

## 기계학습을 활용한 온라인게임 매치메이킹 개선방안

김용우, 김영민  
한양대학교 기술경영학과  
{ywkim, yngmnkim}@hanyang.ac.kr

Improvement of online game matchmaking using machine learning

Yongwoo Kim, Young-Min Kim  
Department of Technology Management, Hanyang University

### 요약

온라인 게임에서 다른 플레이어와의 상호작용은 플레이어의 만족도에 영향을 미친다. 따라서, 비슷한 수준의 플레이어를 매치시켜 원활한 상호작용을 도모하는 것은 플레이어의 게임 경험을 위해 중요하다. 그러나, 게임의 최종승패로만 플레이어의 평가점수를 증감시키는 현재의 평가 방식으로는 신규 및 복귀 플레이어의 원활한 매치가 불가능하다. 본 연구에서는 스타크래프트II의 리플레이를 활용하여 매치메이킹 개선을 위한 기계학습 활용방안을 제시한다. 매치메이킹의 기준이 되는 플레이어의 MMR 점수를 예측하는 기계학습 모델을 생성하고 성능을 평가하였다. 모델의 오차는 리그 평균 MMR 점수 범위의 40.4% 수준으로, 제안된 방식을 통해서 플레이어를 실력과 근접한 리그에 즉시 배치할 수 있음을 확인하였다. 또한, 결과에 대한 플레이어의 수용도를 높일 수 있도록 예측의 근거를 도출하는 방안도 제시되었다.

### ABSTRACT

In online games, interactions with other players may threaten player satisfaction. Therefore, matching players of similar skill levels is important for players' experience. However, with the current evaluation method which is only based on the final result of the game, newbies and returning players are difficult to be matched properly. In this study, we propose a method to improve matchmaking quality. We build machine learning models to predict the MMR of players and derive the basis of the prediction. The error of the best model was 40.4% of the average MMR range, confirming that the proposed method can immediately place players in a league close to their current skill level. In addition, the basis of predictions may help players to accept the result.

**Keywords** : Machine Learning(기계학습), StarCraft(스타크래프트), Matchmaking(매치메이킹)

Received: Nov. 10, 2021    Revised: Dec. 13, 2021  
Accepted: Dec. 23, 2021  
Corresponding Author: Young-Min Kim (Hanyang University)  
E-mail: yngmnkim@hanyang.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. 서 론

광범위한 사람들에게 매력적인 게임을 만드는 것은 게임 흥행을 위해 중요하다[1]. 특히 플레이어 중심의 게임 디자인은 매력적인 게임을 만들기 위한 중요한 요소로 여겨진다[2]. 플레이어 중심의 게임 디자인을 위해 개별 플레이어가 게임을 원활하게 플레이할 수 있도록 게임의 환경(퀘스트, 난이도 등)을 동적으로 디자인하는 방법이 주로 연구되었다. 그러나 최근에는 대부분 게임이 멀티플레이 환경에서 이루어져 게임 내에서 다른 플레이어와 상호작용하는 것이 일상적인 일이 되었으며, 플레이어는 다른 플레이어와 함께 플레이하면서 상호 간 경쟁이나 협력이 일어나고 이로부터 새로운 형태의 재미를 경험한다[3]. 이러한 재미 요소를 극대화하기 위해서는 플레이어들을 적절히 매칭시키는 것이 중요하다. 즉, 플레이어를 어떻게 매칭하느냐에 따라서 플레이어의 게임 경험에 영향을 줄 수 있으므로, 플레이어의 정확한 실력평가에 기반한 신속하고 정확한 매치메이킹<sup>1)</sup>은 중요하다고 할 수 있다. 서로 실력이 비슷한 사람이나 팀끼리 대결하고 승패를 가리는 과정에서 게임을 플레이하는 플레이어들은 몰입을 경험하게 되고 해당 게임을 계속 이용하게 되는 유인으로 작용한다. 만약 실력 차이가 크거나 상대와 매칭되는 경우 쉽게 좌절하고 해당 게임에서 이탈할 가능성이 높아진다. 이처럼 게임의 재미는 물론 이용 의도에도 매칭의 질이 영향을 미치기 때문에 매치메이킹은 게임에서 중요한 요소로 자리 잡았으며, 효과적인 매치메이킹 시스템 구축을 돕는 매치메이킹 프레임워크[4]도 개발되었다. 일부 대형 게임사에서는 실력평가 및 매치메이킹을 위한 시스템을 직접 개발하는 전담팀을 두기도 한다.

