

프롤로그

개인 프로젝트를 시작하며...

요즘 세대에서는 대체로 휴식시간에 스마트 폰으로 유튜브 등을 시청하며 시간을 보낸다. 나 역시 식후 또는 휴식시간 동안 스마트폰으로 유튜브를 틀며 여러 영상들을 시청한다. 유튜버의 액션, 시청자의 반응 등을 통해 웃으면서 킬링 타임을 한다. 다양한 유튜브 영상 중 내가 주로 보는 것은 소싯적 해보았던 ‘스타크래프트’이다. 지금은 스타크래프트의 인기가 비교적 낮아졌을지 몰라도 예전에 스타크래프트의 전성기는 누구나 부정할 수 없을 것이다. 전 프로들의 방송 홍구, 흑운장, 매튜브, 안기호TV 등을 보며 시간을 보낸다. 이런 유튜버들은 재밌게 시청자들을 참 교육 시키거나 때로는 역으로 시청자에게 지기도 하지만 플레이 실력은 누가 보아도 잘 한다고 말 할 것이다. 플레이를 볼 때 마다 재밌게 플레이를 하는구나, 어떻게 저렇게 잘 할 수 있을까 하는 감탄 등을 하며 감상 중이다. 유튜버의 빠른 손놀림과 함께 화면을 보던 중 APM이라는 것이 눈에 들어왔다. 뭔가 키보드의 빠름을 인지하는거 같은데 하는 느낌이 들었고 이러한 지표의 높음이 고수가 아닐까라고 고민을 했다. 그러던 중 문득 캐글에 스타크래프트에 대한 데이터 셋이 있는 것을 보고 이번 프로젝트를 Skill Craft(스타크래프트2-리플레이 영상)에 대한 데이터로 정하여 본격적으로 데이터 분석을 시작하는데.....

데이터 출처: <https://www.kaggle.com/sfu-summit/starcraft-ii-replay-analysis>

참고: 논문에서의 표기, 맞춤법 등 여러 가지 부분들이 틀릴 수 있으며 기타 문의사항에 대해서는 메일로 문의 바랍니다.

스타크래프트 고수의 비법

(스타크래프트로 세상을 통찰하기)

백경민¹

bkmin0215@naver.com

¹서울대학교 핀테크 전문가 양성과정

Abstract

본 연구에서는 스타크래프트2를 티어 별로 분류 할 뿐 아니라 수준 별 차이의 특징에 대해 파악하는 것이다. 티어 별 적절성을 고려하여 스타크래프트2 티어에 대하여 다시 세 개의 수준으로 분류하였다. 첫 번째 수준은 초보자로 브론즈, 실버, 골드이며 두 번째 수준은 중급자로 플래티넘 다이아이며 세 번째 수준은 전문가로 마스터, 그랜드 마스터, 프로로 구분하였다. 이러한 세 개의 수준에 대하여 비교한 방식으로는 각 수준에 대해 OvO(One verse One) 방식, 모든 수준에 대한 비교 OvR(One verse Rest) 방식, 마지막으로 중급자 이하와 전문가로 사전 연구와 비교하였다. 머신러닝 모델은 XGBoost로 선택하였으며 Grid Search 와 early stopping을 활용하여 hyper parameter를 정교하게 세팅하였다. 이번 연구에서는 스타크래프트2 의 변수 중 APM에 대하여 다양하게 분류 하였다. 또한 일부 변수의 경우 인지과학적인 측면에서 측정한 단위가 맞지 않음을 발견하였다. 따라서 변수의 단위 타임스탬프(time stamp)에 대해 일부 변수의 경우 단위를 분으로 바꾸어 분석을 실시하였다. 모델의 평가 지표로는 샘플의 차이가 많이 나는 경우 SMOTE 샘플링 기법으로 aucpr(area under the Precision Recall curve)을 사용하였으며 그렇지 않은 경우 auc(area under the ROC Curve)를 사용하였다.

Key words: multi classification, supervised, starcraft2, SMOTE, OvO(One verse One), OvR(One verse Rest), aucpr, auc, XGBoost, early stopping, grid search

1. Introduction

Skill Craft에 대해 사전연구는 인지과학과 관련된 연구 들로서 Joseph J., Mark R, et al[1, 2]에 대해 소개한다. [1]에서는 나이에 따른 회귀분석의 방법론을 사용하여 데이터를 분석하였다. [1]의 결과에서는 24살부터 인지 운동에 대한 반응시간이 느려짐을 통계적으로 보였다. 그러나 나이가 많은 플레이어에 대해서는 단축키의 활용, 적절한 전략을 사용 등 경험에 의해 이러한 감소를 간접적으로 보상할 수 있다고 가정하였다. 그러나 전문지식이 영역별 인지 저하를 약화시킨다는 것에 대한 증거를 찾지 못했다. 또한 나이와 관련한 리그에서의 수준의 차이에 대해서는 발견하지 못했다. [2]에서는 랜덤 포레스트 기법을 사용하여 데이터를 분석하였다. 그 결과 인지과학에서 전문가와 초보의 차이는 기술을 습득하는 전반에 있어서 차이가 나는 것이 아니며 특성의 차이도 정해져 있지 않다는 것을 보였다. 즉, 초보자와 전문가의 차이를 결정하는 특성은 중급자와 전문가의 차이를 결정함에 있어 덜 중요하다. 스타크래프트의 데이터를 선정한 이유는 체스와 비교하여 움직임이 약 40배 이상 차이가 난다는 것을 관찰하였다. 이는 지속적인 인지 운동 수행을 관찰할 수 있으며 전문 지식을 이해하는 데 중요하다고 볼 수 있다.

다음으로는 스타크래프트 데이터 수집 및 용어에 관한 설명이다.

데이터 수집 - 플레이어마다 고유한 특성을 반영하지 않기 위해 플레이어 개인에 대한 리플레이 영상을 수집하였다. 시간은 스타크래프트2 리플레이 파일의 타임스탬프로 기록된다. 게임이 빠르게 재생될 때 실시간 1초는 대략 88.5개의 타임스탬프에 해당한다.[3]

데이터에서의 용어를 이해하기 위해 논문[2] 또는 자료[4]를 참고하여 설명한다.

PAC(Perception-Action-Cycle variables): PAC는 지각-행동-주기라고 불리며 각 변수는 플레이어가 특정 위치에 고정되어 행동하는 기간과 관련이 있습니다. PAC는 기본적으로 일정 시간 동안 새로운 위치로 화면을 이동한 후 최소한 하나의 작업(일반적으로 4개 에서 6개)을 수행한 다음 다른 위치로 이동하는 것으로 구성된다.

Hotkey usage variables : 단축키라고 하며 플레이어는 유닛 또는 건물을 더 빨리 컨트롤하기 위해 키를 지정합니다. 따라서 마우스로 컨트롤 하는거 보다 더 생산적이다.

Complex unit production and use variables : 특정 유닛은 이중 작업 과제를 내포하고 일부는 명시적인 지시나 표적 지시를 받아야 합니다. 이러한 유닛과 능력의 생산과 사용은 때때로 선택 사항이며, 따라서 그들의 생산과 사용은 플레이어가 자신의 인지 부하를 조절하는 것을 반영할 수 있다.

Direct measures of attentional control : 미니맵(Mini-Map) 변수는 미니맵에서 플레이어의 행동을 반영하며, 우리는 더 나은 플레이어가 이 맵에 주의를 기울이고 사용하는 데 더 나은 작업을 수행할 것이라는 가설을 세웠습니다. 또한 게임 상태에 대한 정보를 찾는 것과 관련이 있다고 생각하는 플레이어가 전체 맵에서 얼마나 많이 보았는지 고려한다.

Actions Per Minute (APM) : 1분당 명령의 횟수로 스타크래프트 커뮤니티의 전문성과 게임에서 자동으로 계산됩니다. 인지 운동 속도의 척도이다.

스타크래프트의 경험을 살려 사전연구에 대해 미흡한 부분을 지적하고 연구의 차별성에 대해 설명한다. 첫째, 데이터에 대한 이해이다. 변수의 상관관계 및 산점도를 이해하지 않으면 통계적인 문제인 다중공선성 뿐 아니라 선후 관계의 변수들을 같이 분석하게 된다. 충분한 데이터의 이해가 있어야 이러한 문제를 해결 할 수 있다. 둘째, 일변수에 대한 분석의 부재이다. 데이터의 수집 후 인지 과학에 대한 지표를 중점적으로 분석하였음을 보았다. 대표적으로 TotalHours, 총 플레이 한 시간이다. 수준에 따라 당연히 플레이에 투자한 시간의 차이가 있을 수 있지만 논문에서는 해당 변수에 대해서는 제외하였다. 이와 더불어 단위의 문제를 들 수 있다. 일부 변수의 데이터 단위가 타임스탬프이며 이는 1초를 88.5개로 나누어 분석하고 있다. 그렇다면 타임스탬프 간을 기준으로 잡으면 실제 플레이에서의 클릭 수를 예로 들자면 당연히 적어질 수 밖에 없다. 따라서 일부 변수의 단위 수정이 필요하다고 생각한다. 셋째, 구간의 비교에 대한 적절성이다. [2]의 분석 결과에서 Bronze-Professional을 비교한 것을 볼 수 있다. 그러나 실제 스타크래프트 커뮤니티[5, 6, 7]를 참고한다면 적절히 구간을 다시 구분할 필요가 있다. Bronze, Silver, Gold를 초보자, Platinum, Diamond를 중수, Master, Grand Master, Professional을 전문가라고 수준별로 다시 그룹화를 하였다. 초보자와 중급자간의 차이 또한 중수와 고수의 차이 그리고 중수 이하와 전문가의 차이 역시 분석할 것이다. 마지막으로 초보자부터 전문가까지 전 구간에서 다분류 분석을 하여 어떤 특징들에 대해서 전반적인 차이점들이 있는지 분석할 것이다. 마지막으로 지표에 대한 부재이다. 머신러닝 기법을 적용하면 분류 문제의 경우 재현율, 정밀도, 정확도 등 지표를 구하게 되는데 사전 연구에서는 결과가 서술되어 있지 않고 특징 별 중요도만 표시되어 있다.

결과적으로 데이터는 분석하고자 하는 목적에 따라 달라질 수 있다. 사전 연구에서 주로 인지과학적인 측면에서 분석하였다면 이번 연구에서는 실질적인 수준에 대한 비교를 통해 데이터를 분석할 것이다.

2. EDA (Exploratory Data Analysis)

아래 그림은 데이터의 설명 및 특징이다.

Feature	Description	설명	특징
GameID	Unique ID number for each game	플레이어의 고유한 ID	
LeagueIndex	Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional leagues coded 1-8	수준에 따른 티어	Bronze, Silver, Gold = 0 (초보) Platinum, Diamond = 1 (중수) Master, GrandMaster, Pro = 2(전문가)
Age	Age of each player	플레이어의 나이	
HoursPerWeek	Reported hours spent playing per week	주 별 플레이하는 시간	노력
TotalHours	Reported total hours spent playing	플레이어에 보낸 총 시간	경력
APM	Action per minute	1분당 명령의 횟수	컨트롤
SelectByHotkeys	Number of unit or building selections made using hotkeys per timestamp	time stamp마다 단축키를 사용하여 유닛이나 건물을 선택하는 횟수	컨트롤
AssignToHotkeys	Number of units or buildings assigned to hotkeys per timestamp	타임스탬프마다 유닛이나 건물에게 단축키를 할당하는 횟수	컨트롤
UniqueHotkeys	Number of unique hotkeys used per timestamp	타임스탬프마다 사용된 고유한 단축키의 횟수	컨트롤
MinimapAttacks	Number of attack actions on minimap per timestamp	타임스탬프마다 미니맵에서 공격한 횟수	상황 판단
MinimapRightClicks	number of right-clicks on minimap per timestamp	타임스탬프마다 미니맵에서 오른쪽 클릭을 한 횟수	상황 판단
NumberOfPACs	Number of PACs per timestamp	타임스탬프마다 PACs 횟수	반응 속도
GapBetweenPACs	Mean duration in milliseconds between PACs	PACs 사이의 평균 밀리세컨드 시간	반응 속도
ActionLatency	Mean latency from the onset of PACs to their first action in milliseconds	PACs에서 첫 번째 행동이 시작되기 까지 걸리는 평균 밀리세컨드 시간	반응 속도
ActionsInPAC	Mean number of actions within each PAC	각 PAC 내에서 행동들의 평균 횟수	반응 속도
TotalMapExplored	The number of 24x24 game coordinate grids viewed by the player per timestamp	타임스탬프마다 플레이어가 본 24*24 크기의 맵에서 그리드 수	정찰 능력 및 시야 확보
WorkersMade	Number of SCVs, drones, and probes trained per timestamp	타임스탬프마다 Scv, 드론, 프로브를 생산한 수	생산
UniqueUnitsMade	Unique unites made per timestamp	타임스탬프마다 생산한 고유한 유닛 수	생산
ComplexUnitsMade	Number of ghosts, infestors, and high templars trained per timestamp	타임스탬프마다 생산한 고급 유닛의 수	빌드
ComplexAbilitiesUsed	Abilities requiring specific targeting instructions used per timestamp	타임스탬프마다 특정한 대상이 되는 명령들을 요구하는 능력	빌드
MaxTimeStamp	Time stamp of game's last recorded event	플레이어가 가장 최근에 기록한 타임스탬프, 플레이어 타임	플레이어 시간

Figure 1 데이터의 설명 및 특징

스타크래프트의 특성을 반영하여 데이터는 노력, 경력, 컨트롤, 상황 판단, 반응 속도, 정찰 능력, 생산, 빌드, 플레이 시간으로 구분 할 수 있습니다.

