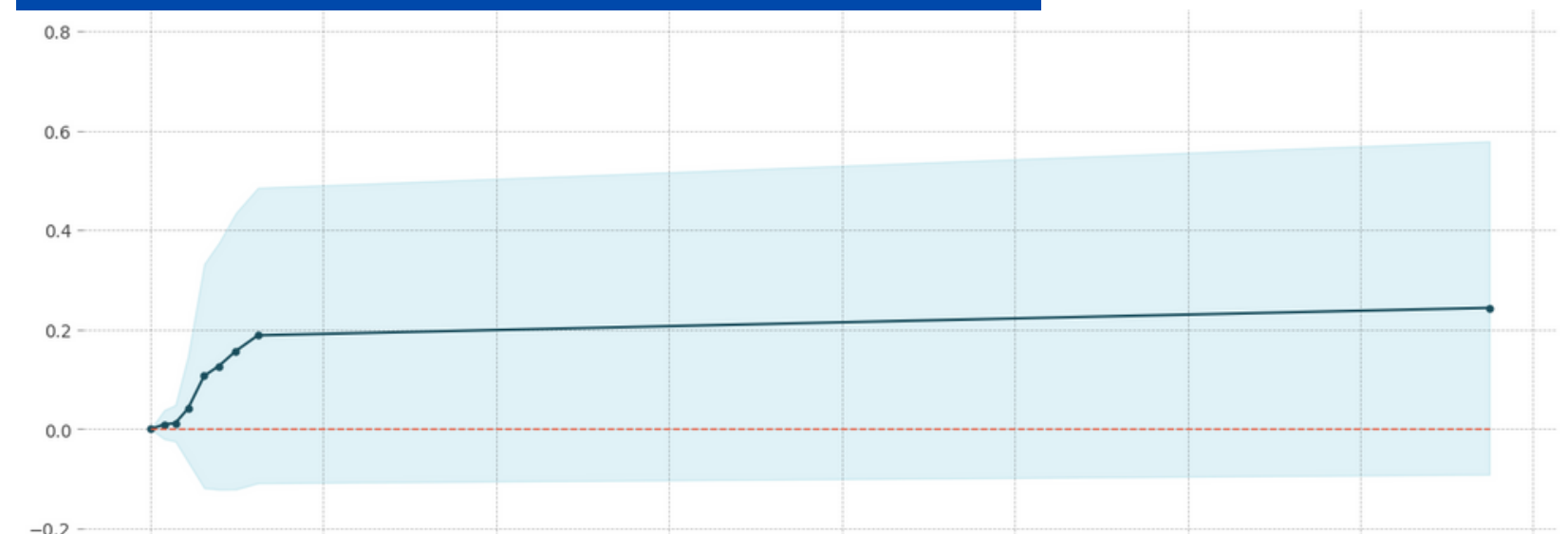
모델 해석

Death per sec - PDP



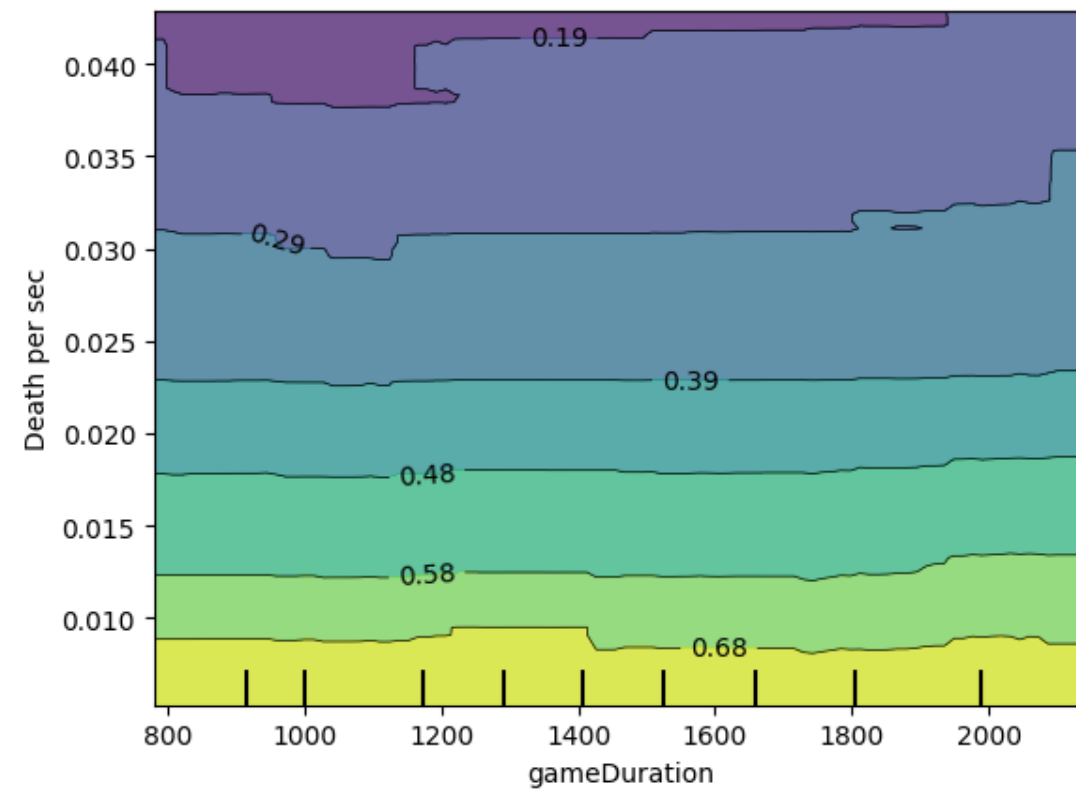
PDP 그래프를 그린 결과
Death per sec의 값이 증가함에 따라
지속적으로 감소하는 것을 통해