플레이어 간 매칭을 위해서는 MMR(Match Making Rating)<sup>2)</sup> 값이 유사한 플레이어들을 찾아서 매칭시키는 방식이 일반적이다. MMR 산출을 위해서는 보통 Elo 레이팅(Elo Rating)을 기반으로 하되, 게임의 특성을 고려하여 변형을 거친 시스템

을 사용하고 있다. 많은 스포츠와 게임에서 오랫동안 사용되어 온 방식으로 그 범용성과 신뢰도가 입증되었고 플레이어들도 작동방식을 충분히 이해하고 있다. 그러나, 새롭게 진입하는 신규 플레이어 혹은 신규 계정을 만든 기존 플레이어가 자신의 실력에 맞는 점수를 획득하기 위해서는 여러 번의 플레이가 필요하며, 이 과정에서 플레이어 간 매칭의 불균형이 일어나게 된다는 점은 이 시스템의 단점이다.

본 연구에서는 스타크래프트II(StarCraft II)의 게임 데이터를 이용하여 플레이어의 MMR(Match Making Rating)을 예측한다. 초심자부터 프로수준까지 모든 실력 대의 플레이어를 대상으로 하여 MMR을 예측하고 예측 정확도를 알아본다. 구축된 예측 모형을 통해서 게임을 플레이한 경험이 거의 없는 플레이어의 실력을 적절히 평가하여 매치메이킹을 보완 및 개선할 수 있는 방법을 제시하고자 한다.

제2장에서는 플레이어의 실력평가를 위해서 이용되는 방법들을 알아본다. 제3장에서는 관련된 연구를 설명하고 본 연구가 가지는 차별점을 제시한다. 제4장에서는 분석에 사용된 데이터를 설명하고, MMR 예측을 위한 모형을 설계한다. 제5장에서는 분석결과를 제시하고, 제6장에서 연구내용의 결과와 시사점을 정리한다.

## 2. 실력평가 시스템

### 2.1 엘로(Elo)

엘로 평점 시스템(Elo Rating System)[5]은 일대일 경쟁 게임에서 사용되는 실력측정 척도이다. 각 경기에서 플레이어가 승리할 확률을 0에서 1사이의 값으로 표현하며, 실제 플레이 결과에 따라서 플레이어의 점수를 변화시킨다.

먼저 플레이어 A가 승리할 확률은 수식(1)과 같다. 여기서 R은 플레이어의 점수이며, N은 어떤 플

1) 온라인게임에서 각 플레이어를 서로 연결시키는 과정.  
2) 매치메이킹을 위해 각 플레이어에게 할당된 점수.

레이어가 다른 플레이어보다 10배 더 높은 기대승률을 가질 때 두 플레이어의 점수 차이를 의미한다. N을 임의로 증가하거나 감소시킴으로써 점수의 스케일을 조정할 수 있다.

$$W_A = 1 / (1 + 10^{R_B - R_A / N}) \quad (1)$$

경기결과에 따라서 플레이어의 점수를 갱신하는 방법은 수식(2)와 같다. K는 한 경기(게임)의 중요도를 나타내는 상수로서, K가 높으면 한 경기가 큰 점수 변화를 가져오게 된다. 이 값은 스포츠의 진행 방식에 따라서 다른 값으로 설정한다. 실제 경기결과를 나타내는  $S_A$ 는 승리 시 1, 무승부 시 0.5, 패배 시 0을 가지게 된다.  $S_A$ 와 기대승률  $W_A$ 의 차이를 K와 곱하여 기존 점수에 더하면 새로운 점수가 된다.

$$R'_A = R_A + K(S_A - W_A) \quad (2)$$

보통 N은 400, 신규 플레이어의 점수는 1,500으로 설정하는 것이 일반적이나 이 수치는 임의로 변경할 수 있다.

## 2.2 글리코(Glicko)

Glicko-1[6]은 엘로 평점 시스템에 편차(Rating Deviation) 개념을 추가하여 실력의 불확정성에 대해 고려하였다. 편차 개념의 도입으로 신규 플레이어의 경우 점수 변동 폭이 크고, 많은 게임을 치를수록 편차가 작아지면서 점수 변동 폭이 작아진다. 신규 플레이어가 본인이 가진 실력에 맞는 점수에 더욱 빨리 도달하게 하는 것이 목적이다.