아래의 변수들은 단위를 타임스탬프에서 분으로 변경한다.

SelectByHotkeys, AssignToHotkeys, UniqueHotkeys, MinimapAttacks, MinimapRightClicks, TotalMapExplored, WorkersMade, UniqueUnitsMade, ComplexUnitsMade, ComplexAbilitiesUsed, MaxTimeStamp

Over View를 하기 전 GameID, 고유의 플레이어는 분석 대상이 아니므로 제외하며 Age는 사전 연구에서 분석 하였으므로 제외한다.

Over View

Correlation Plot

아래는 각 변수에 대한 상관관계를 나타낸 것이다.

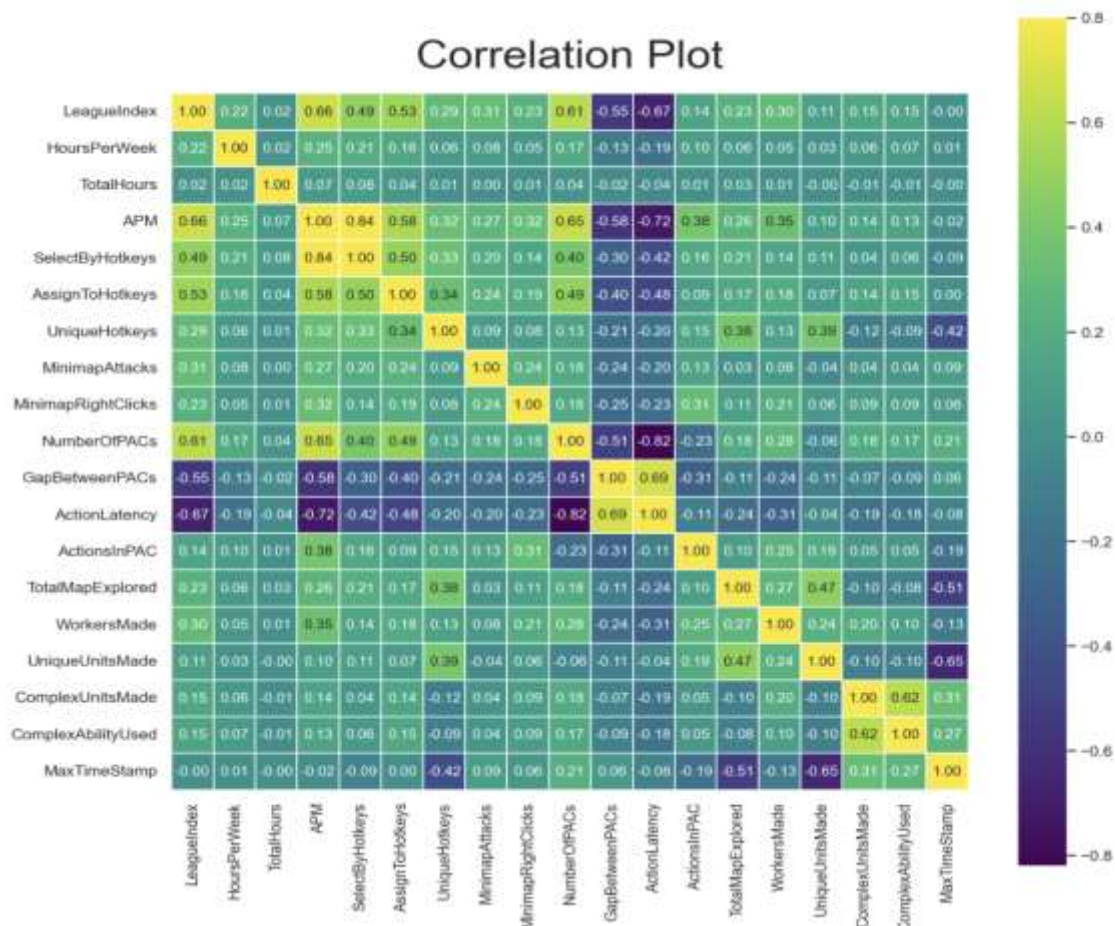


Figure 2 Correlation Plot

우선 상관관계가 상대적으로 큰 경우 중 APM과 관련된 변수로는 SelectByHotkeys, AssignToHotkeys, UniqueHotkeys, NumberOfPACs, GapBetweenPACs, ActionLatency이다. APM이 1분간 명령의 횟수로 컨트롤, 빠르기를 의미하는데 APM이 높을수록 단축키의 활용도가 높을 것이며 반대로 인지 후 행동까지의 시간 등은 짧을 것이다. 이렇듯 컨트롤에서의 빠르기를 의미하는 변수들과 반응 시간 간에는 음의 상관관계가 나타난다.

ComplexUnitsMade와 ComplexAbilitiesUsed는 상대적으로 높은 관련성이 있는데 이는 고스트와 같은 상위 유닛의 생산을 하려면 생산 건물을 지어야 하기 때문이다.

MaxTimeStamp는 플레이어가 가장 최근에 기록한 타임스탬프이다. 이 변수에 대한 의미를 이해하기 위해서 다른 변수와의 상관관계를 보면 UniqueHotkeys, TotalMapExplored, UniqueUnitsMade와는 반비례 이며 상대적으로 ComplexUnitsMade, ComplexAbilitiesUsed와는 비례하는 것을 볼 수 있다. 이는 플레이 타임이 길수록 단일 유닛, 단일 단축키에 대한 사용, 맵 정찰 등은 낮아지고 상위 유닛이나 생산건물은 사용이 높아짐을 볼 수 있다.

종속변수인 LeagueIndex는 APM, SelectByHotkeys, AssignToHotkeys, NumberOfPACs와 양의 상관관계에 있으며 GapBetweenPACs와 ActionLatency는 음의 상관관계를 보인다.

VIF

산점도는 지면 상 생략하지만 상관관계 플롯에서 알 수 있듯이 일부 변수에서 선형성을 나타냄을 볼 수 있다. 따라서 VIF로 다중공선성을 확인하여 변수를 선택한다.

	VIF Factor	features
0	1.000000	Intercept
1	1.008356	scale(TotalHours)
2	1.076999	scale(HoursPerWeek)
3	1.125616	scale(MinimapAttacks)
4	1.290389	scale(MinimapRightClicks)
5	1.434336	scale(WorkersMade)
6	1.488890	scale(UniqueHotkeys)
7	1.611904	scale(AssignToHotkeys)
8	1.681957	scale(ComplexAbilityUsed)
9	1.741917	scale(TotalMapExplored)
10	1.797297	scale(ComplexUnitsMade)
11	1.938547	scale(UniqueUnitsMade)
12	2.296699	scale(GapBetweenPACs)
13	2.578806	scale(MaxTimeStamp)
14	5.386913	scale(ActionLatency)
15	8.085926	scale(ActionsInPAC)
16	12.457887	scale(SelectByHotkeys)
17	13.537011	scale(NumberOfPACs)
18	37.038384	scale(APM)

Figure 3 VIF 변경 전

	VIF Factor	features
0	1.000000	Intercept
1	1.008084	scale(TotalHours)
2	1.073893	scale(HoursPerWeek)
3	1.124224	scale(MinimapAttacks)
4	1.246872	scale(MinimapRightClicks)
5	1.329110	scale(WorkersMade)
6	1.486254	scale(UniqueHotkeys)
7	1.590969	scale(AssignToHotkeys)
8	1.623923	scale(ActionsInPAC)
9	1.671671	scale(ComplexAbilityUsed)
10	1.732678	scale(TotalMapExplored)
11	1.796942	scale(ComplexUnitsMade)
12	1.913955	scale(UniqueUnitsMade)
13	2.267552	scale(GapBetweenPACs)
14	2.492016	scale(MaxTimeStamp)
15	3.299241	scale(APM)
16	3.500570	scale(ActionLatency)

Figure 4 VIF 변경 후

VIF를 참고하여 변수 선택 시 APM과 유사한 NumberOfPACs, SelectByHotkeys를 제외한다.

One Variable

LeagueIndex

LeagueIndex에 대하여 티어 별 개수, 비율을 확인하고 LeagueLevel을 생성하여 다시 세개의 수준으로 티어를 그룹화 한다.

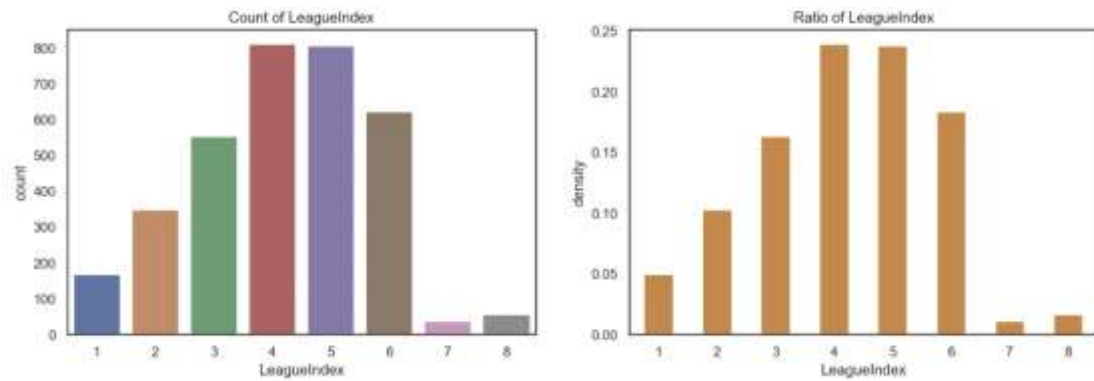


Figure 5 LeagueIndex Count Plot & Density Plot

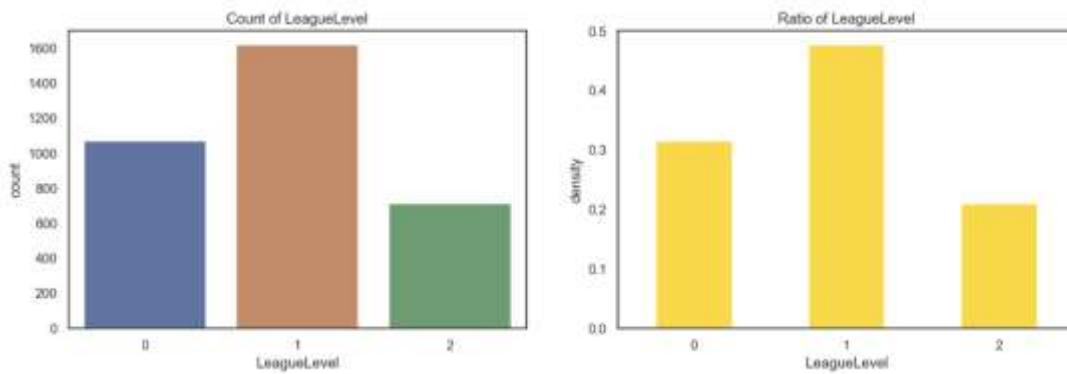


Figure 6 LeagueLevel Count Plot & Density Plot

LeagueLevel에서 보자면 수준별로 중급자(1)에서 가장 많은 빈도를 나타내고 있으며 다음으로 초보자(0), 전문가(2)이다.