타겟과 음의 상관관계를 갖고 있다는 것을
알수 있음

TowerKills per sec - PDP



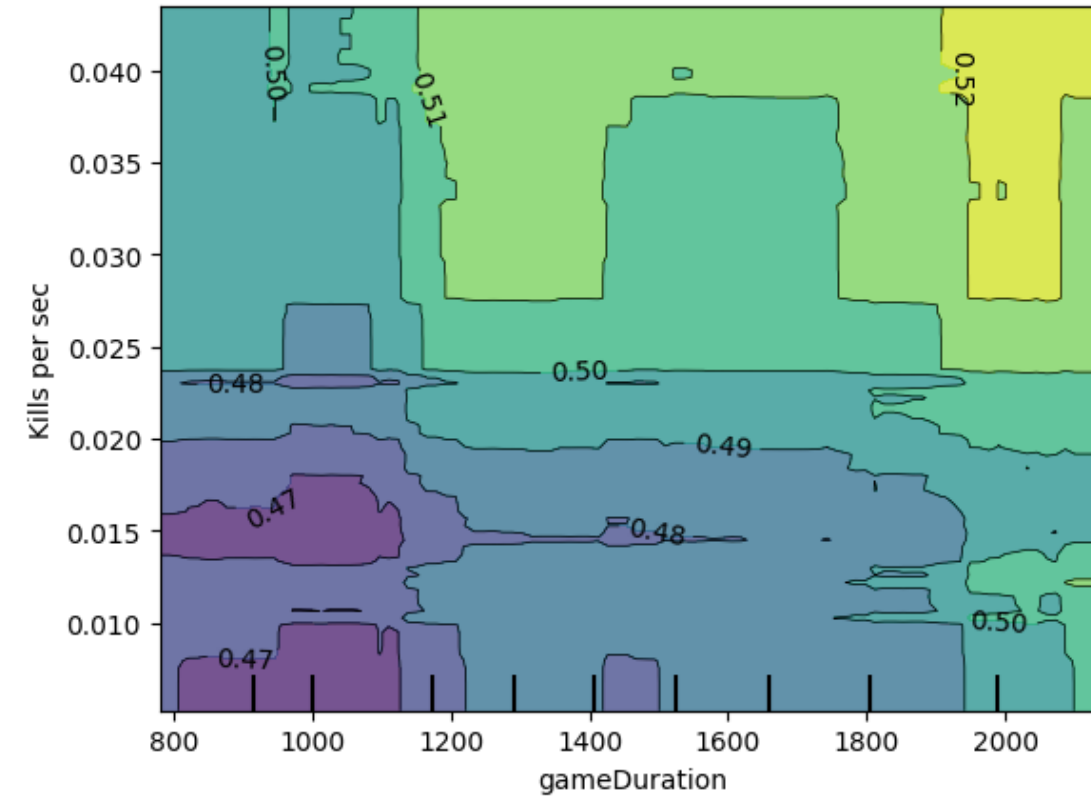
PDP 그래프를 그린 결과
TowerKills per sec의 값이 증가함에 따라
지속적으로 증가하는 것을 통해
타겟과 양의 상관관계를 갖고 있다는 것을
알수 있음