이후 Glicko-2[7]에서는 비활동 변수를 추가하여 게임을 하지 않는 휴식 기간이 길어지면 점차 편차가 증가하도록 설계되었다. 따라서, 플레이어가 휴식 기간을 가진 후 복귀하면 점수의 변동 폭이 커진다. 휴식 기간이 길어질수록 플레이어의 실력이 변화할 수 있다는 점이 고려되었다.

## 2.3 트루스킬(TrueSkill)

트루스킬(Microsoft TrueSkill)[8] 역시 엘로 평점 시스템에 기반한 것으로 마이크로소프트에서 개

발하였다. 글리코와 마찬가지로 플레이어의 실력의 불확정성을 고려하도록 설계되었으며, 레이팅 갱신 알고리즘 역시 유사하게 작동한다. 글리코의 경우 로지스틱 분포를 가정하지만 트루스킬의 경우 가우시안 분포를 가정하고 있다. 그리고 글리코는 휴식 기간과 실력의 불확정성이 선형 관계이지만, 트루스킬에서는 최근 2게임 사이의 휴식 기간을 고려하여 일정량만 증가한다. 가장 큰 차이점은 글리코의 경우 일대일 경기 유형을 염두에 두고 개발되었으나 트루스킬의 경우 다대다 경기 유형에서 사용하기 위해 설계되었다는 점이다.

## 2.4 적용사례 및 문제점

엘로 평점 시스템은 직관적이고 이해하기 쉬워서 많은 호응을 얻었지만 문제점도 존재한다. 신규 플레이어가 실력에 맞는 점수를 획득하기까지 오랜 시간이 걸리고, 휴식 기간으로 인한 실력의 감소를 반영할 수 없다. 또한, 높은 점수의 플레이어가 실력이 불확실한 신예들과의 경기를 기피하기 쉬운 구조이다. 이러한 문제점 때문에 초기의 엘로 평점 시스템을 변형 없이 사용하는 경우는 드물다.

이후 엘로 평점 시스템 기반의 수정된 방법들이 등장하였고 축구, 체스, 바둑 등 기존 스포츠는 물론, 리그오브레전드, 스타크래프트II, DOTA2 등 세계적으로 인기 있는 게임들도 엘로 평점 기반의 방법들을 특성에 맞게 변형하여 플레이어의 순위를 매기고 매치메이킹에 활용하고 있다. 이처럼 엘로 평점에 기반한 실력평가 방법은 다양한 스포츠에서 오랜 기간 운용되어 그 범용성과 신뢰성이 매우 높다고 할 수 있다.

방법들 간 차이는 있으나 모두 엘로 평점 시스템을 기본으로 개량한 방법들이므로 경기결과에 따라서 점수가 조정된다는 것은 공통적이다. 따라서, 경기결과가 거의 없는 신규 플레이어의 실력평가를 할 때 Cold Start Problem<sup>3)</sup>이 문제가 된다. 물론 엘로 평점 시스템에서는 신규 플레이어의 K값을

3) 시스템이 아직 충분한 정보를 수집하지 않은 사용자에게 대한 추론을 도출할 수 없는 문제.

조정하여 이 문제를 완화할 수 있으며, Glicko와 트루스킬에서는 편차 개념을 도입함으로써 이 문제를 완화했다. 그러나 이러한 조정에도 불구하고 한계점이 있다. 첫째로, 여전히 게임의 내용은 철저하게 무시되고 게임의 결과(승패)로만 점수의 조정이 이루어진다는 것이다. 둘째로, 진입 초기에는 점수의 변동 폭이 가장 크기 때문에 최초로 치르는 게임에서 실수하거나 운이 나빠서 패배하는 경우 실력에 맞는 위치로 다시 이동하기까지 많은 시간이 소요된다는 것이다. 만약 게임 내용을 분석하여 단 한 경기만으로도 플레이어가 어느 정도의 수준인지 알 수 있다면, 플레이어 배치에 필요한 시간이 줄어드는 것은 물론이고 배치 정확도의 향상도 기대할 수 있을 것이다.

### 3. 관련 연구

게임 리플레이를 이용하여 게임 종료 후에 게임을 다시 확인할 수도 있고, 리플레이로부터 정보를 추출하는 것도 가능하다. 스타크래프트II의 게임 리플레이를 이용하여 다양한 형태의 연구가 진행되었다. 리플레이에서 추출한 정보로 게임의 데이터셋을 구성하는 연구들이 존재하며[9,10,11], 이를 이용한 승패예측[12,13]이나 전략예측[14,15,16]도 활발하게 이루어졌다. 최근에는 인간 대신 게임을 치르는 인공지능 에이전트의 개발도 이루어져 상당한 진전을 이루었다[17].

