



# 스타크래프트2 승률 예측

통계학과 최대한  
통계학과 김부겸

1. 대회 소개
2. 데이터 살펴보기 및 EDA
3. 전처리 및 모델링
4. 결론 및 아쉬웠던 점

# 1. 대회 소개



## - 월간 데이콘 행동 데이터 분석 인공지능 AI 경진대회



**DACON** 커뮤니티 대회 학습 랭킹 더보기

월간 데이콘 행동 데이터 분석 인공지능 AI 경진대회

알고리즘 | 정형 | 분류 | 게임 | AUC

₩ 상금 100만원

📅 2020.03.01 ~ 2020.04.15 17:59 [+ Google Calendar](#)

👤 1,087명 📄 마감

결과 데이콘3

참여중

대회안내 데이터 코드 공유 토크 리더보드 제출

☰ 개요

📄 규칙

📅 일정

💰 상금

📄 동의사항

**[배경]**

E-Sports 분야에서 한국은 위용이 넘칩니다. 많은 E-sports 대회에서 한국 대표팀이 상위권을 대거 장악하였고, 어떤 대회의 상위권에서는 한국인이 아닌 선수를 찾기 힘든 경우도 있습니다.

선수뿐 아니라 일반인의 실력도 만만치 않습니다. 스타크래프트라는 게임을 만든 블리자드 사에서는 한국 서버를 이렇게 묘사합니다.

‘조용한 아침의 땅은 전술전략에 통달했으며, 지구상에서 가장 유명한 ‘스타크래프트’ 선수들의 고향이기도 하다. 이 아수라장에 놀러 오지 마라.’

‘게임을 잘하는 나라’, ‘E-sports의 성지’라는 호칭을 얻게 된 요인에는 게이머들의 탁월한 전략이 함께합니다.

그리고 여러분은 데이터를 분석하여 전략을 발전시킬 수 있는 능력을 갖추고 있습니다.

E-Sports 속 한국이란 나라의 위용에 걸맞은 알고리즘을 만들어주세요! 여러분이 만든 알고리즘이 우리의 게임 실력을 한층 더 발전시킬 수 있습니다.

## 2. 데이터 살펴보기 및 EDA



### - 데이터 개요

- 기본 column

game_id	경기 구분 기호
winner	선수 , 0: player 0, 1: player 1
time	경기 시간, ex) 2.24 = 2분 24초
player	선수 , 0: player 0, 1: player 1
species	종족, T: 테란, P: 프로토스, Z: 저그
event	행동 종류
event_contents	행동 상세

- event 상세 정보

Ability	생산, 공격 등 선수의 주요 행동
AddToControlGroup	부대에 추가
Camera	시점 선택
ControlGroup	부대 행동
GetControlGroup	부대 불러오기
Right Click	마우스 우클릭
Selection	객체 선택
SetControlGroup	부대 지정

	game_id	winner	time	player	species	event	event_contents
	0	0	1 0.00	0	T	Camera	at (145.25, 21.5078125)
	1	0	1 0.00	1	T	Camera	at (22.75, 147.0078125)
	2	0	1 0.02	0	T	Selection	['OrbitalCommand [3080001]']
	3	0	1 0.02	0	T	Ability	(1360) - TrainSCV
	4	0	1 0.14	0	T	Camera	at (142.99609375, 24.50390625)
	...	...	...	...	...	...	...
	67091771	38871	0 8.51	0	Z	Camera	at (139.578125, 62.58203125)
	67091772	38871	0 8.52	1	T	GetControlGroup	NaN
	67091773	38871	0 8.52	0	Z	Camera	at (122.42578125, 45.4296875)
	67091774	38871	0 8.52	0	Z	Camera	at (122.42578125, 43.25390625)
	67091775	38871	0 8.52	1	T	Ability	(1360) - TrainSCV

