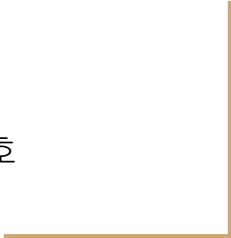


PUBG Top10 예측

PartyQuest

김석연, 임예원, 이광호



CONTENTS

1. 분석목표
2. 팀원/환경 소개
3. Dataset 설정
 - 파생변수
 - Target 및 Feature 설정
4. Modeling
 - Random Forest
 - Hyper parameter Tuning
 - 분류 모형 평가 지표
 - Model Performance
 - Feature importance
 - Dimensionality Reduction
5. 결론

1. 분석 목표

분석목표

“ 유저들의 게임 데이터를 통해
top10 (squad 기준 top3)에 들어갈 수 있을지 예측 ”

2. 팀원/환경 소개

팀원 / 환경 소개

모델링

김석연 ■■■■□
임예원 ■■■■□
이광호 ■■■■□

발표

김석연 ■■■■□
임예원 ■■■■□
이광호 ■■■■□

전처리

김석연 ■■■■□
임예원 ■■■■□
이광호 ■■■■□

작업 환경 : Python 3.8

3. Dataset 설정

Dataset 설정

Aggregate dataset

- 해당 게임(match)에 대한 종합적인 정보를 담고 있음

columns	설명	type
date	게임 시작 날짜	datetime
game_size	게임 참여 인원	int
match_id	해당 게임 고유 식별 id	object
party_size	1인/2인/4인 플레이 게임	int
player_assists	팀원을 살려준 횟수	int
player_dbno	기절했던 횟수	int
player_dist_ride	탈것을 활용한 이동 거리	float

columns	설명	type
player_dist_walk	도보 이동 거리	float
player_dmg	적에게 데미지를 입힌 정도	float
player_kills	킬 횟수	int
player_name	플레이어 game id	object
player_survive_time	생존 시간	float
team_id	팀 고유 id	object
team_placement	해당 매치의 등수	int

Dataset 설정

Death dataset

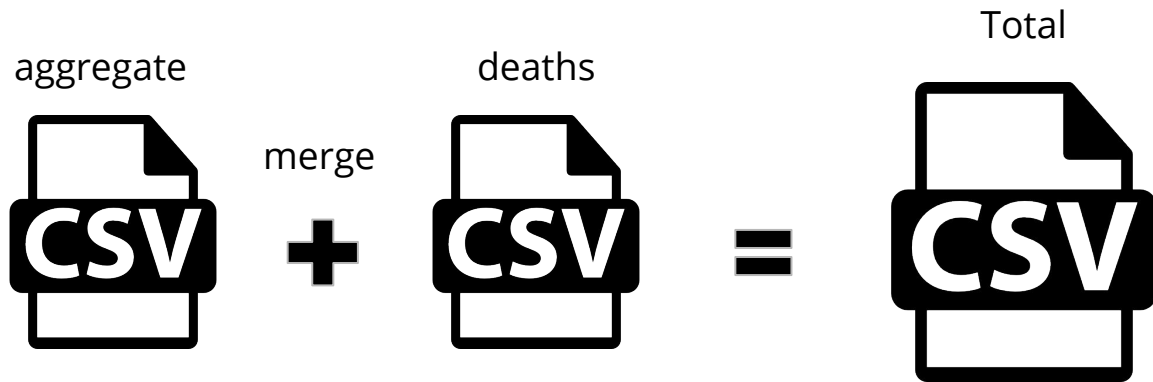
- 해당 게임(match)에서 발생한 Kill에 대한 정보를 담고 있음

columns	설명	type
killed_by	사살 당한 무기 혹은 사고 원인	object
killer_name	killer game id	object
killer_placement	kill 한 플레이어의 등수	int
killer_position_x	killer가 있었던 위치(x좌표)	float
killer_position_y	killer가 있었던 위치(y좌표)	float
map	게임 맵 (Erangel / Miramar)	object

columns	설명	type
time	kill이 발생한 시간	float
victim_name	희생자 game id	object
victim_position_x	희생자가 있었던 위치(x좌표)	float
victim_position_y	희생자가 있었던 위치(y좌표)	float
match_id	해당 게임 고유 식별 id	object