HoursPerWeek

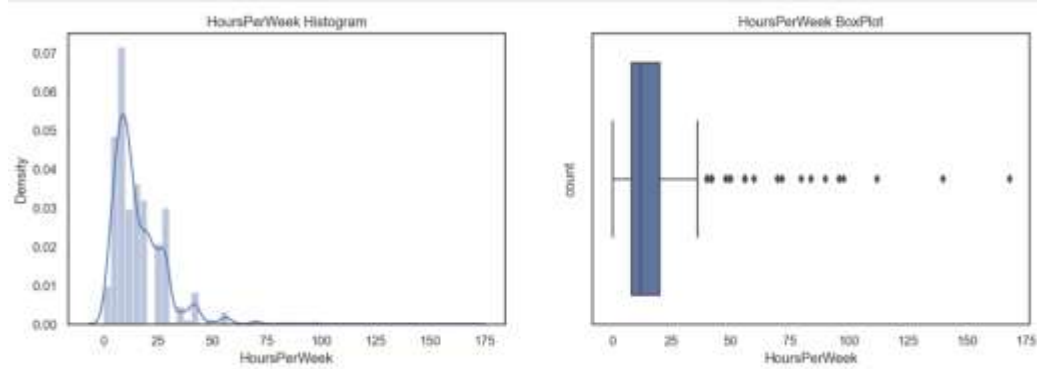


Figure 7 HoursPerWeek Density Plot & Box Plot

프로게이머 기준 평균 하루 14시간으로 정하여 100시간 이상은 이상치로 간주한다.

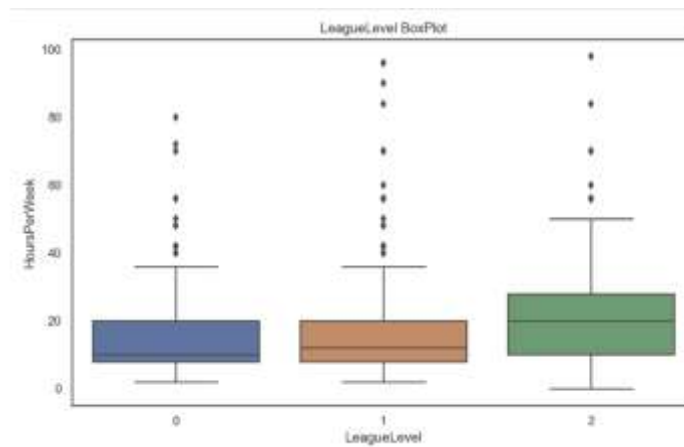


Figure 8 LeagueLevel – HoursPerWeek Box Plot

위의 그래프는 LeagueLevel에 대한 HoursPerWeek의 Box Plot을 나타낸 것이다. 이상치의 기준에 대해 수준0에서는 하루 6시간*7 = 42시간, 수준1에서는 하루 8시간*7 = 56시간으로 하여 해당 기준 초과 시 이상치로 판단한다.

TotalHours

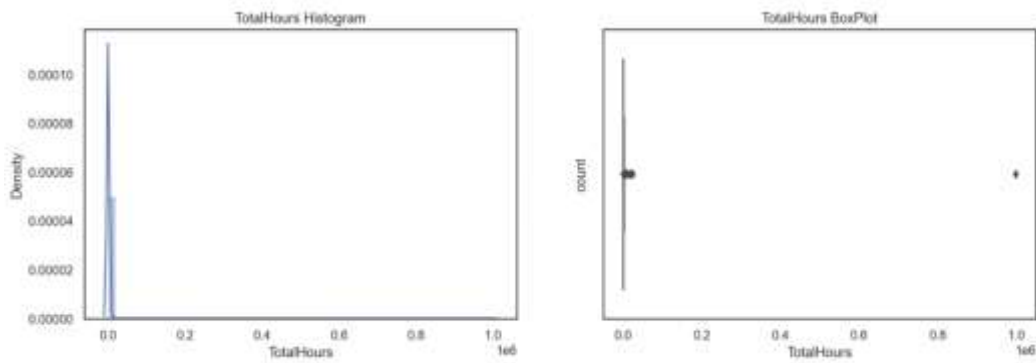


Figure 9 TotalHours Density Plot & Box Plot

TotalHours의 Histogram 및 Box Plot을 나타낸 것이다. 우선 가장 극단값인 1개를 제거하고 다시 분석한다.

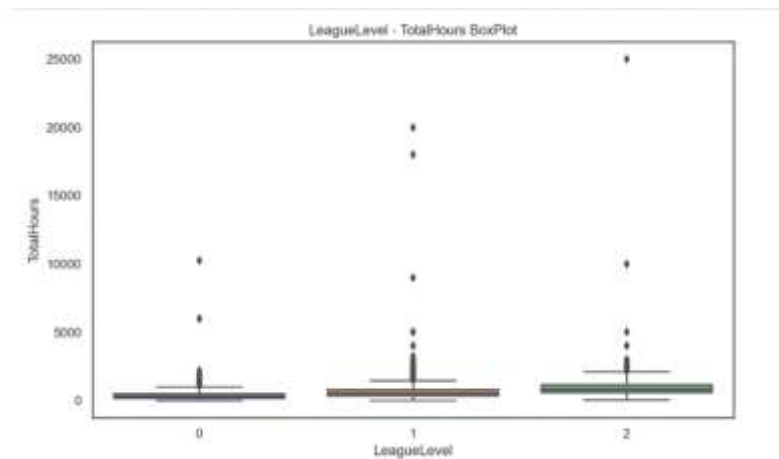


Figure 10 LeagueLevel – TotalHours Box Plot

수준 별 이상치의 기준에 대해서는 수준2에서 상위 1개의 극단값은 티어를 보면 마스터이므로 가장 높은 티어가 아니다. 그러므로 일반적인 기준 1만시간으로 정하여 이상치로 정한다. 또한 수준0에서는 5000시간, 수준1에서는 9000시간 이상인 경우 이상치로 판단한다.

APM

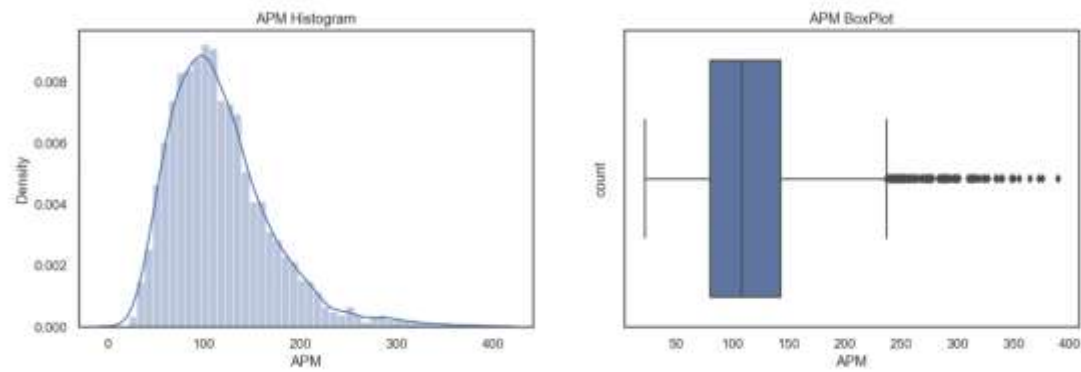


Figure 11 APM Density Plot & Box Plot

APM에 대한 Histogram과 Box Plot을 나타낸 것이다. 극단값에 해당하는 이상치는 없는 것으로 보인다. 다음으로 수준 별 APM에 대한 Box Plot을 확인한다.

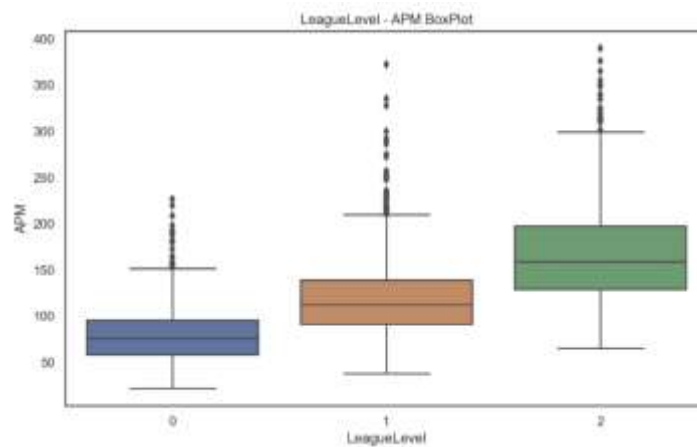


Figure 12 LeagueLevel – APM Box Plot

수준 별 APM의 이상치 기준으로는 프로 기준 APM이 300정도 이므로 300이 넘는 APM에 대해서 이상치로 판단한다.

AssginToHotkeys

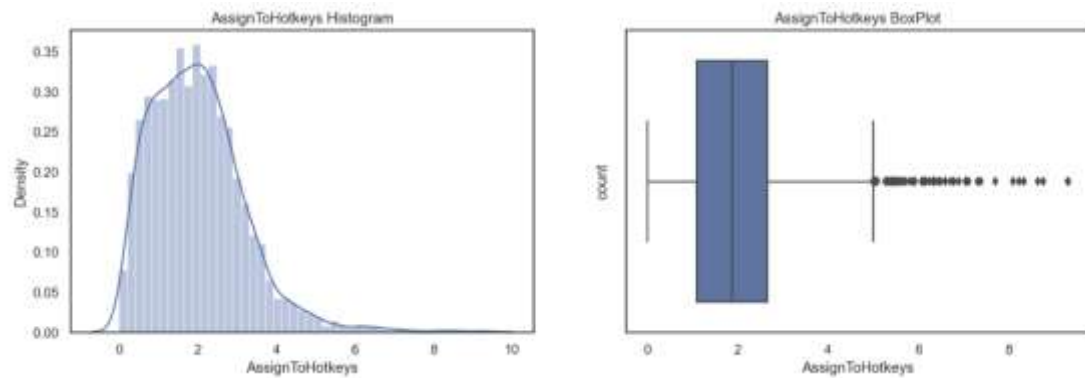


Figure 13 AssginToHotkeys Density Plot & Box Plot

AssginToHotkeys에 대해 Histogram과 Box Plot을 나타낸 것이다. 극값에 해당하는 수준에 대해 알아보기 위해 다시 수준 별 Box Plot으로 확인한다.

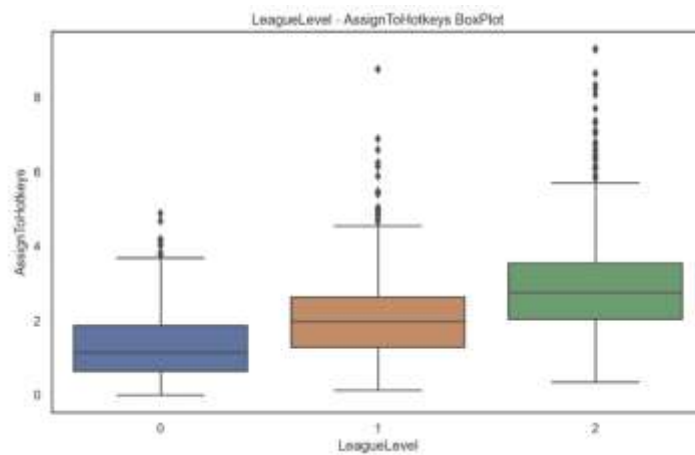


Figure 14 LeagueLevel – AssginToHotkeys Box Plot

위의 Box Plot으로 이상치를 살펴보면 수준 1에서 상위 1개의 극단값을 이상치라고 판단한다.

UniqueHotkeys

UniqueHotkeys에 대해 수준 별 Box Plot을 확인해서 이상치를 판단한다.