플레이어의 실력예측 관한 연구들 역시 계속 진행되어왔다. 게임 내 변수들을 이용한 플레이어의 실력예측 자체에 집중하여 예측 정확도를 향상시킬 수 있는 새로운 예측변수를 발굴하거나, 예측을 위해서 어떤 변수가 더 중요한지를 밝히는 연구가 주로 이루어졌다. [Table 1]<sup>4)</sup>과같이 MMR의 변동에 따라서 각 지역 서버 내에서 리그변동이 이루어지는데, 플레이어가 가장 많은 EU 및 NA 서버에서 각 리그의 평균 MMR Range는 698이다.

연구자들은 게임 내에서 생성된 변수들을 이용하여 플레이어가 속한 7개의 리그를 예측하는 방

식으로 연구를 진행하였다. Thompson JJ et al.(2013)은 3,360개의 리플레이를 활용하여 리그에 따라서 게임 내 플레이어의 행동에 차이가 있다는 것을 발견했다[18]. 플레이어를 각 리그로 분류하려고 시도하였으나, 인접 리그로 오분류하는 경우가 많았기 때문에, 2단계 차이가 나는 리그에 속한 플레이어를 분류하는 다수의 분류기(two-league classifier)를 구축하였다. Avontuur et al.(2013)은 1,273개의 리플레이를 활용하여 플레이어들을 7개의 리그로 분류하였다[19]. 분류정확도는 47.3%로 다소 낮았으나, 오분류의 67%는 인접 리그로 오분류한 경우였다. 이찬민 외(2021)는 기존 연구보다 많은 46,398개의 리플레이를 활용하였으며, 분류정확도를 75.3%까지 개선하였다[20]. 이전 연구에서 게임 내 행동변수를 이용한 실력평가의 가능성이 확인되었고, 공통적으로 운동신경과 공간시각에 관련된 변수들이 예측에 매우 중요하다는 사실이 발견되었다. 나아가 RTS 게임이 체스처럼 인간의 지각과 인지과정을 측정하고 탐구하기 위한 표준적

[Table 1] StarCraft II League MMR Ranges

| League       | Server | EU        | NA        | Users   |
|--------------|--------|-----------|-----------|---------|
| Grand Master |        | 5190-     | 5380-     | TOP 200 |
| Master       | 1      | 4939-5189 | 5119-5379 | 4%      |
|              | 2      | 4690-4939 | 4860-5119 |         |
|              | 3      | 4440-4690 | 4600-4860 |         |
| Diamond      | 1      | 4053-4440 | 4173-4600 | 23%     |
|              | 2      | 3667-4053 | 3747-4173 |         |
|              | 3      | 3280-3667 | 3320-3747 |         |
| Platinum     | 1      | 3133-3280 | 3173-3320 | 23%     |
|              | 2      | 2987-3133 | 3027-3173 |         |
|              | 3      | 2840-2987 | 2880-3027 |         |
| Gold         | 1      | 2693-2840 | 2733-2880 | 23%     |
|              | 2      | 2547-2693 | 2587-2733 |         |
|              | 3      | 2400-2547 | 2440-2587 |         |
| Silver       | 1      | 2200-2400 | 2267-2440 | 23%     |
|              | 2      | 2000-2200 | 2093-2267 |         |
|              | 3      | 1800-2000 | 1920-2093 |         |
| Bronze       | 1      | 1550-1800 | 1660-1920 | 4%      |
|              | 2      | 1301-1550 | 1401-1660 |         |
|              | 3      | 1051-1301 | 1141-1401 |         |

4) 2021년 7월 19일(2021 Season2) 기준 League MMR Ranges.

인 작업 환경을 제공할 수 있다는 점에서 주목받고 있다.

공통적으로 이전 연구들은 종속변수를 해당 플레이어가 속한 리그로 설정하고 분류문제로 접근하였다. 리그의 경계 부분에 위치한 플레이어들의 오분류가 생기기 쉽다는 점은 이러한 설정의 문제점이다. 또한, 플레이어를 각 리그로 분류하는 정확도는 개선되었으나 매치메이킹의 척도가 되는 MMR과의 직접적인 비교가 불가능하여 실제 시스템에 적용할 수 있는지 확인하기 어려웠다.