67091776 rows × 7 columns

## 2. 데이터 살펴보기 및 EDA



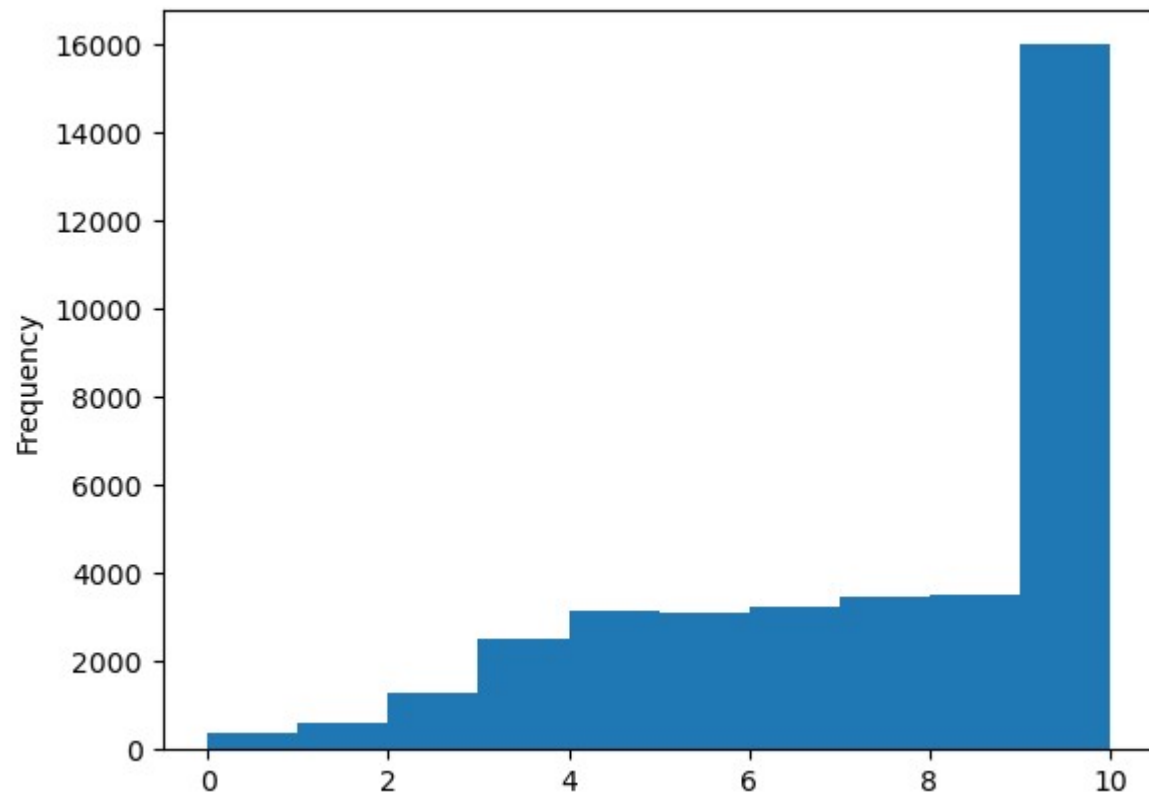
### - 데이터의 간단한 이해

	game_id	winner	time	player	species	event	event_contents
29	0	1	0.33	1	T	Ability	(1360) - TrainSCV
43	0	1	0.37	1	T	Ability	(1021) - BuildSupplyDepot; Location: (28.0, 14...
56	0	1	0.40	1	T	Ability	(1022) - BuildRefinery; Target: CreepOnlyBlock...
57	0	1	0.40	1	T	Selection	[OrbitalCommand [33C0001]]
63	0	1	0.41	1	T	Ability	(1360) - TrainSCV
124	0	1	1.21	1	T	Ability	(1023) - BuildBarracks; Location: (28.5, 144.5...
130	0	1	1.23	1	T	Selection	['SCV [3400001]', 'SCV [3440001]', 'SCV [34C00...
141	0	1	1.30	1	T	Selection	['SCV [3680001]', 'SCV [3840001]]
142	0	1	1.30	1	T	Right Click	Target: Refinery [03800001]; Location: (27.5, ...
146	0	1	1.37	1	T	Selection	['Barracks [3A40002]]
147	0	1	1.39	1	T	SetControlGroup	NaN
183	0	1	2.12	1	T	Selection	[OrbitalCommand [33C0001]]
210	0	1	2.24	1	T	Selection	['SCV [3480001]', 'SCV [36C0001]]
213	0	1	2.25	1	T	Selection	['Barracks [3A40002]]
223	0	1	2.30	1	T	Ability	(1021) - BuildSupplyDepot; Location: (24.0, 14...
235	0	1	2.35	1	T	Ability	(13E0) - TrainMarine