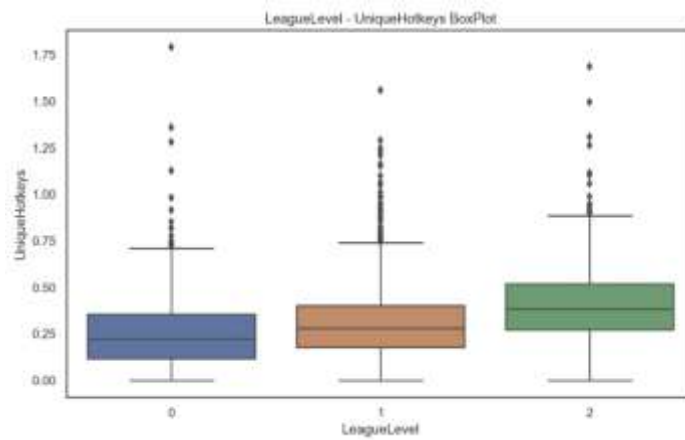


Figure 15 LeagueLevel – UniqueHotkeys Box Plot

수준 별 이상치를 정하는 기준으로 수준0에서는 분당 1회, 수준1에서는 상위 1개의 극단값을 이상치로 정한다.

MinimapAttacks

MinimapAttacks에 대해 수준 별 Box Plot을 확인해서 이상치를 판단한다.

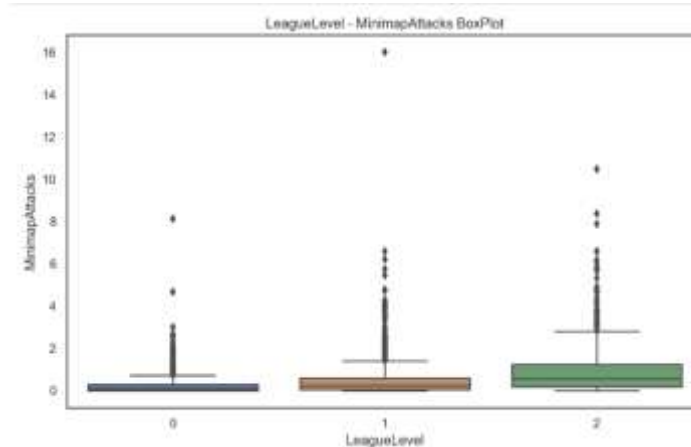


Figure 16 LeagueLevel – MinimapAttacks Box Plot

이상치 판단 기준으로 수준0에서 상위 2개, 수준1에서 상위1개를 이상치라고 판단한다.

MinimapRightClicks

MinimapRightClicks에 대해 수준 별 Box Plot으로 이상치를 확인한다.

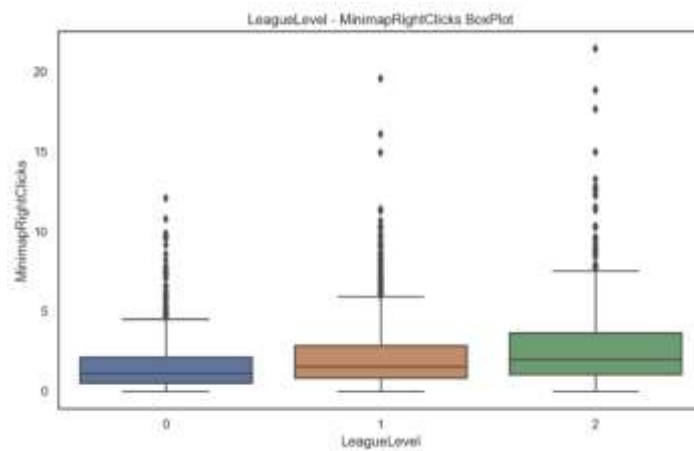


Figure 17 LeagueLevel – MinimapRightClicks Box Plot

수준0에서 상위 2개, 수준1에서 상위 3개를 이상치로 판단한다.

GapBetweenPACs

GapBetweenPACs에 대해 수준 별 Box Plot으로 이상치를 판단한다.

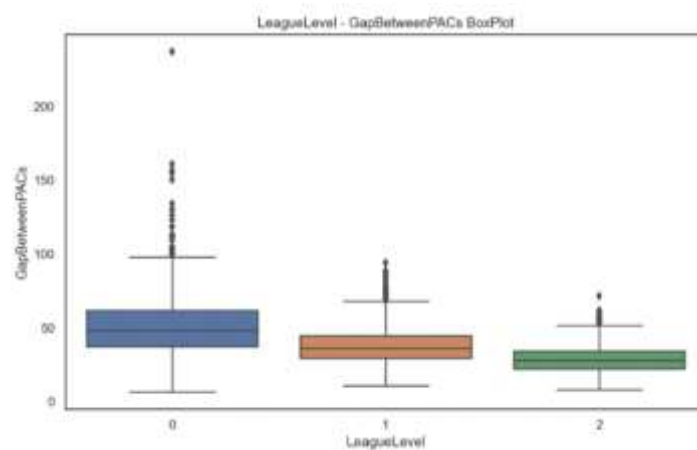


Figure 18 LeagueLevel – GapBetweenPACs Box Plot

수준0에서 상위 1개를 이상치로 판단한다.

ActionLatency

ActionLatency에 대해 수준 별 Box Plot으로 이상치를 확인한다.

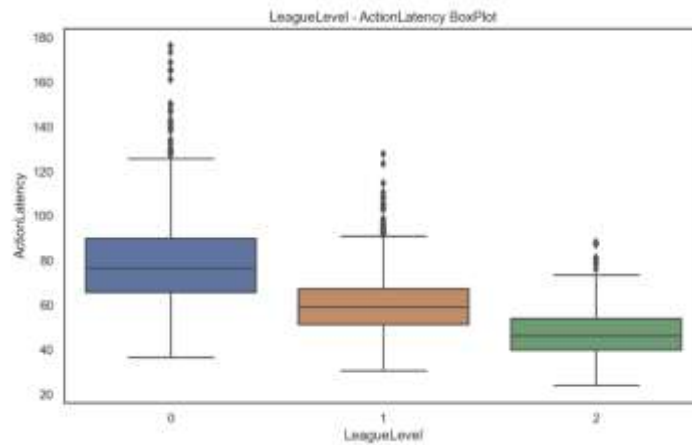


Figure 19 LeagueLevel – ActionLatency Box Plot

Box Plot에서 수준 별 극단값이 관찰되지 않아 이상치는 없는 것으로 판단한다.

ActionsInPAC

ActionsInPAC에 대해 수준 별 Box Plot으로 이상치를 확인한다.

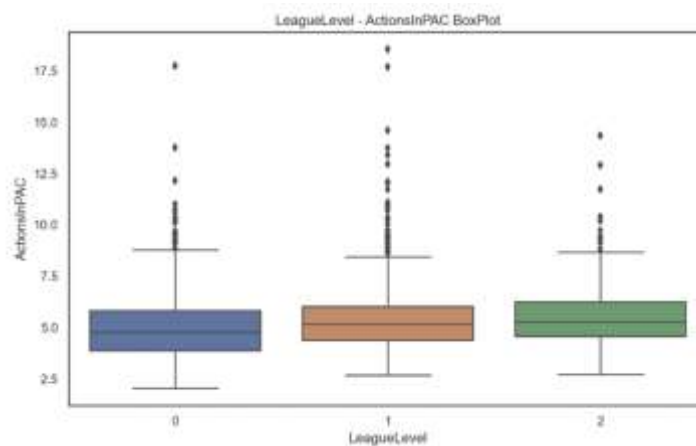


Figure 20 LeagueLevel – ActionsInPAC Box Plot

프로보다 PAC 내 행동의 횟수 높다는 것에 대해 이상치로 정하며 수준0에서는 상위 3개, 수준1에서도 상위 3개의 값에 대해 이상치로 판단한다.

TotalMapExplored

TotalMapExplored에 대해 수준 별 Box Plot으로 이상치를 확인한다.

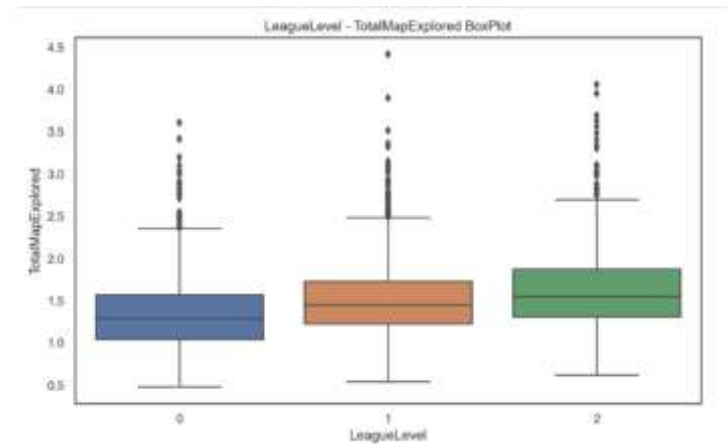


Figure 21 LeagueLevel – TotalMapExplored Box Plot

전문가보다 TotalMapExplored이 높다는 것에 대해 이상치로 정한다. 추가로 수준0에서 분당 약 3.2회 이상 수준1에서 약 4회 이상에 대해 이상치로 판단한다.

WorkersMade

WorkersMade에 대해 수준 별 Box Plot으로 이상치를 확인한다.

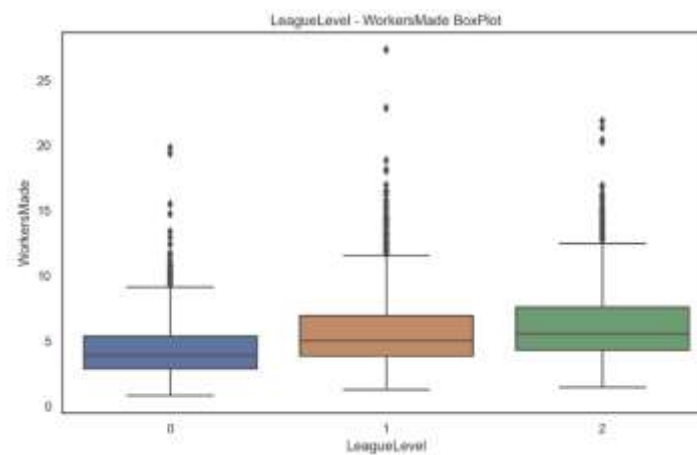


Figure 22 LeagueLevel – WorkersMade Box Plot

전문가보다 WorkersMade가 높다는 것에 대해 이상치로 정한다. 따라서 수준0에서 상위 2개, 수준1에서 상위 2개에 대해 이상치로 판단한다.

UniqueUnitsMade

UniqueUnitsMade에 대해 수준 별 Box Plot으로 이상치를 확인한다.

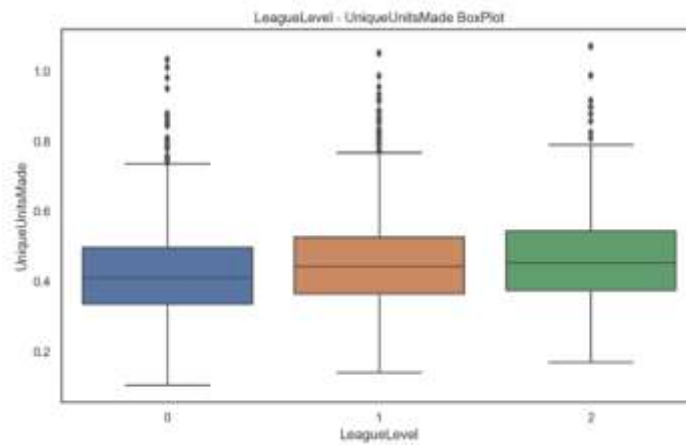


Figure 23 LeagueLevel – UniqueUnitsMade Box Plot

UniqueUnitsMade는 수준에 관계없이 단일 유닛을 생산하는 것에서 이상치는 없다고 판단한다. 예를 들어 테란의 경우 극단적으로 사신만 뽑는 경우도 있다.

ComplexUnitsMade

ComplexUnitsMade에 대해 수준 별 Box Plot으로 이상치를 확인한다.