본 연구에서는 리플레이에서 추출한 다양한 예측변수들로 MMR을 예측하는 회귀문제를 정의한다. 그리고 이러한 예측 방식이 신규 플레이어의 리그 배치에 활용될 수 있는지 그 가능성을 확인하고자 한다.

## 4. 데이터 및 모형설계

### 4.1 데이터 세트 구축

본 연구에서는 국내외에서 가장 활발하게 연구되고 있는 게임 중 하나인 스타크래프트II의 게임 리플레이로부터 추출된 데이터를 이용한다. 게임의 리플레이를 수집하기 위해 gggreplays.com<sup>5)</sup>, spawningtool.com<sup>6)</sup>, drop.sc<sup>7)</sup>로부터 플레이어들이 자발적으로 업로드한 리플레이를 수집하였다.

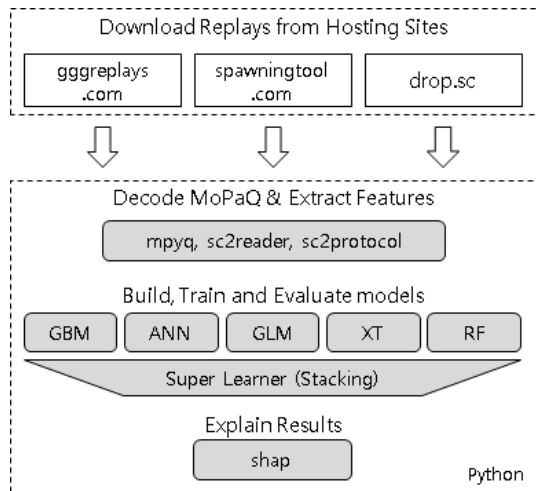
스타크래프트II의 리플레이는 Blizzard에서 개발한 압축 포맷인 MoPaQ(MPQ) 형태로 저장되어 있다. Python 환경에서 sc2reader, s2protocol, mpyq 패키지를 이용하여 MPQ파일을 해독하고 플레이어별 행동을 요약한 지표들을 생성하였다. 즉, 플레이어를 기준으로 한 개의 데이터 인스턴스가 생성되었다. 일대일 형태의 게임 내에 2명의 플레이어가 존재하므로, 각 게임으로부터 2개의 데이터 인스턴스가 생성된다. 리플레이로부터 추출된 예측변수는 플레이어 정보, 플레이어의 공간사각 및 운동신경, 게임 내 경제, 게임 내 전략에 관련된 총 43개이며 변수에 관한 설명은 [Table 2]와 같다.

분석의 일관성을 위하여 다음과 같은 경우 분석 대상에서 제외되었으며, 조건을 충족하는 199,729개의 인스턴스를 이용하여 분석을 진행하였다.

- 총 플레이타임 3분 미만
- 일대일 게임이 아님
- 래더게임<sup>8)</sup>이 아님
- 게임 진행속도가 'Faster'가 아님
- 빌드번호가 77379 미만(최신 대규모 패치 이전)
- 플레이어에 할당된 MMR이 없음
- 게임이 플레이된 서버가 NA나 EU가 아님

### 4.2 모형설계

Artificial Neural Networks(ANN), Extra Trees(ET), Gradient Boosting Models(GBM), Generalized Linear Models(GLM), 그리고 Random Forest(RF)를 이용하여 MMR을 예측하는 5개의 모델을 구축한다. 학습 및 테스트 데이터는 8:2 비율로 분할하고, 모델 학습 시 모델의 초매개변수(hyperparameter) 조정은 5겹 교차검증(5-fold cross validation)을 통해 진행한다. 이후



[Fig. 1] Overview of the analysis process

5) <https://gggreplays.com/matches>

6) <https://lotv.spawningtool.com/replays/>

7) <https://drop.sc/>

8) 공식적인 순위 결정이 이루어지는(MMR 변동이 있는) 게임.