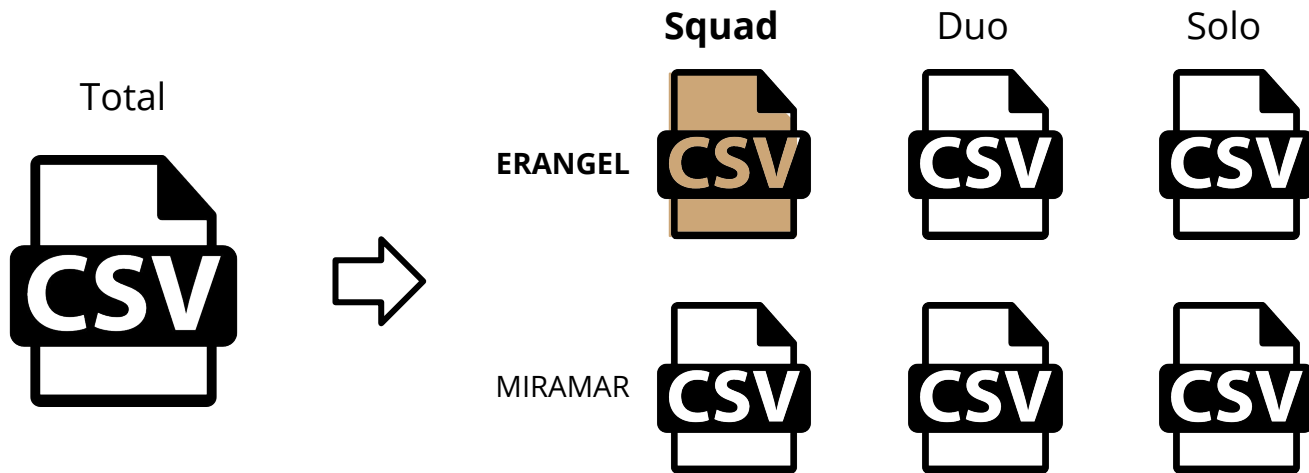
Dataset 설정

aggregate 데이터와 deaths 데이터를 합쳐 총합 데이터를 만듦



Dataset 설정

1. 가장 높은 선호도를 보인 ERANGEL 맵을 한정
2. 팀 별 데이터를 이용하고자 Squad 데이터로 한정



파생변수

기존 데이터가 사후 데이터이기 때문에, 가설에 입각한 새로운 변수를 생성하여 분류 모델에 반영

가설 : 해당 변수들이 top10 분류에 있어서 효과적일 것이다.

무기별 영향력

탈 것 활용 비율

outlier 존재의 영향력

교전거리

날짜별 match(시간별, 요일별)

팀의 호전성

match의 수준/team의 수준

핫플레이스에서 킬(17곳)

킬당 데미지

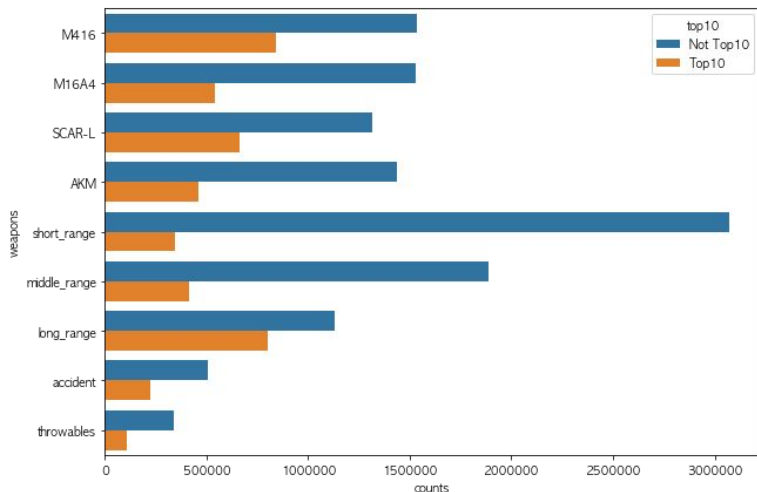
파생변수

무기별 영향력

- Top10의 경우 장거리용 무기를 선호하지만, Top10이 아닌 경우 단거리용 무기 사용 빈도가 매우 높음

M416	주류무기 M416
SCAR-L	주류무기 SCAR-L
M16A1	주류무기 M16A1
AKM	주류무기 AKM
short_range	근거리무기
middle_range	중거리무기 (smg, AR)
long_range	장거리무기 (sniper, DMR)
throwables	투척무기
accident	사고사

무기 별 사용 빈도



파생변수

outlier 존재의 영향력

- 팀 내 Abusing user가 없는 경우가 89%로 대부분을 차지함

Abusing user 선정 기준

탈것을 이용한 이동 거리	30km 초과
걸어서 이동한 거리	10km 초과
킬	30 kill 초과
데미지	3000 damage 초과
교전 거리	400m 초과
기절 횟수	11번 초과
게임에서 살아남은 시간	1900초 초과

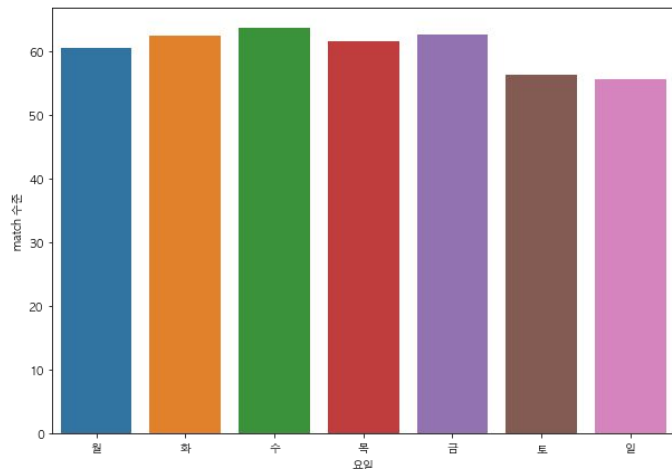
팀 내 Abusing user의 수 (명)	비율
0	89%
1	9%
2	1%
3	1%
4	1%