## 2. 데이터 살펴보기 및 EDA



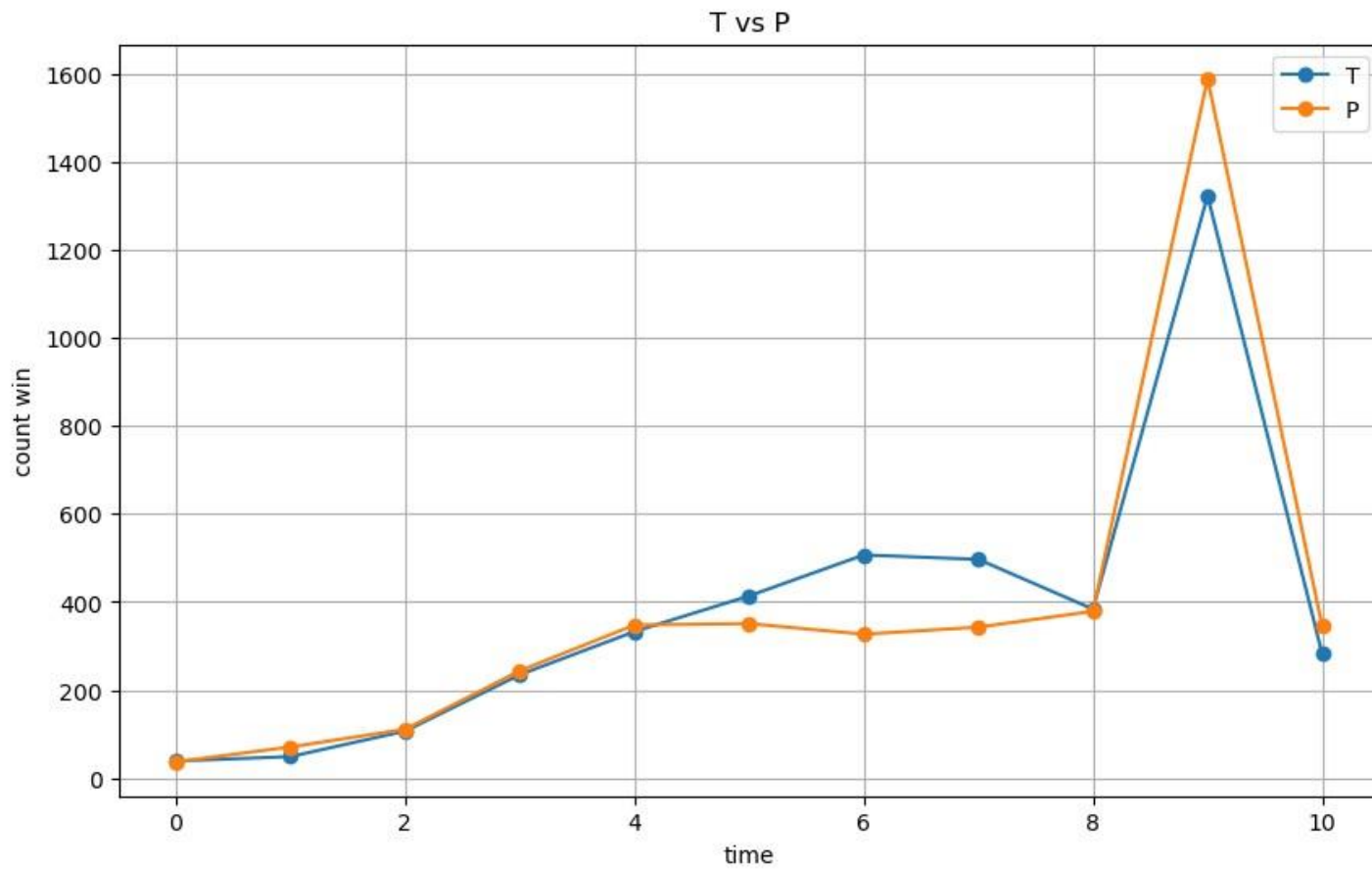
### - EDA : 게임 시간 분포



## 2. 데이터 살펴보기 및 EDA



### - EDA : 게임 시간과 승률

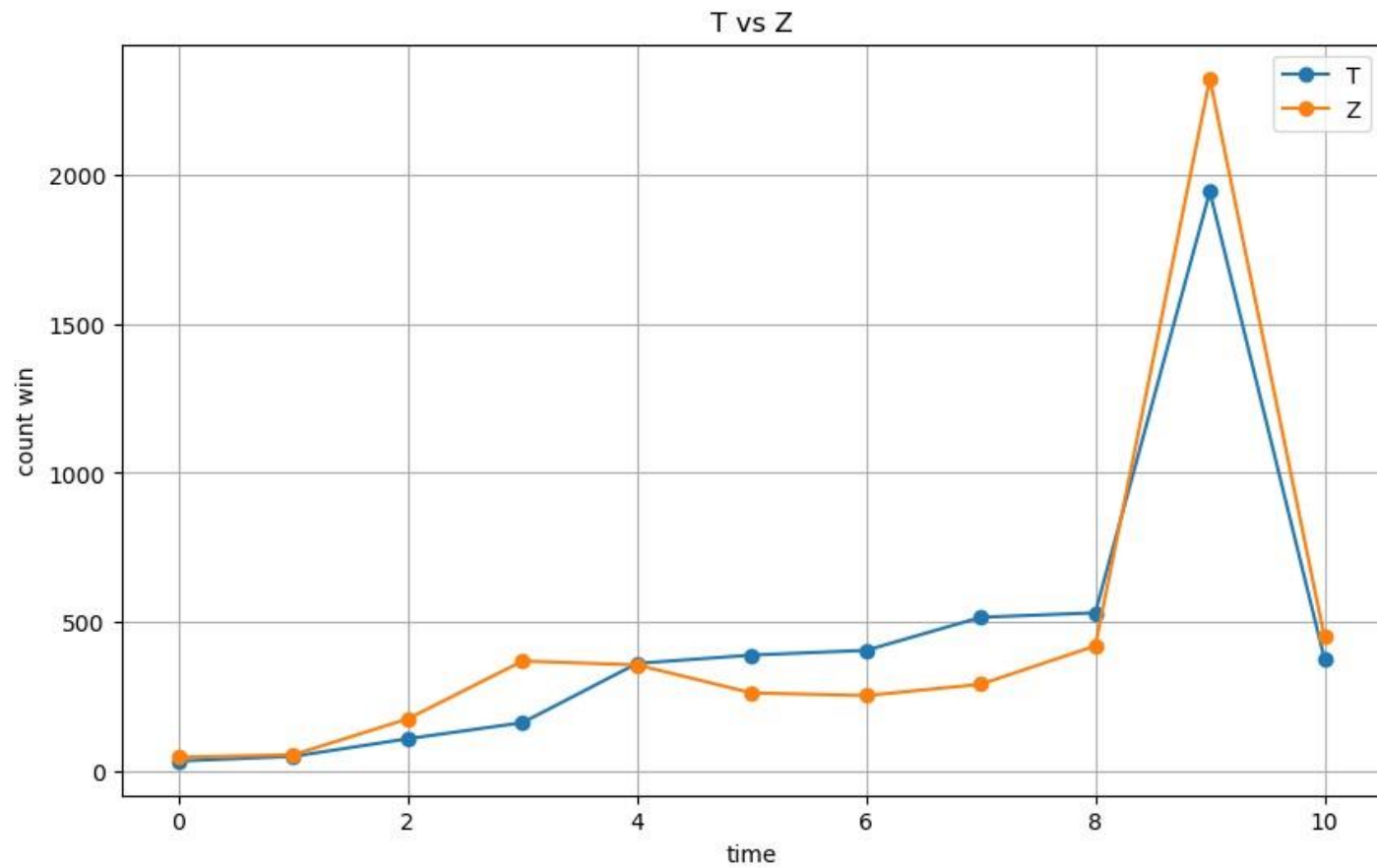




## 2. 데이터 살펴보기 및 EDA



### - EDA : 게임 시간과 승률

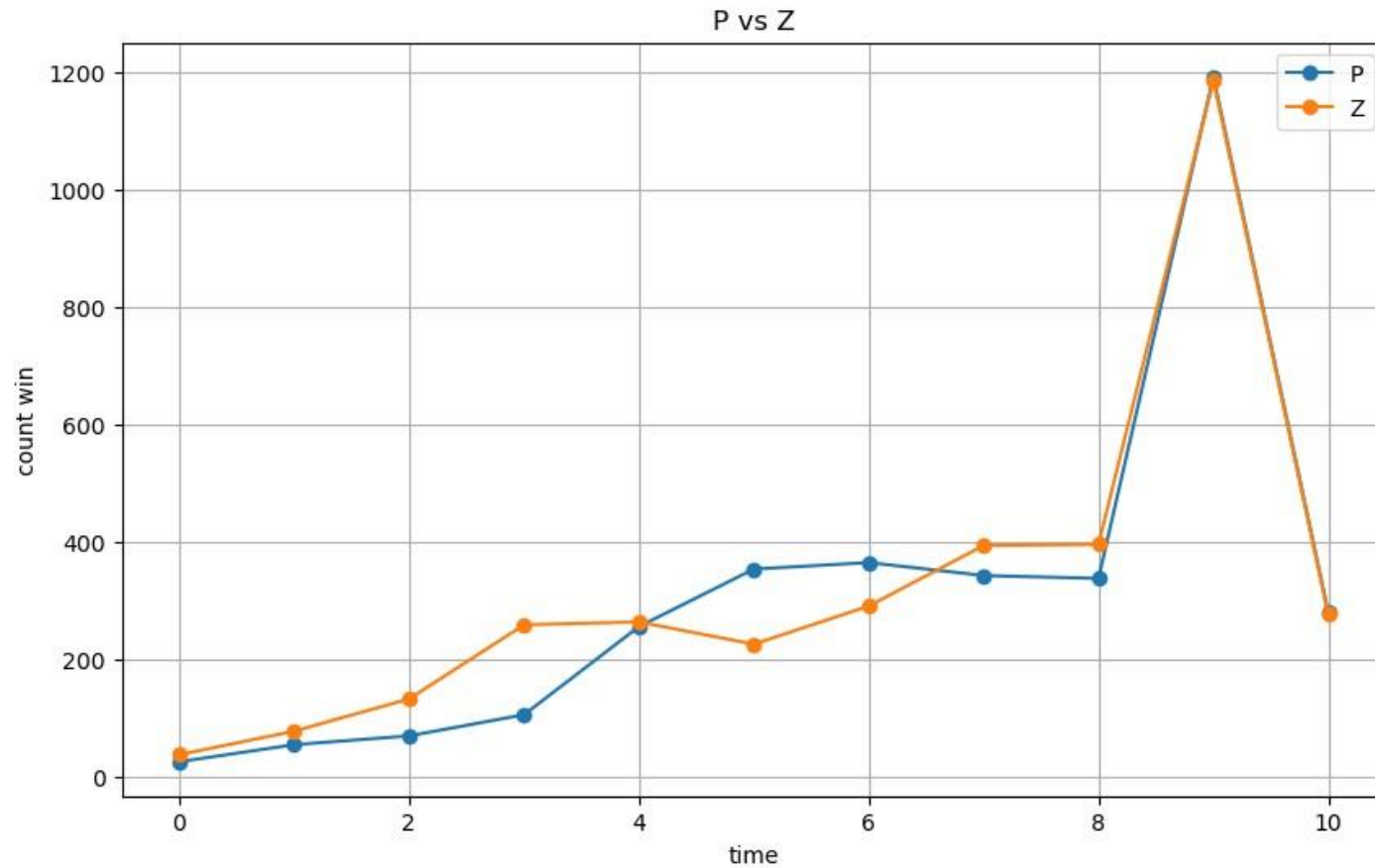




## 2. 데이터 살펴보기 및 EDA



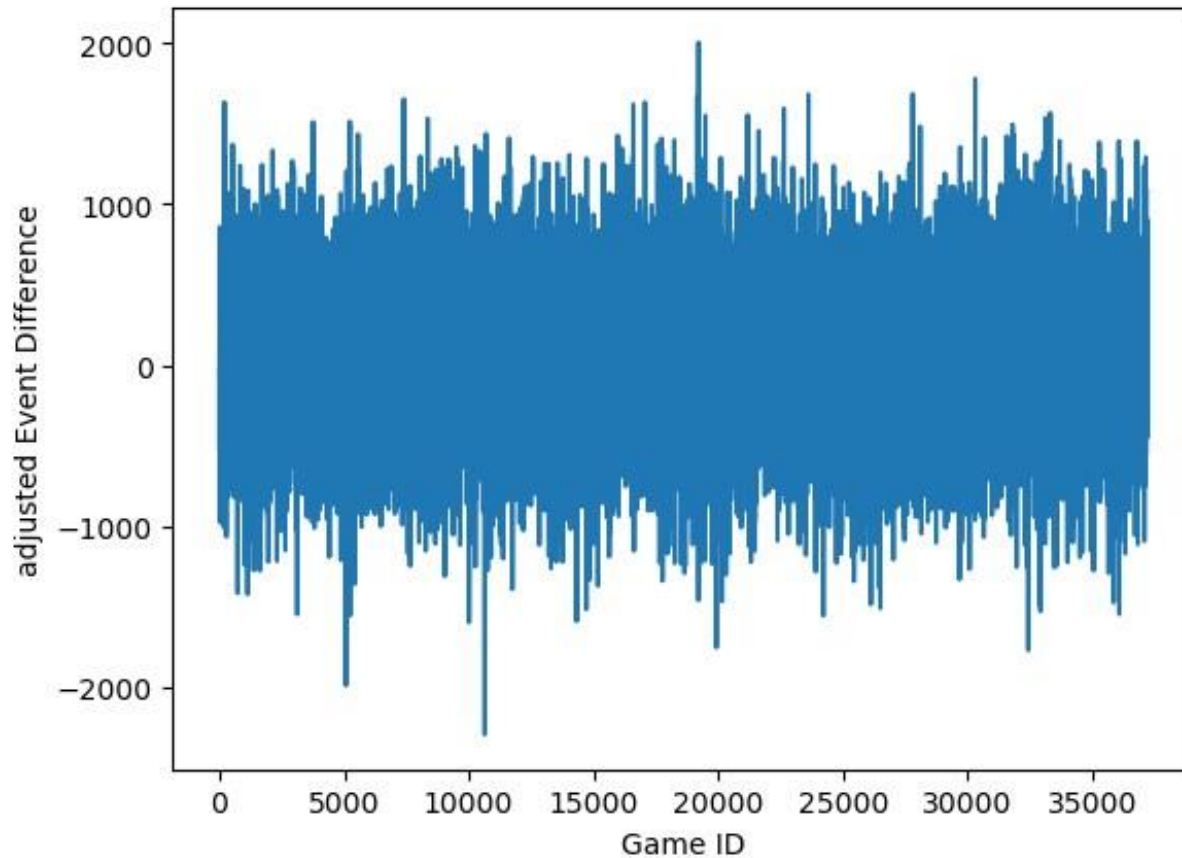
### - EDA : 게임 시간과 승률



## 2. 데이터 살펴보기 및 EDA



### - EDA : APM



- APM : Actions Per Minute의 줄임말
- event의 횟수가 높을수록 손이 바쁘다고 판단
- 승자가 패자보다 많은 event를 한 경기의 비율 : 54.17%