[Table 2] List of Features Used

| Type     | Feature                             | Description   |
|----------|-------------------------------------|---|
| General  | bnet_region                         | Server region where the game was played   |
| General  | play_race                           | Player play race(Terran, Protoss, or Zerg)  |
| General  | oppo_play_race                      | Opponent play race (Terran, Protoss, or Zerg)                                       |
| General  | is_winner                           | Player is winner or not   |
| General  | pick_race                           | Player selected Random or not   |
| General  | oppo_pick_race                      | Player selected Random or not   |
| General  | play_length                         | Game play duration  |
| General  | map_hash                            | The map that a game was played on   |
| General  | base_build                          | The version that a game was played  |
| Motor    | avg_apm                             | Average actions-per-minute  |
| Motor    | permin_CameraEvent                  | Camera events per minute generated when the player camera moves, zooms, or rotates. |
| Motor    | permin_GetControlGroupEvent         | The number of retrieving control group per minute                                   |
| Motor    | permin_SetControlGroupEvent         | The number of setting control group per minute                                      |
| Motor    | permin_AddToControlGroupEvent       | The number of setting adding units to the group, per minute                         |
| Motor    | permin_TargetPointComsmandEvent     | The number of command that targets a location and not a unit per minute             |
| Motor    | permin_TargetUnitCommandEvent       | The number of command that targets a unit per minute                                |
| Motor    | distinct_GetControlGroup            | The number of different hotkeys retrieved   |
| Motor    | distinct_SetControlGroup            | The number of hotkeys set   |
| Resource | permin_minerals_used_current_{Vary} | Average mineral value of the current {Economy / Army / Technology} in a game        |
| Resource | permin_vespene_used_current_{Vary}  | Average vespene gas value of the current {Economy / Army / Technology} in a game    |
| Resource | max_minerals_lost_{Vary}            | The total mineral value of all {Vary : Economy / Army / Technology} lost            |
| Resource | max_vespene_lost_{Vary}             | The total vespene gas value of all {Vary : Economy / Army / Technology} lost        |
| Resource | max_minerals_killed_{Vary}          | The total mineral value of enemy {Vary : Economy / Army / Technology} killed        |
| Resource | max_vespene_killed_{Vary}           | The total vespene gas value of enemy {Vary : Economy / Army / Technology} killed    |
| Resource | mean_minerals_collection_rate       | Average of players' mineral collection rate in a game                               |
| Resource | mean_vespene_collection_rate        | Average of players' vespene collection rate in a game                               |
| Resource | mean_workers_active_count           | The average number of active workers the player has in a game                       |
| Strategy | mean_food_ratio                     | Average supply consumption rate in a game   |
| Strategy | mean_mineral_store_rate             | Average of (current unused mineral / mineral collection rate) in a game             |
| Strategy | mean_vespene_store_rate             | Average of (current unused vespene / vespene collection rate) in a game             |
| Strategy | permin_food_block                   | The minute the supply is blocked per minute   |

생성된 여러 모델을 결합(stacking)하는 super learner를 생성한다. 개별 모델의 초매개변수 조정 중 생성된 모든 모델이 결합 대상이 된다. Stacking[21,22]은 개별 학습 모델의 최적 조합을 찾기 위한 super learner를 학습하여 모델의 성능을 증대시키는 방식이다. Stacking을 통해서 성능을 증대시킬 수 있다는 사실이 밝혀지면서[23] 최근에는 성능향상을 위한 일반적인 방법 중 하나로 자리잡았다.

최종적으로 생성된 개별 모델과 결합된 모델을 평가하고 플레이어 배치에 사용될 수 있는 수준인지 검토한다. 또한, 학습된 모델이 예측한 MMR에 큰 영향을 미친 변수들과 그 영향력을 시각화하여 보여준다. 데이터 수집부터 모델 구축 및 평가 그리고 결과의 해석 과정은 [Fig. 1]과 같다.

## 5. 분석결과

### 5.1 성능평가

개별 모델 404개와 결합된 모델 13개를 포함하여 총 417개의 모델이 생성되었다. 모델종류별 성능평가 결과는 [Table 3]에 나타나 있다. 먼저 개별 모델의 오차를 살펴보면 MAPE (Mean Absolute Percentage Error) 9.22%-11.38%, MMR 수치와 직접적인 비교가 가능한 MAE (Mean Absolute Error)는 302.46-374.45로 각 리그의 MMR 평균 범위인 698보다 작은 수치이다.

개별 모델 중에서는 GLM의 성능이 가장 좋지 못했고 GBM의 성능이 가장 우수했다. 이러한 성능 차이는 플레이어의 행위가 게임 내에서 매우 복잡하게 일어나며 그것을 선형적으로 모델링 하기 어렵다는 것을 의미한다.