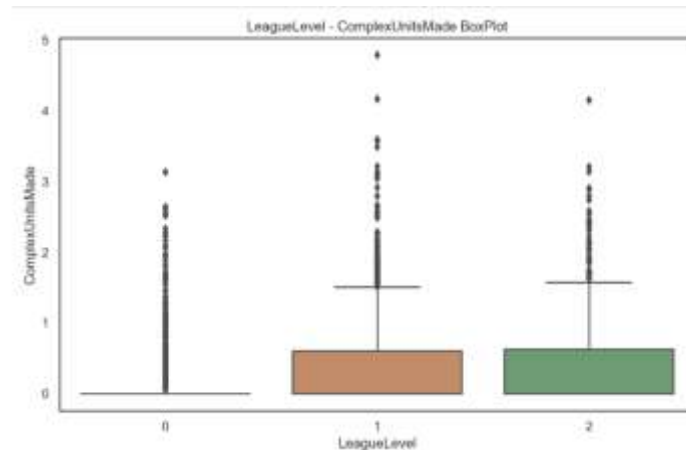


Figure 24 LeagueLevel – ComplexUnitsMade Box Plot

전문가보다 ComplexUnitsMade가 높다는 것에 대해서 이상치로 정한다. 따라서 수준0에서 상위 1개, 수준1에서 상위 2개에 대해 이상치로 판단한다.

ComplexAbilityUsed

ComplexAbilityUsed에 대해 수준 별 Box Plot으로 이상치를 확인한다.

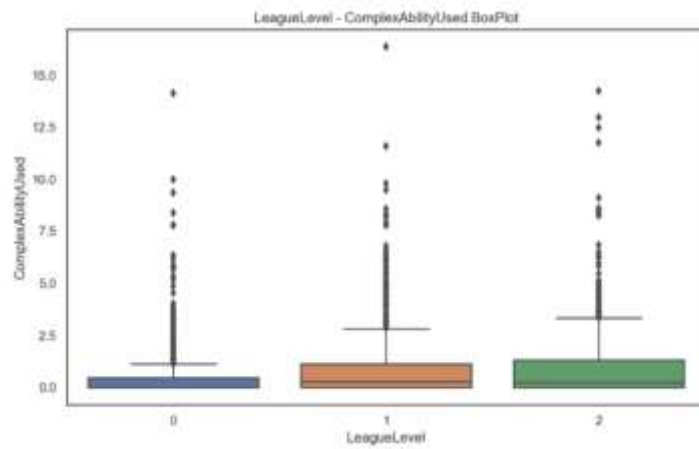


Figure 25 LeagueLevel – ComplexAbilityUsed Box Plot

상위 유닛을 생산하기 위한 건물에 대해서 수준0에서는 분당 약 7.8개, 수준1에서는 분당 약 9.5개 이상 데이터에 대해서 이상치로 판단한다.

MaxTimeStamp

MaxTimeStamp는 플레이 타임으로 볼 수 있는데 앞서 설명했듯이 MaxTimeStamp가 커질수록 UniqueHotkeys, TotalMapExplored, UniqueUnitsMade와 음의 상관관계를 나타내고 있으며 ComplexUnitsMade와는 양의 상관관계를 나타내고 있다. 즉, 플레이 타임이 짧을수록 단일 유닛의 생산, 단축키, 정찰 능력은 많아진다. 반대로 플레이 타임이 길수록 비교적 상위 유닛의 생산이 필요하다는 것을 알려준다. MaxTimeStamp의 단위가 타임스탬프인데 이를 단위 분으로 변경하여 변수들에 대해 산점도를 확인한다. 생성된 변수는 PlayTime이다.

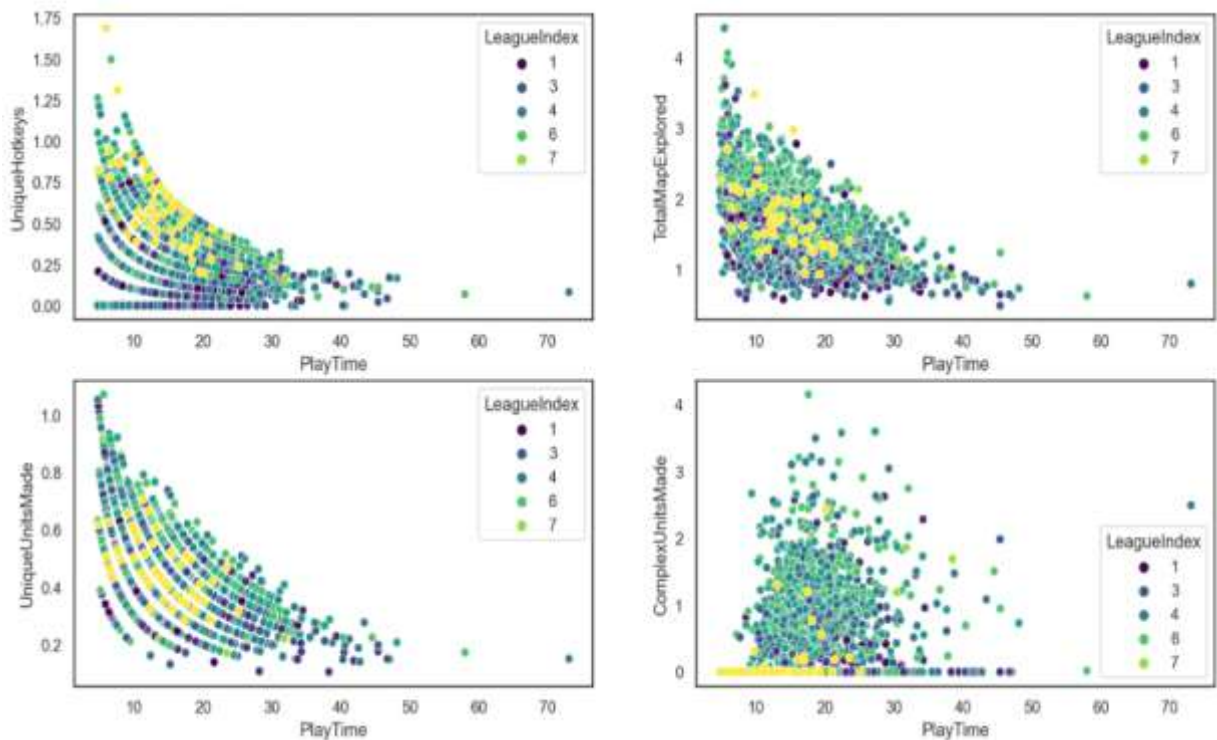


Figure 26 PlayTime with 4 variables Scatter Plot

Insight

LeagueLevel에 대하여 각 수준 별로 변수를 분석하였다. 모델을 개발하여 데이터를 학습하기 전 각 변수의 평균을 구하고 어떤 변수가 수준에 영향을 미치는지 알아본다.

LeagueLevel	0	1	2
LeagueIndex	2.358601	4.495256	6.205382
HoursPerWeek	13.026239	14.691139	21.145929
TotalHours	377.574344	650.409120	980.915515
APM	79.535070	117.932720	168.474692
AssignToHotkeys	1.309845	1.996437	2.952593
UniqueHotkeys	0.245920	0.306184	0.412318
MinimapAttacks	0.244622	0.490687	0.963810
MinimapRightClicks	1.509488	2.080183	2.674176
GapBetweenPACs	51.608287	37.889884	28.997210
ActionLatency	79.543422	60.427185	47.508314
ActionsInPAC	4.940105	5.317300	5.477614
TotalMapExplored	1.356172	1.518435	1.652331
WorkersMade	4.383986	5.745108	6.388899
UniqueUnitsMade	0.426111	0.454577	0.466198
ComplexUnitsMade	0.168749	0.364651	0.391152
ComplexAbilityUsed	0.426366	0.803060	0.969491
MaxTimeStamp	84137.912536	83348.888046	83330.827195
PlayTime	15.845181	15.696589	15.693188

Figure 27 Mean of Grop By LeagueLevel

초보자(LeagueLevel0), 중급자(LeagueLevel1), 전문가(LeagueLevel2)를 그룹화 하여 각 변수에 대한 평균값을 보면 컨트롤, 생산, 빌드 등 전반적으로 전문가에서 더 빠름을 알 수 있다. 그리고 인지 시점부터 반응까지의 시간, 인지 시점의 간격 등에 관한 변수는 전문가에서 더 짧은 시간을 보임을 알 수 있다.

마지막으로 이상치 비율은 $(3395 - 3316) / 3395$ 로 약 2.3% 정도이다. 2~3% 사이로 제거한 이상치는 적절하다.

2. Preprocessing

전처리에서는 결측치를 다루고, 변수의 구간화를 통한 더미변수를 생성할 것이다.

결측치 처리

	Total	Prob
TotalHours	57	0.017189
HoursPerWeek	56	0.016888

Figure 28 Missing Value Table

결측치에 해당하는 변수는 TotalHours와 HoursPerWeek이며 개수와 비율을 구하였다. 해당 결측치가 어느 티어에서 존재하는지 확인하기 위해 LeagueIndex를 그룹화 하여 개수를 구한다.

	HoursPerWeek	TotalHours	APM
LeagueIndex			
1	163	163	163
2	334	334	334
3	532	532	532
4	798	798	798
5	782	781	783
6	616	616	616
7	35	35	35
8	0	0	55

Figure 29 Group By LeagueIndex Table

APM을 기준으로 하여 결측치를 확인한 결과 결측치가 존재하는 티어는 다이아 구간에서 HoursPerWeek 1개, TotalHours 2개이다. 또한 프로 구간에서 HoursPerWeek, TotalHours 55개 전체이다. 결측치를 해결하기 위해서 다이아 구간의 경우 결측치를 제외한 해당 구간에서의 평균으로 결측치를 처리 할 것이고 프로 구간의 경우에는 프로인 만큼 이전 구간인 그랜드 마스터에서의 최대값으로 결측치를 처리한다.

APM

APM의 경우 전처리에 대해 분포를 사용하여 구간을 나눈다. 수준 별 APM의 차이가 많이 나는 만큼 구간의 범위를 15%씩 정한다. 15% ~ 90%까지 6구간으로 나누었으며 90%부터 상위 5%씩 나누어 총 8구간으로 Level_0부터 Level_7까지 정한다.

PlayTime

Play Time의 경우 전처리에 대해 범위를 3등분 하여 구간을 나눈다. 플레이 시간을 짧음, 중간, 깊으로 나누기 때문에 분포 대신 범위를 사용한다.

Modeling & Result

XGBoost

모델은 XGBoost로 선정한다. XGBoost는 Boosting모델 중 하나로 연속한 약한 학습기들을 여러 개 결합하여 예측 혹은 분류 성능을 높이는 알고리즘이다. XGBoost는 다양한 loss function을 지원해 task에 따른 유연한 튜닝이 가능하다. 장점으로는 예측 성능이 좋으며 과적합 컨트롤, Early stopping기능이 있다. 단점으로는 복잡한 Hyper parameter, 반복횟수가 많은 경우 계산 시간이 오래 걸린다.[8]

XGBoost – Hyper parameter tuning[9]

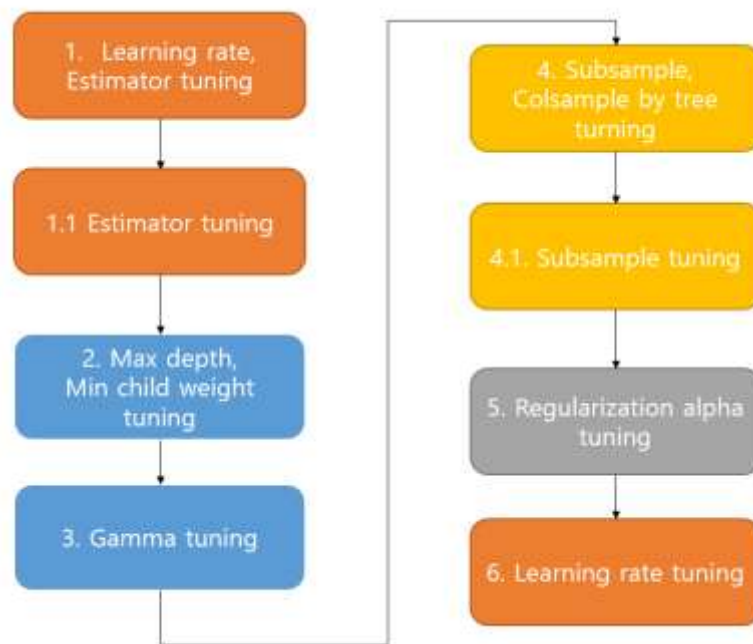


Figure 30 Hyper parameter tuning guide

Condition

Grid Search: Stratified K-fold, Cross validation 5 fold 지정

Early stopping: early_stopping_rounds 60번을 지정

Seed : 2000 ~ 2021

OvO, OvR에서 샘플링의 차이가 많이 나는 경우 SMOTE기법을 적용

OvO 평가지표: 중급자와 전문가는 aucpr, 그 외는 auc

OvR 평가지표: mlogloss

Test size : 일반적인 값인 0.2로 한다. 단, 사전연구와의 비교를 위한 OvO(중급자 이하 와 전문가의 비교)에서는 0.25로 지정한다.