## 2. 데이터 살펴보기 및 EDA



### - EDA : Camera 변수 살펴보기

- Camera 변수 활용을 위해 정규표현식을 사용해서 event\_contents를 목적에 맞게 분리

	game_id	winner	time	player	species	event	event_contents	x_position	y_position		unit	action_name	Attack
	0	0	1	0	0	0	at (145.25, 21.5078125)	145.250000	21.507812		NaN	NaN	NaN
	1	0	1	0	1	0	at (22.75, 147.0078125)	22.750000	147.007812		NaN	NaN	NaN
	2	0	1	0	0	1	['OrbitalCommand [3080001]']	NaN	NaN		'OrbitalCommand [3080001]	NaN	NaN
	3	0	1	0	0	2	(1360) - TrainSCV	NaN	NaN		NaN	TrainSCV	NaN
	4	0	1	0	0	0	at (142.99609375, 24.50390625)	142.996094	24.503906		NaN	NaN	NaN
	...	...	...	...	...	...	...	...	...		...	...	...
	64204072	37217	1	1	0	5	NaN	NaN	NaN		NaN	NaN	NaN
	64204073	37217	1	1	0	5	NaN	NaN	NaN		NaN	NaN	NaN
	64204074	37217	1	1	1	0	at (100.25390625, 103.9921875)	100.253906	103.992188		NaN	NaN	NaN
	64204075	37217	1	1	0	5	NaN	NaN	NaN		NaN	NaN	NaN
	64204076	37217	1	1	1	1	['Probe [3580001]']	NaN	NaN		'Probe [3580001]	NaN	NaN

64204077 rows × 12 columns

### - EDA : 마지막 교전 지점 살펴보기



- 하얀 색 원에서 마지막 교전이 일어났다고 가정
- 주황색 플레이어가 교전에서 지속적으로 밀렸다고 생각할 수 있다.
- 만약 그 이후로 게임이 끝났다면, 아마도 주황색 플레이어가 게임에서 패배했을 것이라고 가정