개별 모델 중에서 종류별로 가장 우수한 5개를 결합한 super learner(Best 5)의 경우 MAPE 8.92%, MAE 292.58로 개별 모델의 성능보다 우수했다. 리그의 평균 MMR 범위인 698의 41.9% 수준의 낮은 MAE를 기록하였다. 생성된 모든 개별 모델을 결합 대상으로 삼은 super learner(All)의

경우 리그 평균 MMR 범위의 40.4% 수준인 MAE 282.26로 가장 우수한 성능을 보였다. 생성된 모델의 성능지표를 종합해볼 때 제안된 방식을 통한 신규 플레이어 배치가 충분히 가능하다고 볼 수 있다.

[Table 3] Error Terms of the Best Models

| Error<br>Type             | MAE    | RMSE   | MAPE     |
|---------------------------|--------|--------|----------|
| ANN                       | 306.42 | 410.23 | 9.47(%)  |
| GLM                       | 374.45 | 487.42 | 11.38(%) |
| GBM                       | 302.46 | 400.27 | 9.22(%)  |
| RF                        | 320.55 | 424.82 | 9.75(%)  |
| ET                        | 328.51 | 431.49 | 9.98(%)  |
| Super Learner<br>(Best 5) | 292.58 | 388.46 | 8.92(%)  |
| Super Learner<br>(All)    | 282.26 | 378.26 | 8.65(%)  |

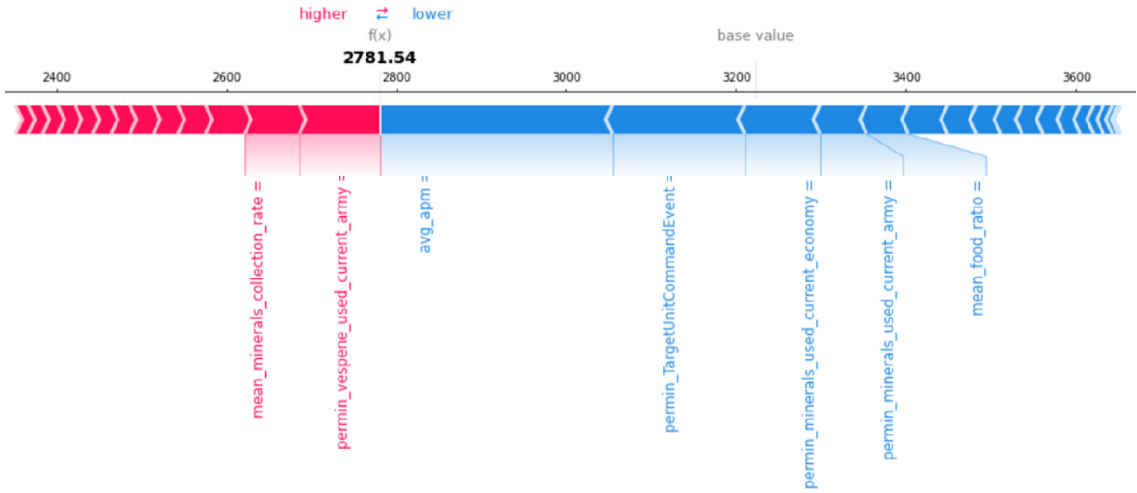
[Table 4]는 생성된 417개의 모델 중 성능평가 상위 10개의 모델을 나타낸다. 상위 10위까지 모두 결합된 모델이 차지했으며 단일 모델로서 가장 우수했던 GBM은 성능상으로 11위에 머물렀다. 결합된 모델의 성능은 단일 모델보다 우수하지만, 많은 모델이 결합되어 있는 만큼 그 내부는 단일 모델보다 복잡하여 모델에 대한 직관적인 설명이 어려워지는 단점이 있다.

[Table 4] TOP10 Models

| Error<br>Type | MAE    | RMSE   |
|---------------|--------|--------|
| Super Learner | 282.26 | 378.26 |
| Super Learner | 288.68 | 384.93 |
| Super Learner | 292.58 | 388.46 |
| Super Learner | 294.01 | 390.32 |
| Super Learner | 294.01 | 390.32 |
| Super Learner | 294.83 | 390.78 |
| Super Learner | 296.06 | 392.19 |
| Super Learner | 296.43 | 392.63 |
| Super Learner | 301.57 | 399.70 |
| Super Learner | 301.65 | 399.82 |

### 5.2 결과해석

복잡한 black box models를 사용할 때의 약점은 예측된 결과에 대한 설명이 부족하다는 점이다. 모



[Fig. 2] Visualization Example

델을 결합해서 성능을 향상시키는 경우 이러한 문제는 심화된다. 플레이어가 기계학습을 통한 배치 시스템을 받아들이기 위해서는 결과에 대한 설명 역시 필요하다. 인공지능이 예측한 결과를 사람이 이해할 수 있는 형태로 제시하는 XAI(eXplainable Artificial Intelligence) 분야에서는 이러한 부분을 해결하기 위해서 끊임없이 노력해 왔다. 대표적으로 LIME[24]과 SHAP[25]이 많이 이용되고 있는데, 여러 분야에서 다양한 모델에 적용되고 있다[26,27].