OvO(One verse One)

초보자(Novice)와 중급자(Intermediate)

아래는 Seed 2000 ~ 2021에서의 정확도를 나타낸 그림이다.

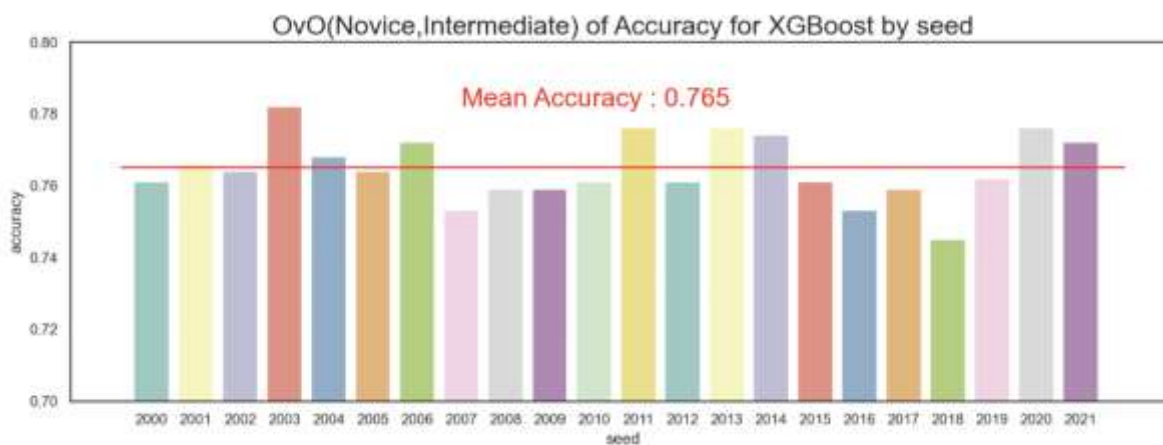


Figure 31 OvO Graph

Seed에 대한 모델의 평균 정확도는 0.765이며 정확도가 가장 높은 seed는 2003이다. 정확도가 가장 높은 seed에 대한 결과는 아래 그림에서 볼 수 있다.

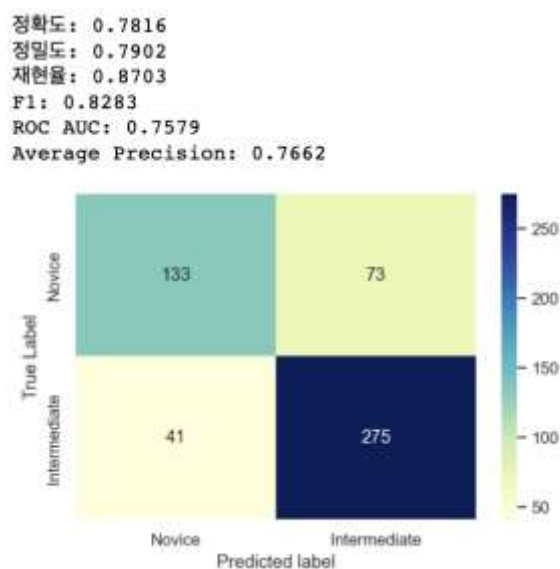


Figure 32 Confusion Matrix and Metric

Feature Importance

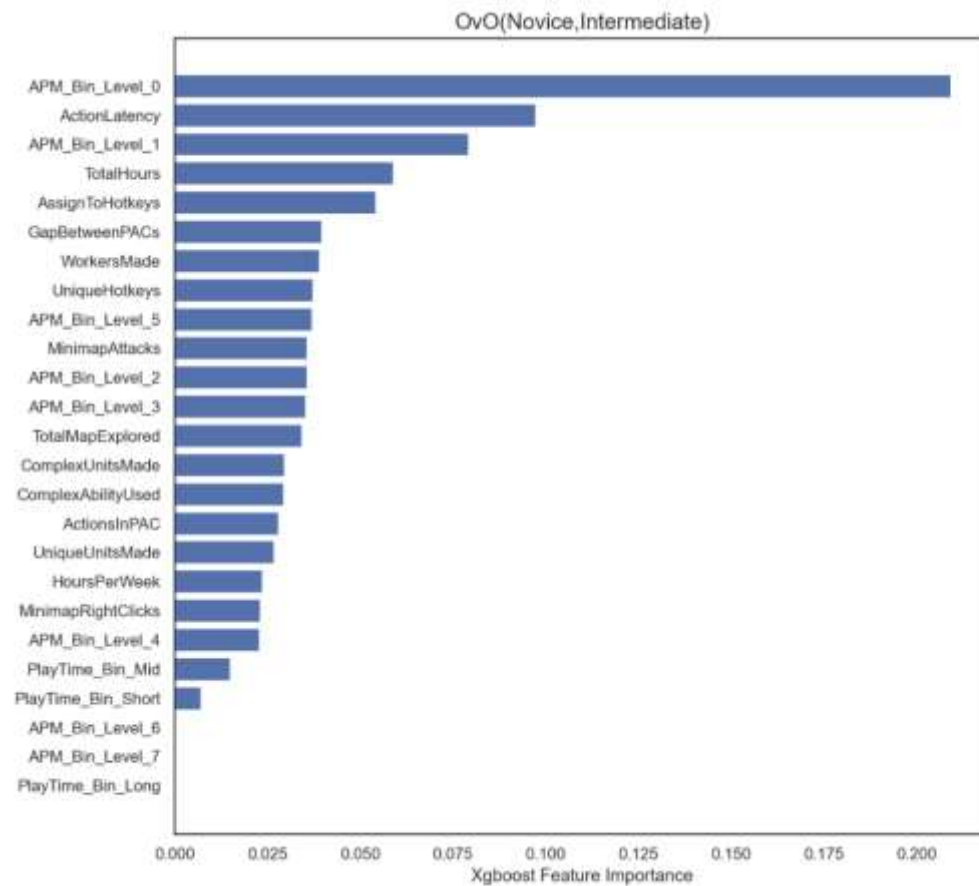


Figure 33 Feature Importance

Feature Importance의 중요도

1. APM_Level 0: 하위 15% 구간 아래, 컨트롤
2. Action Latency: 처음 인지부터 행동까지 걸린 시간, 반응 속도
3. APM_Level 1: 하위 15% ~ 30% 구간, 컨트롤

OvO(One verse One)

중급자(Intermediate)와 전문가(Expert)

아래는 Seed 2000 ~ 2021에서의 정확도를 나타낸 그림이다.

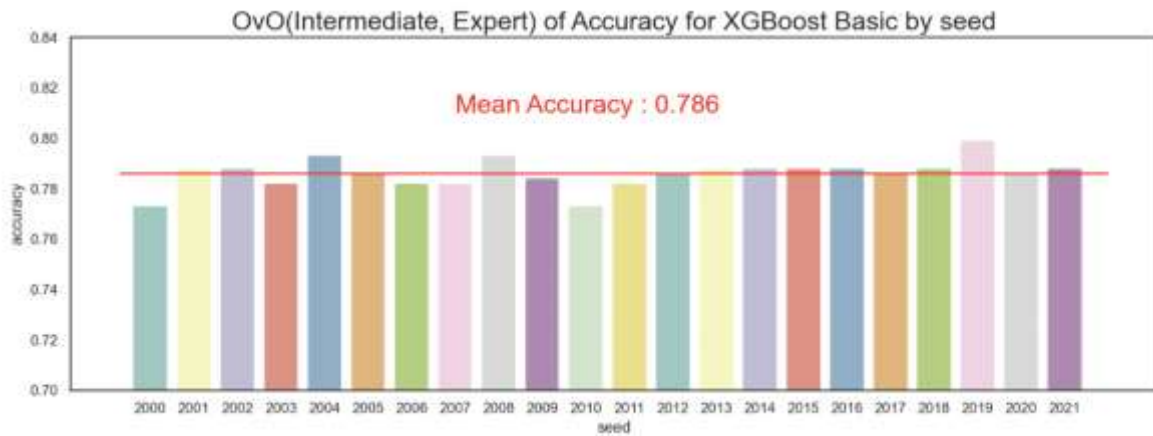


Figure 34 OvO Graph

Seed에 대한 모델의 평균 정확도는 0.786이며 정확도가 가장 높은 seed는 2019이다.

정확도가 가장 높은 seed에 대한 결과는 아래 그림에서 볼 수 있다.

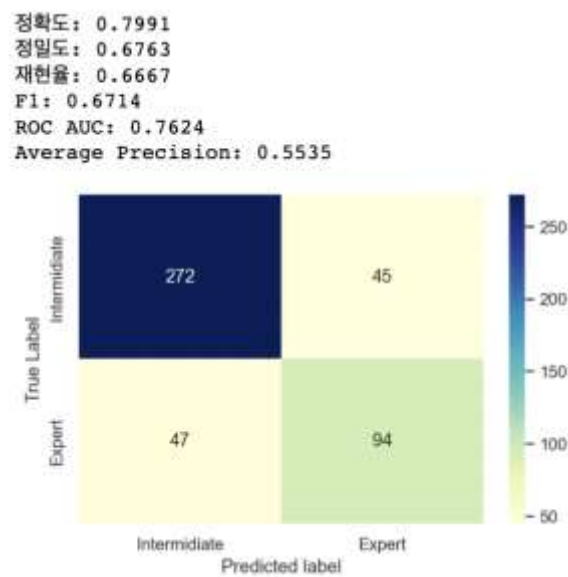


Figure 35 Confusion Matrix and Metric

Feature Importance

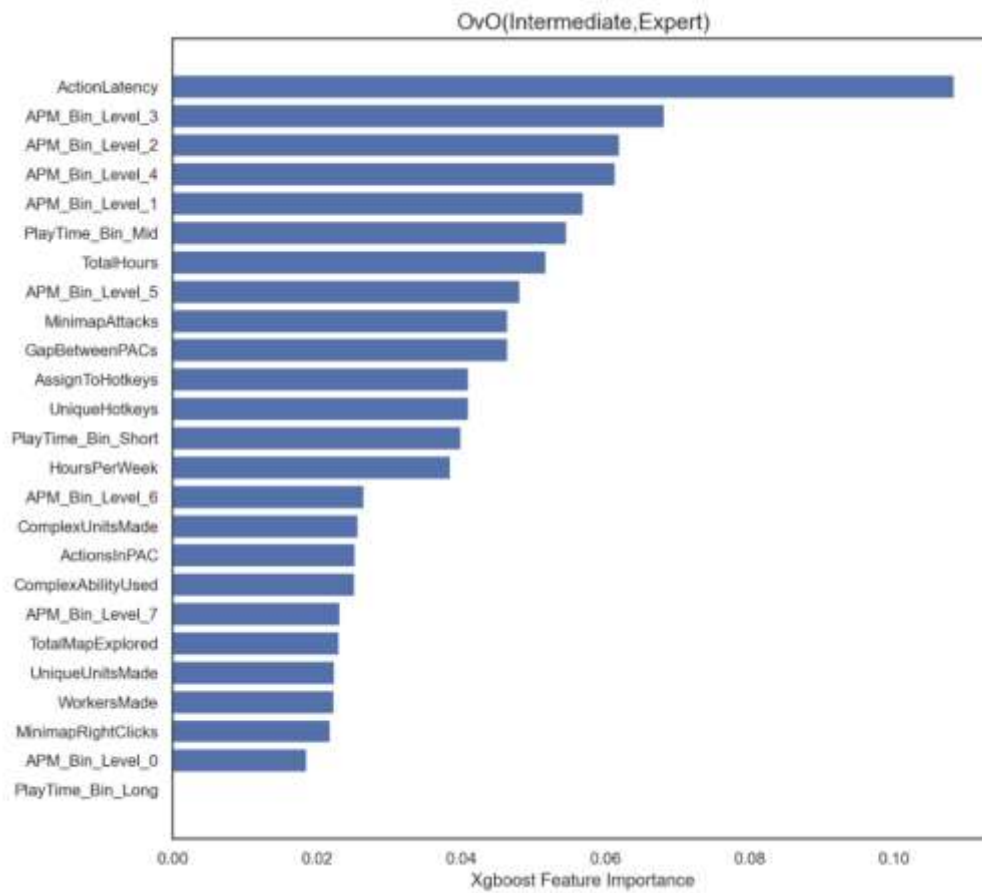


Figure 36 Feature Importance

Feature Importance의 중요도

- 1). Action Latency : 처음 인지부터 행동까지 걸린 시간, 반응 속도
- 2). APM_Level3 : 중위 45% ~ 60% 구간, 컨트롤
- 3). APM_Level2 : 하위 30% ~ 45% 구간, 컨트롤

OvO(One verse One)

전문가(Expert)와 초보자(Novice)

아래는 Seed 2000 ~ 2021에서의 정확도를 나타낸 그림이다.