## 2. 데이터 살펴보기 및 EDA



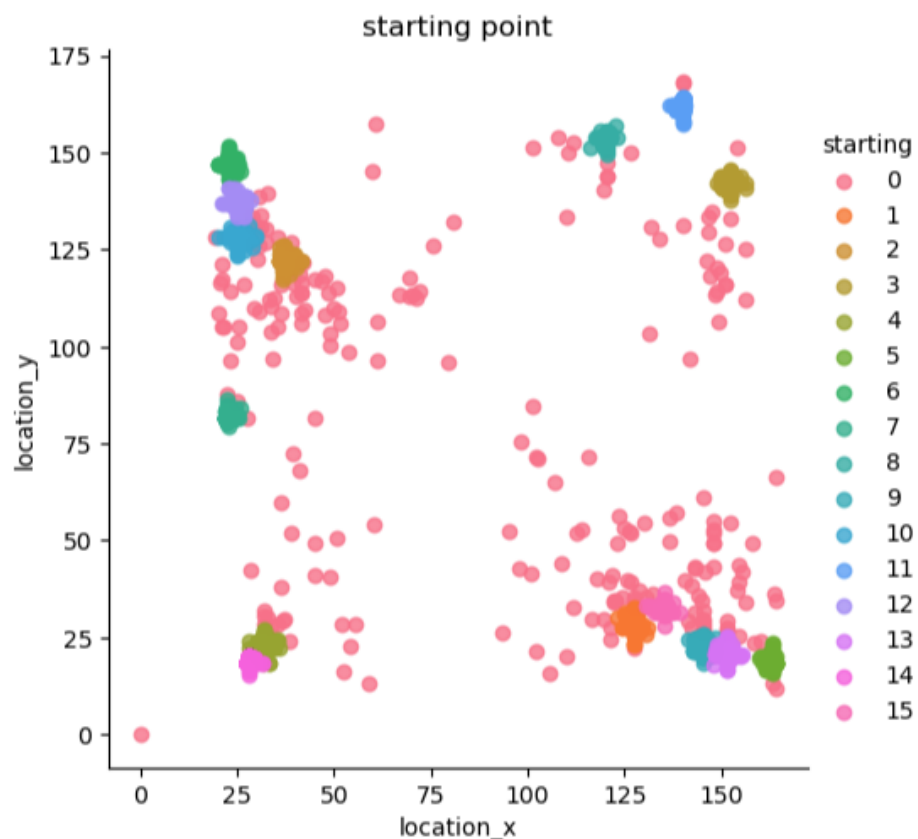
### - EDA : 마지막 교전 지점 살펴보기

	game_id	player0_starting	player1_starting	Last_Attack	Closer_Player	Further_Player	winner
0	0	at (145.25, 21.5078125)	at (22.75, 147.0078125)	(28.266357421875, 122.27685546875)	1	0	1
1	1	at (140.0, 162.0078125)	at (28.0, 18.5078125)	(37.140869140625, 47.5185546875)	1	0	1
2	2	at (151.25, 20.5078125)	at (24.75, 128.0078125)	(54.7626953125, 103.628662109375)	1	0	0
3	3	at (127.25, 27.5078125)	at (24.75, 137.0078125)	(122.658447265625, 54.605712890625)	0	1	0
4	4	at (36.75, 122.0078125)	at (163.25, 18.5078125)	(55.038818359375, 116.45654296875)	0	1	0
...	...	...	...	...	...	...	...
32241	37211	at (28.0, 18.5078125)	at (140.0, 162.0078125)	(143.161376953125, 113.11376953125)	1	0	0
32242	37212	at (36.75, 122.0078125)	at (163.25, 18.5078125)	(78.136962890625, 93.005126953125)	0	1	1
32243	37213	at (151.25, 20.5078125)	at (24.75, 128.0078125)	(47.311279296875, 127.938720703125)	1	0	1
32244	37214	at (22.75, 147.0078125)	at (145.25, 21.5078125)	(90.146240234375, 82.214599609375)	1	0	1
32245	37216	at (151.25, 20.5078125)	at (24.75, 128.0078125)	(60.875244140625, 111.19482421875)	1	0	0

32246 rows × 7 columns

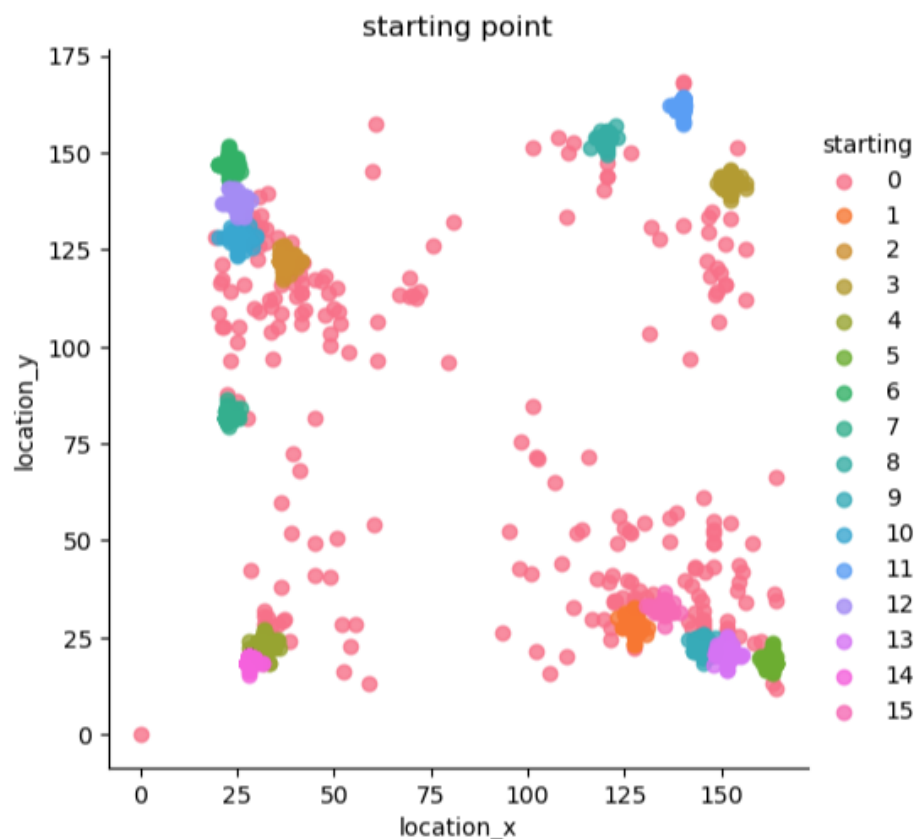
- Further\_Player와 winner가 같은 비율 : 50.62%

### - EDA : 맵 클러스터링



- game\_id 별로 각 플레이어들의 첫 Camera 좌표를 시작 지점이라고 설정
- 시작 지점들을 전부 좌표 위에 찍어본 다음, 가장 많이 겹친 15개를 기준으로 해서 K-means clustering을 시행
- Clustering한 군집의 각각 centroid 지점부터 거리가 5 이상인 점들은 핑크색 점(0)으로 다시 분류

### - EDA : 맵 클러스터링



```
array([[ 1,  6, 999],  
       [ 2, 11, 999],  
       [ 3, 14, 999],  
       [ 4,  8, 999],  
       [ 5, 10, 15],  
       [ 7,  9, 999],  
       [12, 13, 999]], dtype=int64)
```



- 맵을 총 7가지 케이스로 분류
- 한 쪽이라도 0으로 찍힌 경우는 다른 플레이어의 시작지점을 참고해서 결정



## 2. 데이터 살펴보기 및 EDA



### - EDA : 사용한 자원량(Mineral, Gas)

Unit	Race	Supply			Build time
Archon	Protoss	4	0	0	12
Auto-Turret	Terran	-	-	-	-
Baneling	Zerg	0.5	25	25	20
Banshee	Terran	3	150	100	60
Battlecruiser	Terran	6	400	300	90
Brood Lord	Zerg	4	150	150	34