파생변수

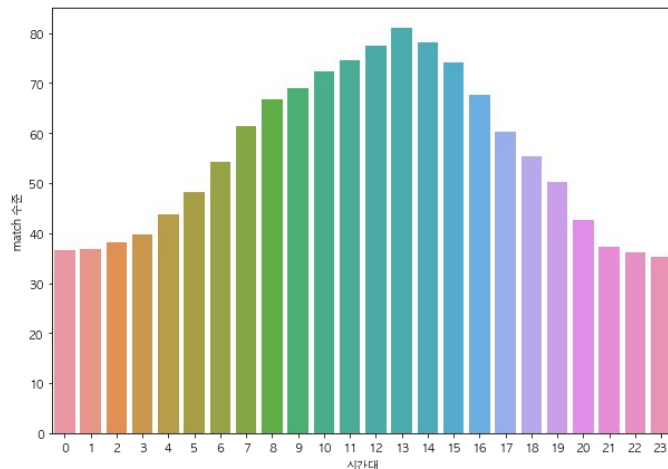
날짜별 match 수준(시간별, 요일별)

- 요일에 따른 match의 수준은 큰 차이는 없으나, 시간대 별 차이는 존재하는 것으로 확인됨
- 오전 8시 이후부터 오후 5시까지 난이도가 높은 match가 많음

요일 별 match 수준



시간대 별 match 수준



파생변수

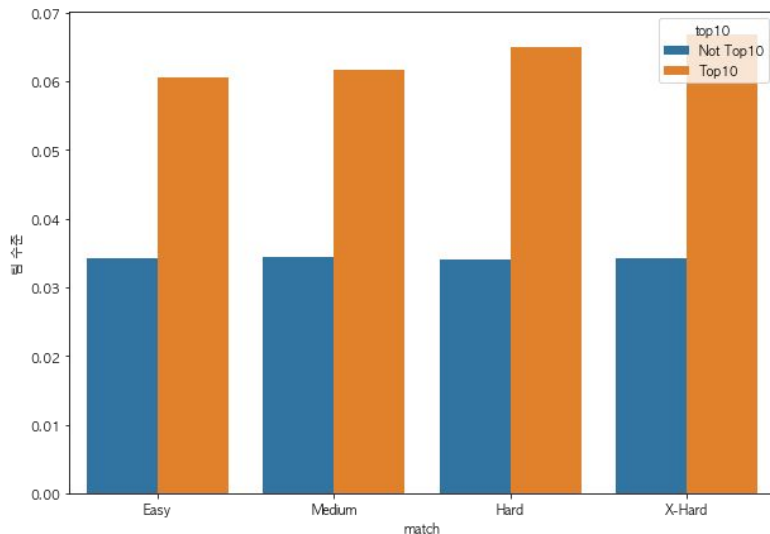
match의 수준 / team의 수준

- 매치의 수준과 상관 없이 Top10을 달성한 팀원의 누적 실력은 Top10을 달성하지 못한 팀에 비해 월등히 높음

Score (등수별 점수)*0.1 + np.log(데미지+1) + 기절한 횟수*(-0.1)

match의 수준 해당 match에 참여한 player들의 누적 점수의 평균 값

team의 수준 해당 match의 점수에서 team의 점수가 차지하는 비중



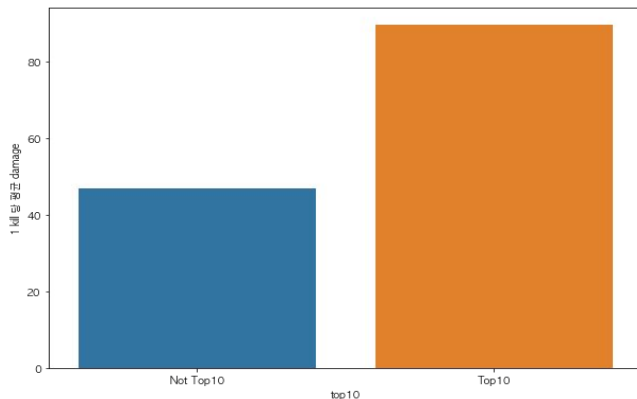
파생변수

킬당 데미지

- Top10인 팀이 1 kill을 달성할 때 평균 90의 damage를 입히는 반면 Top10이 아닌 팀은 47밖에 입히지 못함
- Top10인 경우 급소타격 능력이 높아 적은 횟수의 공격으로 적을 죽이는 반면, Top10이 아닌 경우 급소타격 능력이 낮음
- Top10의 경우 개별 작전 수행이 어느정도 가능하나, Top10이 아닌 경우 개별 작전 수행이 위험함