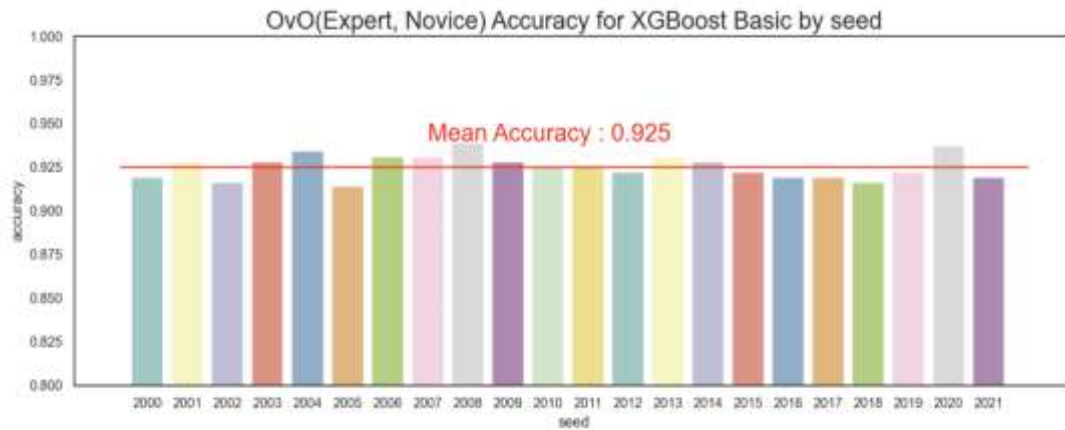


Figure 37 OvO Graph

Seed에 대한 모델의 평균 정확도는 0.925이며 정확도가 가장 높은 seed는 2008이다.

정확도가 가장 높은 seed에 대한 결과는 아래 그림에서 볼 수 있다.

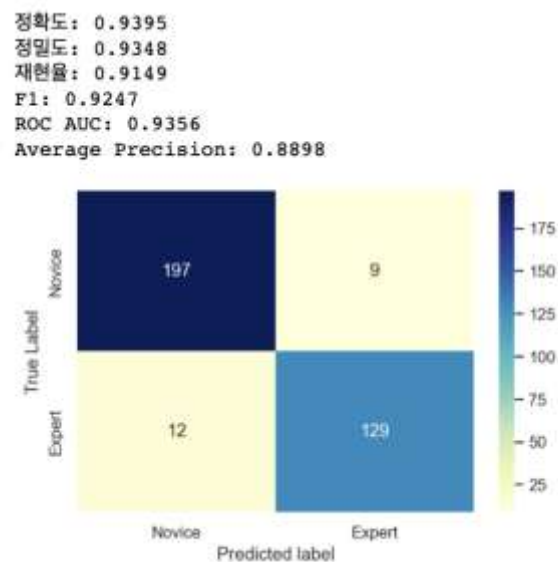


Figure 38 Confusion Matrix and Metric

Feature Importance

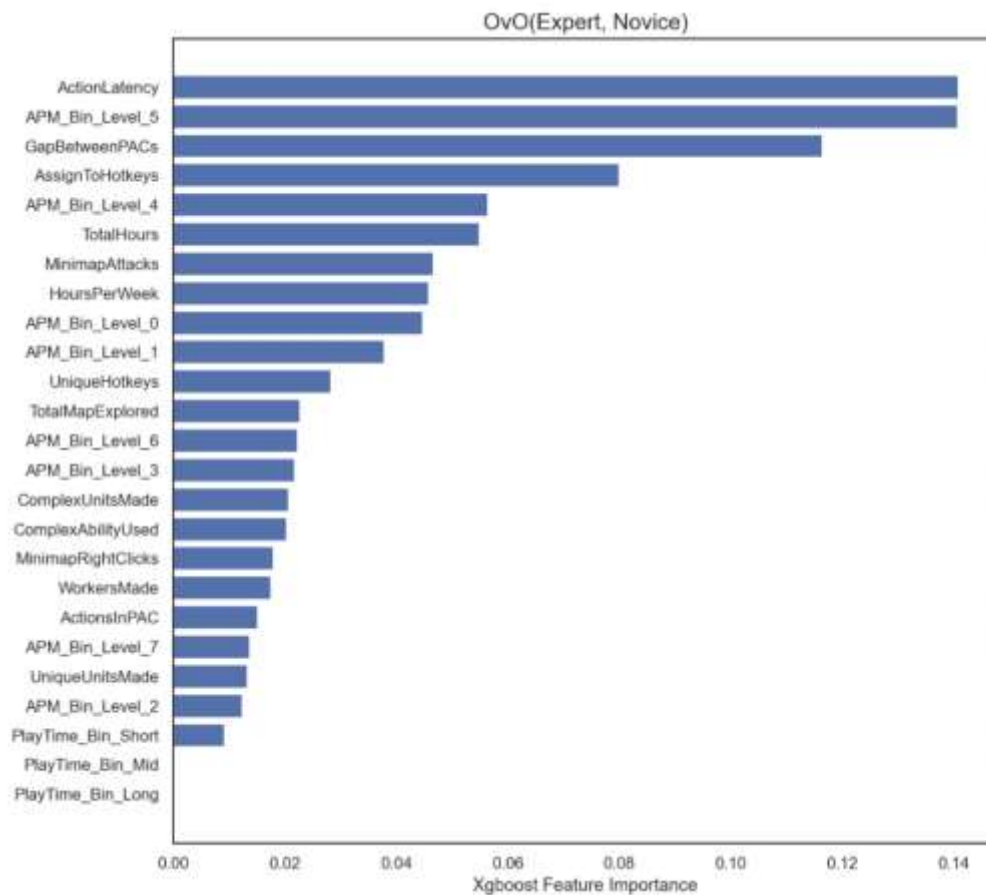


Figure 39 Feature Importance

Feature Importance의 중요도

- 1). Action Latency : 처음 인지부터 행동까지 걸린 시간, 반응 속도
- 2). APM_Level5 : 상위 25% ~ 10% 구간, 컨트롤
- 3). GapBetweenPACs : PAC 시간 간격, 반응 속도

OvR(One verse Rest)

아래는 Seed 2000 ~ 2021에서의 정확도를 나타낸 그림이다.

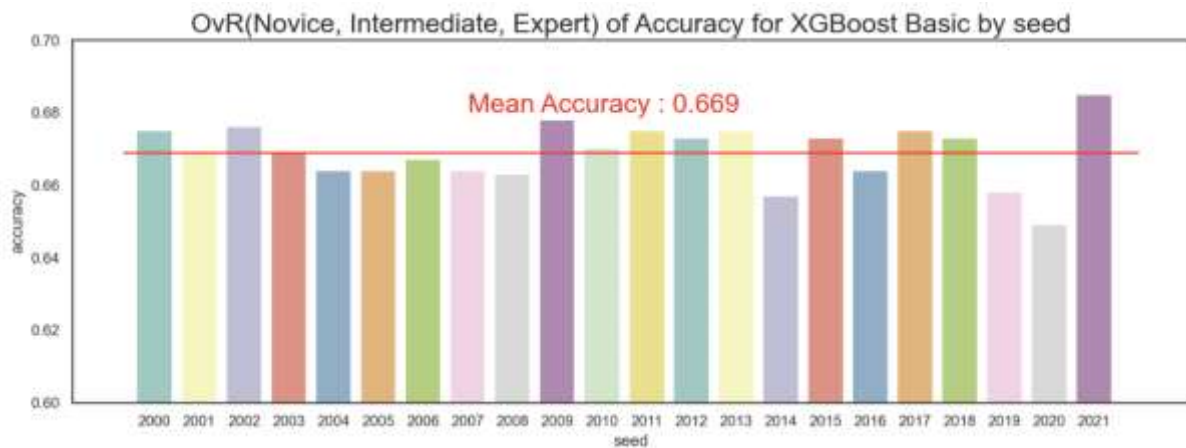


Figure 40 OvR Graph

Seed에 대한 모델의 평균 정확도는 0.669이며 정확도가 가장 높은 seed는 2021이다.

정확도가 가장 높은 seed에 대한 결과는 아래 그림에서 볼 수 있다.

	precision	recall	f1-score	support
0	0.68	0.73	0.70	206
1	0.67	0.64	0.65	317
2	0.66	0.66	0.66	141
accuracy			0.67	664
macro avg	0.67	0.68	0.67	664
weighted avg	0.67	0.67	0.67	664