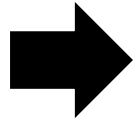
```
mineral_prices = {
    'Adept': 100,
    'AdeptPiercingWeapon': 0,
    'AiurLightBridgeAbandonedNE100ut': 0,
    'AiurLightBridgeAbandonedNE8': 0,
    'AiurLightBridgeAbandonedNE80ut': 0,
    'AiurLightBridgeNE100ut': 0,
    'AiurLightBridgeNE8': 0,
    'Archon': 0,
    'Armory': 150,
    'Assimilator': 75,
    'AutoTurret': 0,
    'Baneling': 25,
    'BanelingBurrowed': 0,
    'BanelingCocoon': 0,
```

```
gas_prices = {
    'Adept': 25,
    'AdeptPiercingWeapon': 0,
    'AiurLightBridgeAbandonedNE100ut': 0,
    'AiurLightBridgeAbandonedNE8': 0,
    'AiurLightBridgeAbandonedNE80ut': 0,
    'AiurLightBridgeNE100ut': 0,
    'AiurLightBridgeNE8': 0,
    'Archon': 0,
    'Armory': 100,
    'Assimilator': 0,
    'AutoTurret': 0,
    'Baneling': 25,
    'BanelingBurrowed': 0,
    'BanelingCocoon': 0,
```

- 게임별로 각 플레이어가 게임 진행동안 사용한 총 자원량을 측정
- 유닛 정보 사이트를 참고해서 유닛과 건물 필요 자원량을 저장해서 산출

### - EDA : 사용한 자원량(Mineral, Gas)

	game_id	player	winner	TotalMineral	TotalGas
0	0	0	1	1925	285
1	0	1	1	2550	325
2	1	1	1	4175	875
3	1	0	1	4025	500
4	2	0	0	4925	1125
...	...	...	...	...	...
64476	37213	0	1	3150	610
64477	37214	0	1	4475	585
64478	37214	1	1	5475	1035
64479	37216	0	0	2600	200
64480	37216	1	0	2650	275



	WinnerMineral	WinnerGas	LoserMineral	LoserGas	MineralDifference	GasDifference
game_id						
0	2550.0	325.0	1925.0	285.0	625.0	40.0
1	4175.0	875.0	4025.0	500.0	150.0	375.0
2	4925.0	1125.0	4525.0	635.0	400.0	490.0
3	5500.0	835.0	6950.0	1075.0	-1450.0	-240.0
4	1900.0	225.0	2775.0	100.0	-875.0	125.0
...	...	...	...	...	...	...
37212	4225.0	285.0	2700.0	355.0	1525.0	-70.0
37213	3175.0	350.0	3150.0	610.0	25.0	-260.0
37214	5475.0	1035.0	4475.0	585.0	1000.0	450.0
37216	2600.0	200.0	2650.0	275.0	-50.0	-75.0

- 사용한 자원량의 차이가 승패에 영향이 있는지 가설 검정을 해 본 결과, 유의미한 것으로 판단

### - 데이터 사전 전처리

Attack_count	Last_Attack
3	(28.266357421875, 122.27685546875, 40935)
6	(37.140869140625, 47.5185546875, 40929)
18	(54.7626953125, 103.628662109375, 32748)
25	(122.658447265625, 54.605712890625, 56356)
16	(55.038818359375, 116.45654296875, 46872)
...	...
17	(47.311279296875, 127.938720703125, 49137)
59	(90.146240234375, 82.214599609375, 32744)
0	NaN
12	(60.875244140625, 111.19482421875, 32748)
0	NaN

- 게임 특성상 승패가 결정되려면 Attack 행위가 필연적인데 Attack event가 한번도 발생하지 않은 게임은 플레이어가 나간 경우 등 비정상적으로 게임이 끝난 케이스
- 승률 예측에 있어서 좋지 않은 데이터라고 판단하여서 삭제함

#### - 데이터 전처리

- 플레이어 1 의 승률을 예측하는 것
- 게임 하나당 한 행으로 압축해서 확인하고자 했음.
- Camera 변수와 Ability 변수를 활용해 map과 총 자원 사용량 파생변수 생성
- Camera 변수와 Ability 변수 등을 제외하면 남는 변수가 거의 없을 뿐더러 GetControlGroup 등의 변수들은 전부 NaN값
- 나머지 변수들은 빈도를 센 후 (플레이어 1의 지표) - (플레이어 0의 지표) 로 해서 diff라고 덧붙임
- 위에서 설명했던 APM 관련 지표로 볼 수 있다.

## - 데이터 전처리

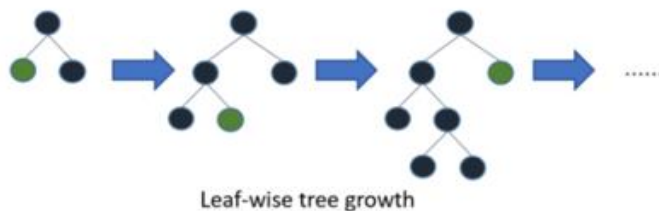
- Time 변수는 위에서 봤던 그래프를 기준으로 해서 0~4분대, 4~8분대, 8분대 이상 으로 분류
- 종족별 대진은 총 9가지 경우가 생김.
- 이러한 범주형 자료들을 모두 생성한 후 One hot Encoding을 시켜줌.

	game_id	winner	Camera_diff	Selection_diff	Ability_diff	Right Click_diff	SetControlGroup_diff	GetControlGroup_diff	AddToControlGroup_diff
0	0	1	-19.0	7.0	-0.0	-7.0	-2.0	-21.0	-2.0
1	1	1	231.0	-70.0	-10.0	-29.0	-2.0	-131.0	-1.0
2	2	0	312.0	142.0	16.0	44.0	-5.0	10.0	-1.0
3	3	0	-325.0	-32.0	7.0	-8.0	13.0	578.0	-0.0
4	4	0	-158.0	59.0	-21.0	-71.0	-2.0	-125.0	3.0
...	...	...	...	...	...	...	...	...	...
32241	37211	0	-206.0	41.0	26.0	-3.0	-1.0	-158.0	-6.0
32242	37212	1	93.0	90.0	24.0	73.0	8.0	53.0	-0.0
32243	37213	1	387.0	-51.0	36.0	132.0	1.0	57.0	3.0
32244	37214	1	54.0	24.0	21.0	145.0	9.0	150.0	-0.0
32245	37216	0	-247.0	22.0	-20.0	-134.0	2.0	-341.0	-5.0