dmg_per_kills 1 kill 당 damage

1 kill 당 평균 damage

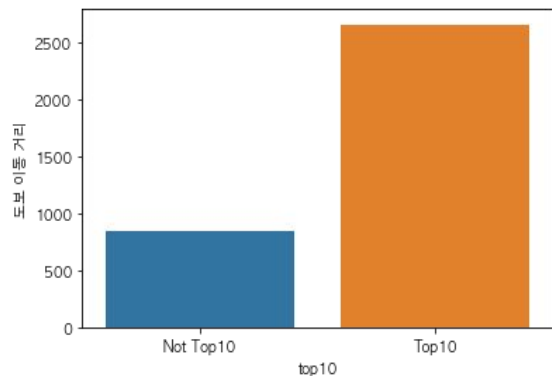


파생변수

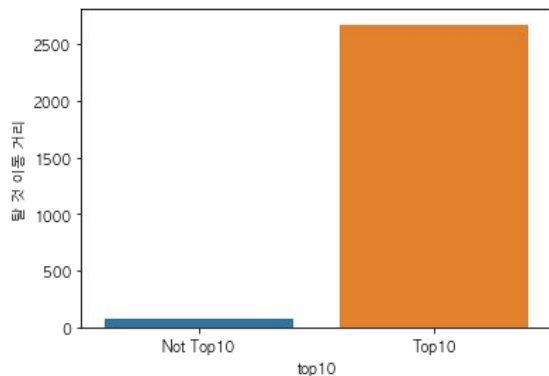
탈 것 활용 비율

- Top10은 게임에서 전반적인 이동 거리가 긴 편임
- Top10은 도보로 이동한 거리와 탈 것을 이용하여 이동한 거리가 비슷하지만, Top10이 아닌 팀은 탈 것을 활용한 이동 거리가 현저히 떨어짐

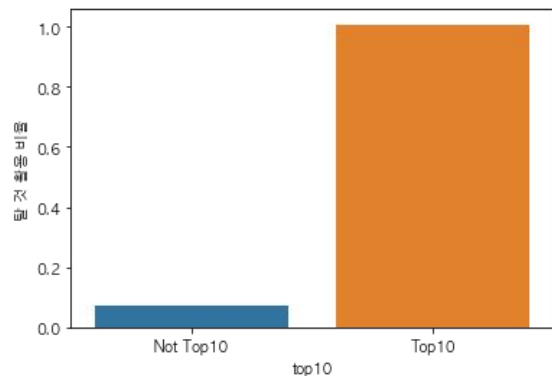
도보 이동거리



탈 것 이동거리



탈 것 활용 비율



파생변수

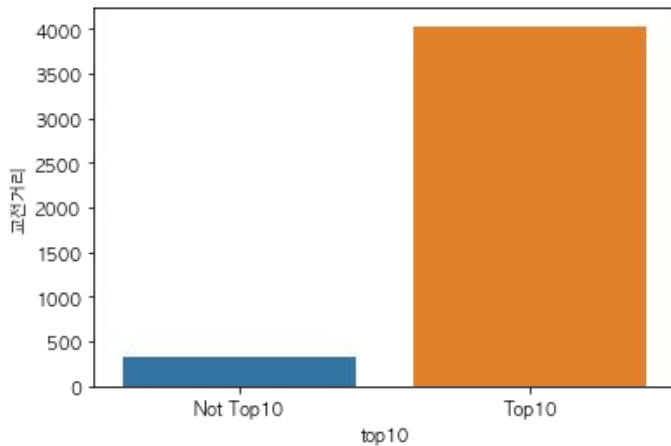
교전거리

- Top10의 경우 평균 교전거리는 40m로 장거리 공격이 많음

kill_dist

교전거리

팀 별 교전거리



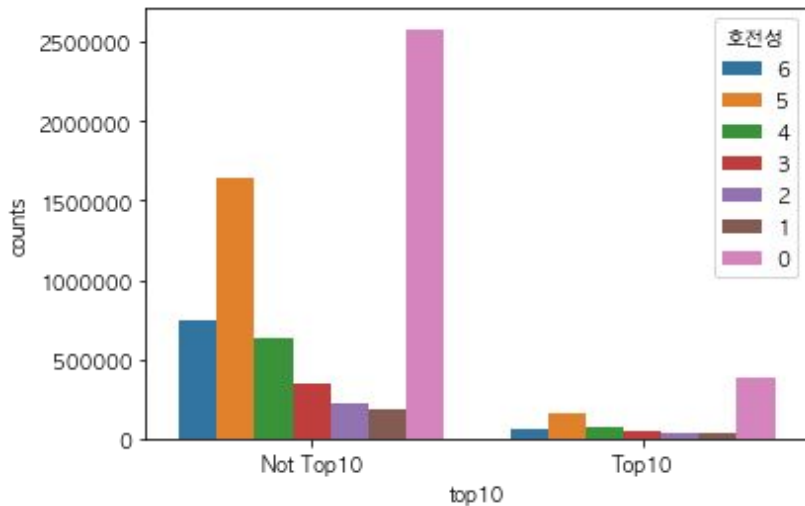
파생변수

팀의 호전성

- 게임 시작 후 초반부(750초 이내)에 교전을 하는 경우, Top10 여부와 상관 없이 대부분의 팀은 경기가 시작되고 250초 이내에 첫 킬을 달성함

tendency

팀의 호전성



호전성	의미
6	125초 이내
5	126초 ~ 250초 이내
4	251초 ~ 375초 이내
3	376초 ~ 500초 이내
2	501초 ~ 625초 이내
1	626초 ~ 750초 이내
0	경기 초반부 이후

파생변수

핫플레이스에서 킬 (17곳)

- 아이템이 풍부한 마을과 전략적 요충지에서
kill을 한 횟수

아이템이 풍부한 마을

- 배그 inven 의 아이템 지도를 통해 선정

전략적 요충지

- 게임을 하며 쌓인 경험적 지식을 통해 선정

아이템이 풍부한 지역이란?