Figure 41 Classification Report

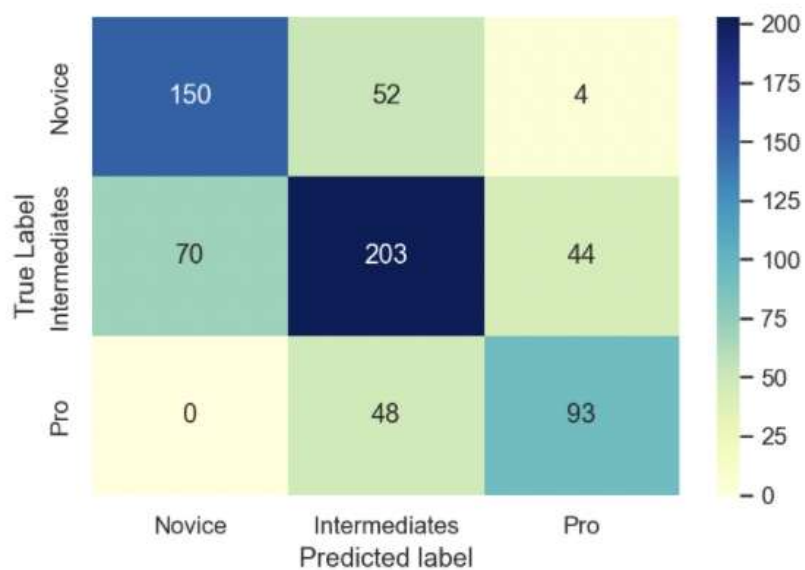


Figure 42 Confusion Matrix

Feature Importance

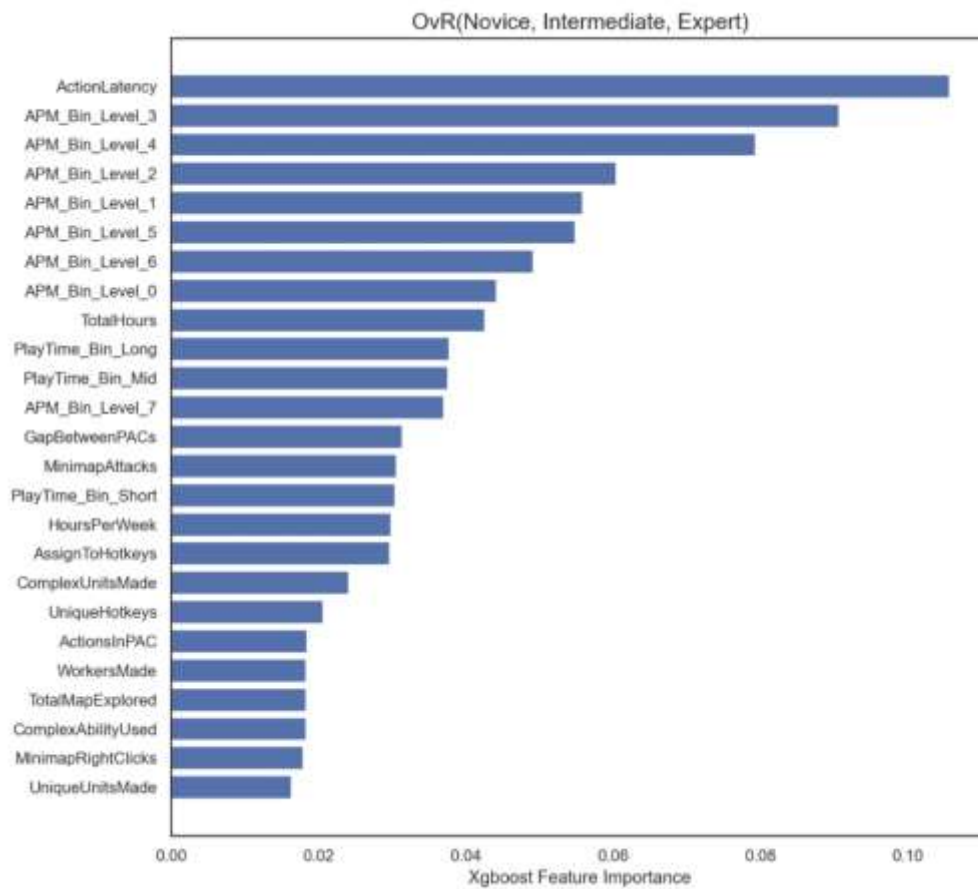


Figure 43 Feature Importance

Feature Importance의 중요도

- 1). Action Latency : 처음 인지부터 행동까지 걸린 시간, 반응 속도
- 2). APM_Level3 : 중위 45% ~ 60% 구간, 컨트롤
- 3). APM_Level4: 중위 60% ~ 75% 구간, 컨트롤

Comparison with results of previous study

OvO(Not Expert verse Expert)

Torin Betting[10]와 동등한 비교를 위해 test size를 0.25로 변경하였고 수준0과 수준1을 동일 그룹으로 지정한다. 그리고 Seed별 모델의 결과에서 이전까지는 정확도의 최대값의 seed를 선정했다면 이번에는 최악의 결과와 비교하기 위해 정확도의 최소값으로 Seed를 선정한다.

아래는 Seed 2000 ~ 2021에서의 정확도를 나타낸 그림이다.

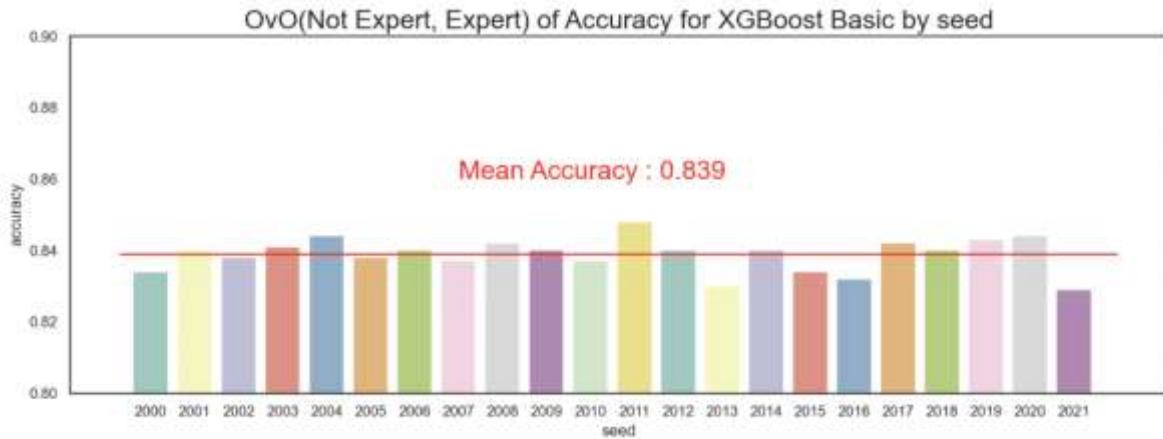


Figure 44 OvO Graph

Seed에 대한 모델의 평균 정확도는 0.839이며 정확도가 가장 낮은 seed는 2021이다. Seed 2021에 대해 threshold를 조정하지 않은 것과 F1 Score가 가장 높은 threshold에 대해 이전 결과와 비교를 할 것이며 오차는 $\text{Err}((\text{this study}-\text{prev study})/\text{prev study}, \%)$ 로 표기한다.

Case1: Basic threshold Table

Metrics	KNN	This Study	Err(%)	Random Forest	This Study	Err(%)	Logistic	This Study	Err(%)
F1	0.5929	0.5966	0.62%	0.5563	0.5966	7.24%	0.5913	0.5966	0.90%
Precision	0.4897	0.60	22.52%	0.7117	0.60	-15.69%	0.4739	0.60	26.61%
Recall	0.7513	0.5932	-21.04%	0.4566	0.5932	29.92%	0.7861	0.5932	-24.54%
Accuracy	0.7684	0.8287	7.85%	0.8504	0.8287	-2.55%	0.7767	0.8287	6.69%
Average Precision	0.4237	0.4428	4.51%	0.4366	0.4428	1.42%	0.4165	0.4428	6.31%
ROC AUC	0.7680	0.7429	-3.27%	0.8523	0.7429	-12.84%	0.8558	0.7429	-13.19%

Case2: Maximize F1 Score threshold Table

Metrics	KNN	This Study	Err(%)	Random Forest	This Study	Err(%)	Logistic	This Study	Err(%)
F1	0.5929	0.6472	9.16%	0.5563	0.6472	16.34%	0.5913	0.6472	9.45%
Precision	0.4897	0.5373	9.72%	0.7117	0.5373	-24.50%	0.4739	0.5373	13.38%
Recall	0.7513	0.8136	8.29%	0.4566	0.8136	78.19%	0.7861	0.8136	3.50%
Accuracy	0.7684	0.8106	5.49%	0.8504	0.8106	-4.68%	0.7767	0.8106	4.36%
Average Precision	0.4237	0.4769	12.56%	0.4366	0.4769	9.23%	0.4165	0.4769	14.50%
ROC AUC	0.7680	0.8117	5.69%	0.8523	0.8117	-4.76%	0.8558	0.8117	-5.15%

불균형이 있는 데이터에서 중요한 지표는 정밀도(Precision)와 재현율(Recall)이다. 정밀도와 재현율은 서로 상충관계에 있기 때문에 F1 Score로 모델을 평가하는 것이 적절하다. 또한 Average Precision은 Precision-Recall 그래프에서 선 아래쪽의 면적에 해당하므로 Average Precision이 높을수록 모델의 성능이 우수하다고 볼 수 있다. 따라서 사전연구와 지표들을 비교해 볼 때 이번 연구에서의 XGBoost 모델은 우수하다. (참고: ROC AUC 및 Average Precision은 test label과 predict label로 정하였다.)

Feature Importance Comparison

사전연구에서 선정한 Feature Importance와 이번 연구에서의 Feature Importance를 비교하여 어떤 변수들이 중요한지 살펴본다.

Feature Importance	Previous Study	This Study
First	APM	ActionLatency
Second	NumberOfPACs	APM_Lelvel3
Third	MnimapAttacks	APM_Lelvel4
Fourth	UniqueHotKeys	APM_Lelvel0

사전연구의 결과와 유사하게 이번 연구에서도 APM이 중급자 이하와 전문가의 차이를 구분하는데 중요한 변수로 선정되었음을 알 수 있다. 대신 ActionLatency의 변수를 추가하여 분석을 하였는데 이 변수는 행동을 인지하고 나서 처음 행동까지 걸리는 시간으로 가장 중요한 변수로 선정되었다. 전문가 일수록 어떤 행동을 하는데 사전에 계획, 예를 들어 빌드 구성 이라던지 유닛 생산 등 스타크래프트를 플레이 함에 있어 능동적인 모습을 볼 수 있다.

Concluding remark

지금까지 OvO와 OvR을 분석하였다. 결론을 내기 위해 표로 정리하여 변수들이 어느 구간에서 중요한지 파악한다.

Feature Importance Rank	OvO			OvR
	Novice Intermediate	Intermediate Expert	Expert Novice	All
First	APM_Level0	Action Latency	Action Latency	ActionLatency
Second	Action Latency	APM_Level3	APM_Level5	APM_Level3
Third	APM_Level1	APM_Level2	GapBetween PACs	APM_Level4
Forth	TotalHours	APM_Level4	AssignToHotkeys	APM_Level2
Fifth	AssignToHotkeys	APM_Level1	APM_Level4	APM_Level1

위 표는 Feature Importance를 토대로 1순위부터 5순위까지 변수들을 나타낸 것이다. 모든 비교 구간에서 공통적으로 APM과 Action Latency가 상위 순위에 있다. 또한 Action Latency의 경우 1순위 또는 2순위, 즉 가장 중요한 변수로 나타났다. APM이 높으면 스타크래프트 티어가 높을 것이다 라는 것 에서는 사전 연구에서도 이미 보인 바 있다. 필자의 견해이지만 Action Latency는 미리 계획되어 있지 않고서는 줄이기가 쉽지 않을 것으로 생각된다. 스타크래프트를 할 때 중요한 요소 중 하나는 적절한 시점에 적절한 자원을 조절하며 상대방의 빌드에 맞게 유닛을 생산하는 것이다.[11] 적절한 이라는 단어는 많은 훈련과 연습이 없이는 어렵다고 본다. 단순히 빠른 명령이 아닌 의미 있는 명령을 내리는 것은 스타크래프트를 하는 누구나 동의하는 부분이라고 생각한다. 또한 무작정 게임을 하는 것이

아닌 의식적인 연습을 통해서 실력을 쌓는 습관을 기르는 것이 바람직하다. 데이터에서 보여 주듯이 TotalHours, 경력이 높다고 무조건 잘하는 것은 아니다. 이 점 꼭 명심하길 바란다.

Reference

- [1] <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0094215#s1>
- [2] <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0075129>
- [3] <http://summit.sfu.ca/item/13328>
- [4] <https://tl.net/forum/starcraft-2/401425-starcraft-2-science-skillcraft-results>
- [5] https://gall.dcinside.com/board/view/?id=starcraft2_new&no=1209967
- [6] https://www.playxp.com/sc2/bbs/view.php?article_id=5397134
- [7] <https://beone.tistory.com/558>
- [8] <https://brunch.co.kr/@snobberys/137>
- [9] <https://www.kaggle.com/code/lifesailor/xgboost/notebook>
- [10] <https://torinrettig.net/StarCraft-Pro-Scout/>
- [11] <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=tmzkdltplqj&logNo=220600182194>

에필로그

개인 프로젝트를 마치며...

누가 알려주지는 않을지라도 나의 첫 화이트 페이퍼 작업(드디어 나도 제 일 저자다!)을 한 것에 대해 개인적으로는 만족하는 거 같다. 프로젝트를 하며 아쉬웠던 점은 종족에 대한 정보가 없는 것이다. 종족마다 APM이 다르기에 종족 변수가 있었으면 더 좋은 분석이 될 수 있다고 생각한다. 또한 단순히 티어가 아닌 MMR과 함께 있었으면 세분화 하여 볼 수 있을 것이다. 마지막으로 EAPM이 측정되지 않은거 같다. 단순한 빠르기를 의미하는 것이 아닌 의미 있는 명령을 집계한다면 더욱 수준 차이를 볼 수 있을 것이라고 확신한다.

거창하게 세상을 통찰하기라고 부제를 지었지만 어쩌면 전문가라고 하는 것이 내가 상상하는 많은 의식적인 연습, 부족한 부분에 대한 피드백, Why와 How를 고루 갖춘 것이 아닐까 물론 분야가 다르기에 모든 분야에서의 전문가는 될 수 없을 지 몰라도 전문가에 대해서 생각할 때 대다수 비슷하게 느낄 것이라고 생각한다.

모델링 분석기법에서 이번에는 비모수적인 방법 XGBoost로 분석하였지만 다음번 에는 모수적인 방법도 차근히 분석하여 모델링 능력도 키워보고 싶다.

읽어주셔서 감사합니다!