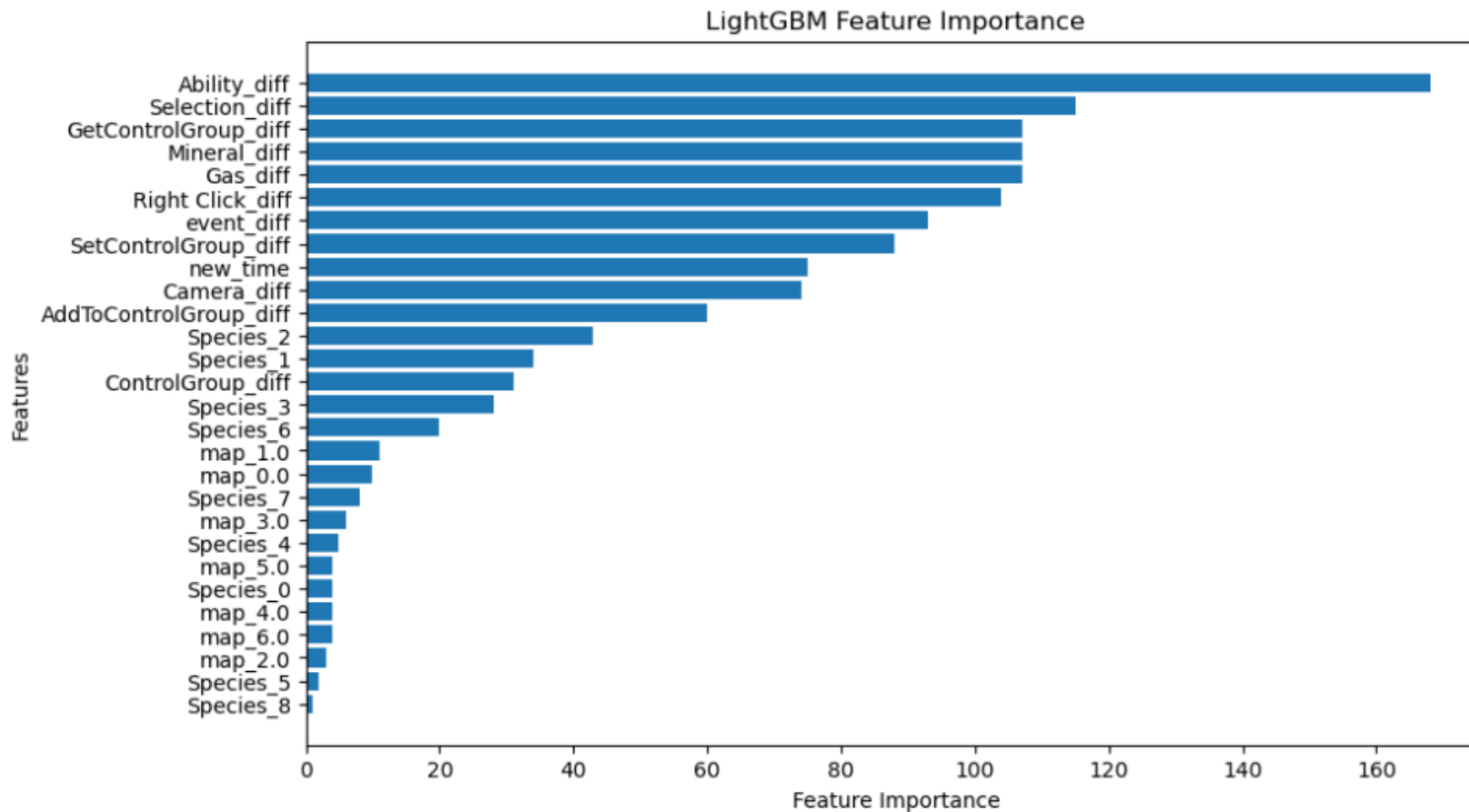
32107 rows x 31 columns

#### - 모델링

- 게임의 특성 상, 승패가 무조건 나뉘게 되고 이는 분류 문제로 볼 수 있다.
- 데이터의 크기가 크고 카테고리 변수가 많기 때문에 LightGBM 모델을 사용하기로 결정
- Proba 값을 반환해서 플레이어 1의 승률 예측값으로 사용



#### - 모델링



- 변수 중요도를 시각화

- AUC : 0.67



### - 결론 및 아쉬웠던 점

game_id	win_rate
38872	0.590833
38874	0.450816
38875	0.215902
38879	0.658672
38880	0.427555
...	...
55653	0.477422
55654	0.594860
55655	0.466596
55656	0.677540
55658	0.656173

14430 rows x 1 columns

- 대회 결과랑 비교해봤을 때 상위권 사람들의 AUC 점수보다 낮게 나왔음.
- 변수가 너무 많고 여러 행에 한 게임에 대한 정보가 들어있는 경우를 처음 해봐서 EDA나 데이터 전처리 과정이 미흡했음.
- Event\_contents 행처럼 복잡한 변수에서 새로운 파생변수를 뽑아내는 과정이나 근거를 생각하기가 힘들었음.
- 각 player들의 빌드를 하나의 변수로 활용하고 싶었지만, 게임 시간이 10분이내로 한정되어있어서 유형화가 쉽지 않았다.