출처 : 배그 inven

파생변수

핫플레이스에서 킬 (17곳)

- 최종 선정 지역

hot_place

핵심구역에서 kill을 한 횟수



Target 및 Feature 설정

Target

target	설명	type
top_10	0 : top10 달성 실패 1 : top10 달성	category

damage 관련 feature

feature	설명	type
dmg_per_kills	1 kill 당 damage	float

거리 / 위치 관련 feature

feature	설명	type
kill_dist	사살 거리	float
ride_ratio	탈 것 타고 이동/도보 이동	float
hot_place	핵심 구역에서 kill을 한 횟수	float

시간 관련 feature

feature	설명	type
date_time	시간대 별 분류	category
day_of_week	요일 별 분류	category

Target 및 Feature 설정

무기 관련 feature

feature	설명	type
M416	주류무기 M416	int
SCAR-L	주류무기 SCAR-L	int
M16A1	주류무기 M16A1	int
AKM	주류무기 AKM	int
short_range	근거리무기	int
middle_range	중거리무기(smg, AR)	int
long_range	장거리무기(Sniper, DMR)	int
throwables	투척무기	int
accident	사고사	int

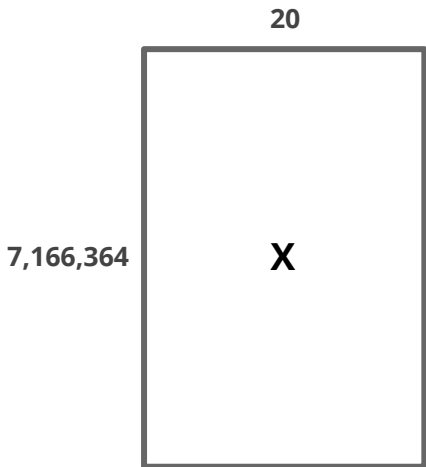
match / team 관련 feature

feature	설명	type
outlier	팀별 어뷰징 유저	int
team_level	매치 별 팀의 수준	float
match_level	각 매치의 수준	float
team_cum_num	팀 게임 누적 횟수	int
tendency	팀의 호전성	category

Target 및 Feature 설정

Problem

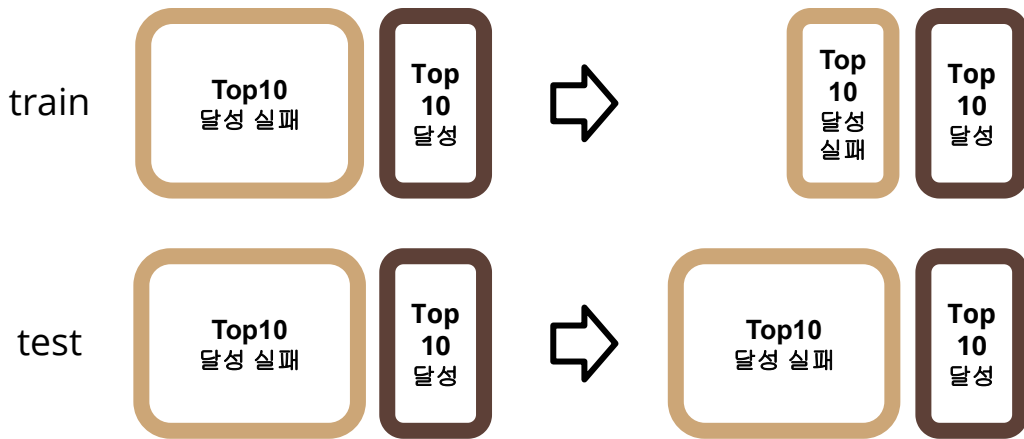
- data set의 크기가 너무 큼
- target의 class가 imbalanced 함