모델 해석



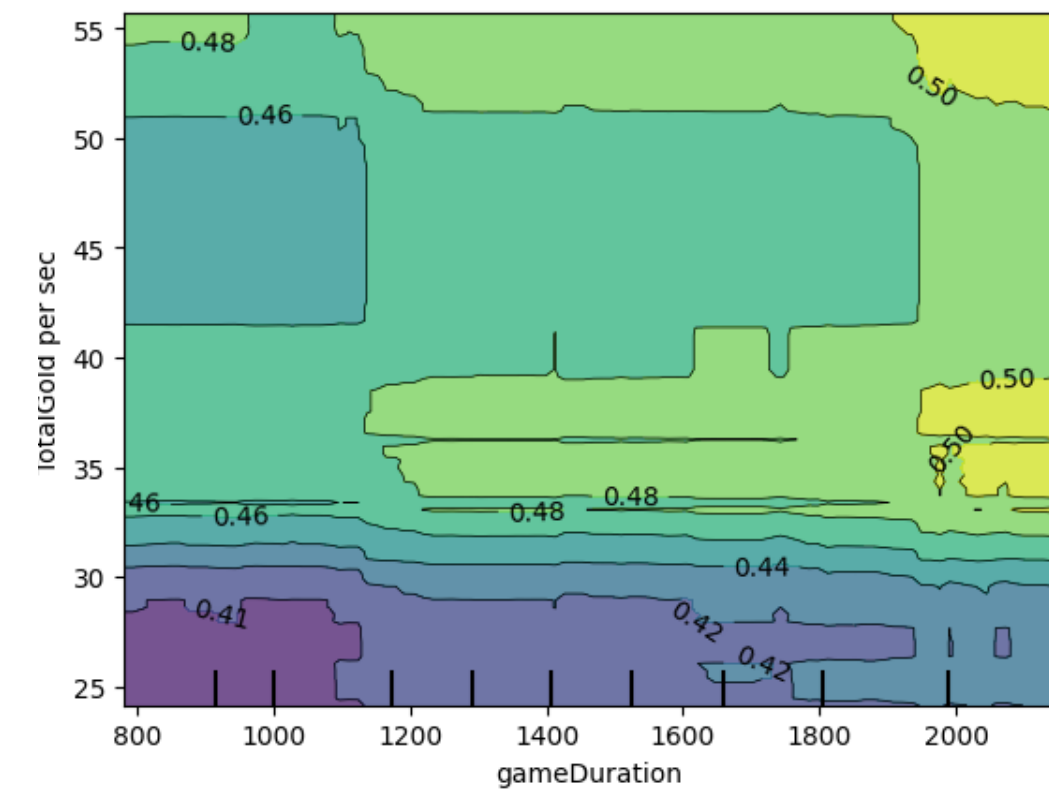
DEATH PER SEC

Analysis of PDP
based on gameDuration



KILLS PER SEC

Analysis of PDP
based on gameDuration



TOTALGOLD PER SEC

Analysis of PDP
based on gameDuration

한계점과 개선사항

- ◆ 데이터셋의 한계
- ◆ 공정성 문제
- ◆ 티어별 솔루션 제공
- ◆ 팀단위 → 개인단위 서비스 확대



OUR TOP SERVICES

1

데이터셋의 한계

- 상위 티어만 집계
- 2020년 데이터

2

공정성 문제

- 프로경기에 사용될 가능성
- 미사용자와 사용자와의 격차
- 반강제되는 사용

3

개선사항

- 티어에 따라 가중치 부여
- 팀단위에서 개인단위로도 서비스 확장
- 공정성에 관한 지속적인 논의

The background features several overlapping rectangles in blue and orange. A large orange rectangle is centered behind the text. Other blue and orange rectangles are scattered around, some overlapping each other and the central orange one. The text "THANK YOU" is centered in a large, bold, black, sans-serif font.

THANK YOU