Target 및 Feature 설정

Solution

- train set에 undersampling 진행
- test set은 원래의 비율을 유지한 채 sampling 진행

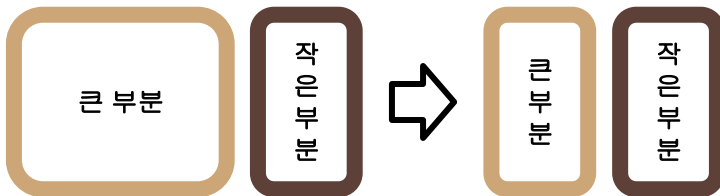


Target 및 Feature 설정

Imbalanced한 Data 에서의 Sampling Method

1. Undersampling

- 작은 부분에 맞춰 큰 부분을 제거
- 문제점 : 가지고 있는 데이터 손실

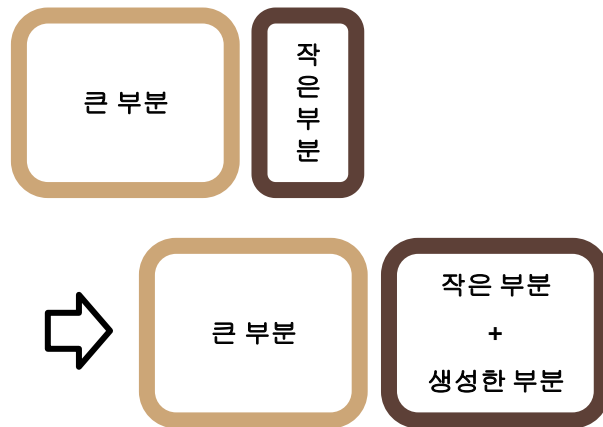


Target 및 Feature 설정

Imbalanced한 Data 에서의 Sampling Method

2. Oversampling

- 작은 부분을 중복해서 뽑거나 새로 생성
- 문제점
 1. 새로 생성 : 실제 데이터를 대표할 수 없는 데이터 생성 가능
 2. 중복 추출 : 적은 부분의 데이터에 가중치를 가진 모형이 학습될 수 있음



Target 및 Feature 설정

Undersampling을 선택한 이유

1. 현재 Data 크기가 너무 큼

: Data의 크기를 줄이기 위한 방법으로 선택

2. top10에 들어가는 데이터를 만들기에는 문제가 있음

: Oversampling 은 top10에 들어가는 새로운 Data를 만들거나, 반복 추출하는 Method 인데,

과적합 문제라던지, 노이즈 또는 이상치에 민감함

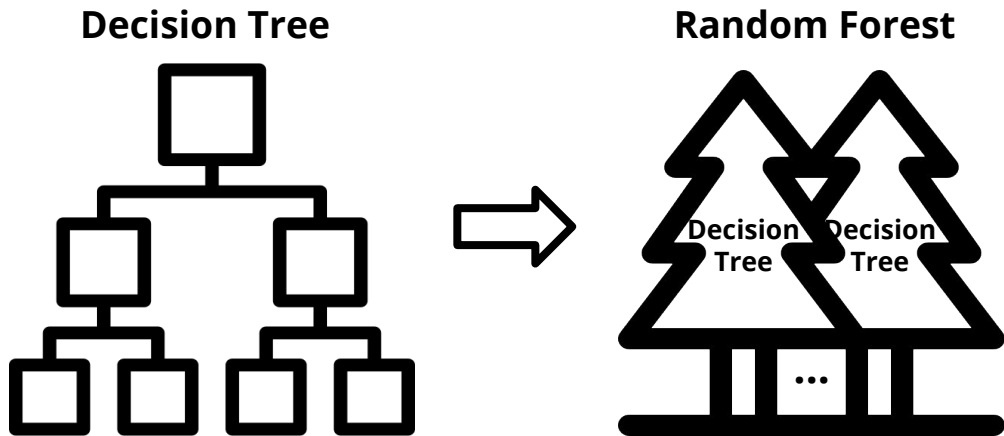
따라서 분석할 Dataset에 적합하지 않다고 판단함

4. Modeling

Random Forest

Random Forest

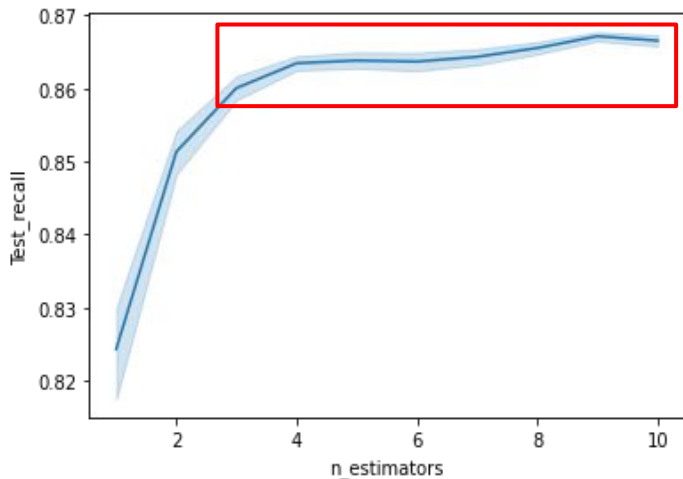
- 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 결정 트리로부터 부류(분류) 또는 평균 예측치(회귀 분석)를 출력함으로써 동작



Hyper parameter Tuning

하이퍼 파라미터

- Grid search를 활용하여 최종 Parameter 값 결정



n_estimators

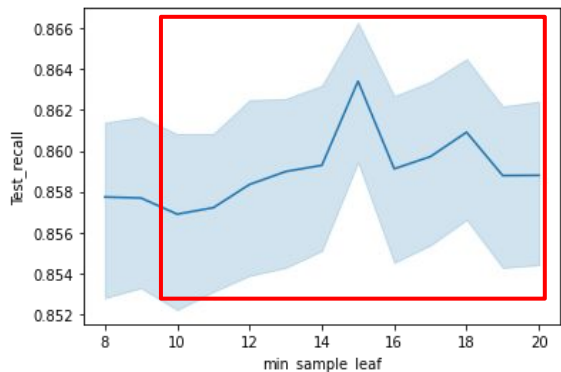
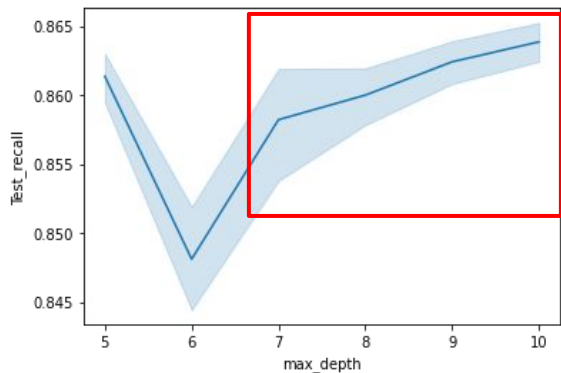
: RandomForest에서 DecisionTree의 수

N_estimators가 증가함에 따라 Test set의 recall 값이 커지다가 일정 값에 수렴하는 경향을 보임

∴ Elbow method를 차용하여 꺾이는 부분인 3이후의 값으로 grid search 범위를 결정

*실선은 해당 지표값에서 recall의 평균 값이며, 실선 주위의 범위는 해당 지표를 제외한 parameter를 조정함에 따라 생기는 값의 범위에 해당

Hyper parameter Tuning



max_depth

: 트리의 최대 깊이

Max_depth가 증가함에 따라 Test set의 recall 값이 6에서 최저점을 기록하고 이후 꾸준히 올라갈 경향을 보임

∴ 최고점을 기록하는 6이후의 값으로 grid search 범위를 결정

*실선은 해당 지표값에서 recall의 평균 값이며, 실선 주위의 범위는 해당 지표를 제외한 parameter를 조정함에 따라 생기는 값의 범위에 해당

min_samples_leaf

: 리프 노드가 되기 위한 최소 샘플 데이터 수

Min_samples_leaf가 증가함에 따라 Test set의 recall 값이 15에서 최고점을 기록하고 이후 진동할 것으로 기대되는 경향을 보임

∴ 증가하는 경향을 보이기 시작하는 10이후의 값으로 grid search 범위를 결정

*실선은 해당 지표값에서 recall의 평균 값이며, 실선 주위의 범위는 해당 지표를 제외한 parameter를 조정함에 따라 생기는 값의 범위에 해당

Hyper parameter Tuning

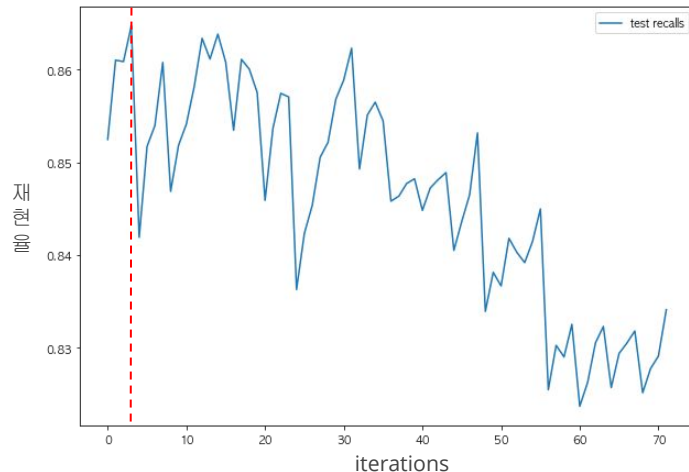
Grid search란?

: 관심 있는 매개변수들을 대상으로 가능한 모든 조합을 시도하여 최적의 매개변수를 찾는 방법
매개변수를 튜닝하여 일반화 성능을 개선해줌

test recall이 가장 높은 값을 가지는 **parameter 조합**

parameters	values
max_depth	7
min_samples_leaf	12
n_estimators	10

Grid search 각 iteration의 test recall 값



분류 모형 평가 지표

Confusion Matrix		실제	
		True	False
예측	True	TP	FP
	False	FN	TN

Accuracy(정확도)	실제 값과 분류 값이 같은 비율	F1-score	Precision과 Recall의 조화평균
$\frac{TP + TN}{TP + FP + FN + TN}$		$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	
Recall(재현율)	실제 True인 것 중 True로 분류한 비율	Precision(정밀도)	True로 분류한 것 중 실제 True인 비율
$\frac{TP}{TP + FN}$		$\frac{TP}{TP + FP}$	

Model Performance

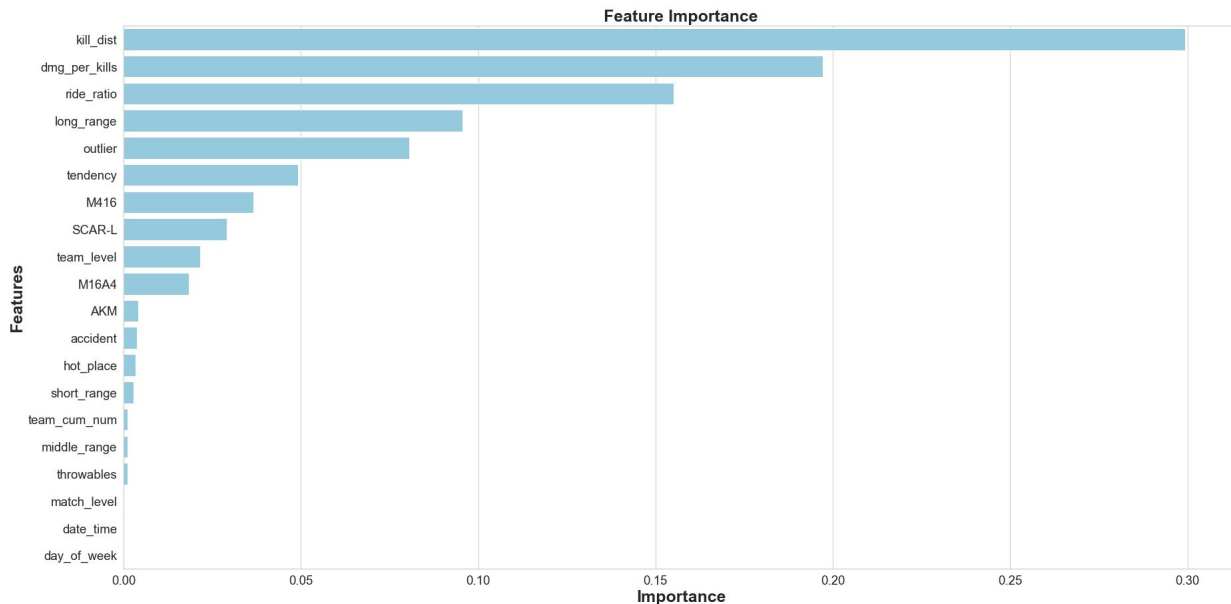
평가지표	Recall
<p>불균형 데이터를 분류하는 모형에서 평가지표로는 F1-score가 타당하다고 생각하지만, Recall과 Precision이 모두 Overfitting이 없는 결과를 도출하는데 어려움이 존재함</p> <p>따라서 F1-score를 구하는데 사용되는 지표들 중 분석 목적이 TOP10을 달성한 데이터를 잘 분류하는 것이므로 Recall이 더 적절하다고 판단함</p> <p>∴ Recall을 기준으로 grid search를 수행하여 parameter를 튜닝하였고, 오른쪽 표와 같은 성능을 얻게됨</p>	

	Train	Test
Recall	86.9%	86.8%
Accuracy	82.1%	78.5%
Precision	79.3%	32.5%
F1-score	82.9%	47.3%

Feature Importance

변수별 영향력

- Importance가 높은 10개의 컬럼만 선택해 차원을 축소한 데이터로 모델을 다시 구축함



	Features	Gini-Importance
0	kill_dist	0.299
1	dmg_per_kills	0.197
2	ride_ratio	0.155
3	long_range	0.096
4	outlier	0.081
5	tendency	0.049
6	M416	0.037
7	SCAR-L	0.029
8	team_level	0.022
9	M16A4	0.018
10	AKM	0.004
11	accident	0.004
12	hot_place	0.003
13	short_range	0.003
14	team_cum_num	0.001
15	middle_range	0.001
16	throwables	0.001
17	match_level	0.000
18	date_time	0.000
19	day_of_week	0.000

Dimensionality Reduction

Before reduction

	Train	Test
Recall	86.9%	86.8%
Accuracy	82.1%	78.5%
Precision	79.3%	32.5%
F1-score	82.9%	47.3%

After reduction

	Train	Test
Recall	87.1%	87.0%
Accuracy	82.1%	78.2%
Precision	79.2%	32.3%
F1-score	83.0%	47.2%

5. 결론

결론

- 최종 모형
RandomForest 알고리즘으로 차원이 축소된
10개의 변수(교전거리, 킬당 데미지 등)와 Top10 달성
여부를 분류 하는 모형을 학습하였고,
N_estimator : 10, Max_depth : 7, Min_sample_leaf : 12
일 때 실제 Top10을 87% 구별 할 수 있는 모형을
제작함

Algorithm	RandomForest	
# of feature	10	
Target	Top10 달성 여부	
Parameter	N_estimator	: 10
	Max_depth	: 7
	Min_sample_leaf	: 12
Recall	Train	: 87.1%
	Test	: 87.0%

- 향후 발전 방향
Model 성능의 평가 지표로 Recall을 사용하였지만, Imbalanced data의 평가 지표로는
F1-score가 더 적절하므로 F1-score의 최적화를 목표로 새로운 Modeling 과정을 수행



Q & A

감사합니다 :)