본 연구에서는 kernel SHAP을 이용하여 개별 게임에서의 플레이어 지표가 MMR에 어떻게 영향을 미치는지 알아보고 시각화하였다. 가중치를 가진 선형회귀 모델을 이용하여 Shapley value[28]를 추정하는 방식으로 복잡도가 높아 계산시간은 오래 걸리지만, 모델의 종류에 상관없이 사용할 수 있는 특성(model-agnostic) 때문에 복잡한 모델에도 활용이 가능하다.

[Fig. 2]는 가장 우수한 모델이 예측한 특정 플레이어의 MMR 수치(2,782)와 예측의 근거가 제시되어 있다. 파란색으로 표기된 부분은 MMR에 부정적으로 작용한 변수들이고, 빨간색으로 표현된 부분은 MMR에 긍정적으로 작용한 변수들이다. 이 플레이어에게는 단위시간 당 더 많은 명령을 내릴 수 있도록 조작능력을 계발하는 것과 잉여 미네랄<sup>9)</sup>

을 적시에 소비하여 건물을 짓거나 유닛을 생산하는 것이 권장된다. 예측값에 긍정적인 영향을 준 변수와 부정적인 영향을 준 변수들을 제시하고 그 영향력 순위를 제시해줌으로써 예측된 MMR에 대해 플레이어를 이해시키는 물론, 부족한 부분을 스스로 점검할 수 있게 도와줄 수 있다.

## 6. 결론 및 논의

게임의 최종승패로만 플레이어의 MMR을 증감시키는 방식은 여러 가지 문제점을 안고 있다. 본 연구에서는 스타크래프트II 데이터를 이용하여 신규 플레이어의 매치메이킹 불균형 개선을 위한 기계학습 활용방안을 제안하였다. 구축된 모델들은 모두 각 리그의 평균 MMR 범위보다 훨씬 낮은 MAE를 기록하였고, 이러한 접근방식이 신규 플레이어 배치에 유용하게 사용될 수 있음을 발견했다.

게임을 서비스하는 회사는 게임 내 다양한 데이터를 기록하도록 로그 설계를 직접 할 수 있고 훨씬 더 많은 데이터를 사용할 수 있어서 더욱 우수한 모델을 구축할 수 있을 것으로 보인다. 매치메이킹이 일어나는 다른 장르의 게임(격투 게임 등)에서도 해당 게임의 핵심적인 행동들을 기록할 수 있

9) 스타크래프트II에서 유닛 생성에 필요한 자원의 한 종류.



는 로그만 설계되어 있다면 제안된 방식을 활용하여 매치메이킹 개선이 가능하다.

이러한 접근법은 초기 진입 플레이어나 복귀 플레이어의 배치에 효과적일 수 있지만, 플레이어를 배치하는 메커니즘이 투명하지 않다는 점에서 플레이어들이 이러한 방식의 배치를 쉽게 납득하지 못할 가능성이 있다. 특히 스스로 생각하는 MMR 수준보다 낮게 예측된 경우 플레이어는 불만을 가지기 쉽다. 따라서 배치결과의 이유를 플레이어에게 설명하는 것은 중요하다. 이는 플레이어가 배치 결과를 쉽게 수용할 수 있도록 도울 수 있음과 동시에 표준적인 실력 향상을 위한 방향성도 제공해줄 수 있다. 또한, 지각된 공정성을 증대시키는 것은 플레이어의 부정적인 감정반응을 감소시킬 수 있고 [29], 재이용 의도에도 영향을 미칠 수 있어[30] 결과에 대한 설명은 꼭 필요하다고 할 수 있다.

제안된 접근방식은 다대다 게임에서 더욱 효과를 발휘할 수 있을 것으로 보인다. 다대다 게임에서는 본인의 행동뿐만 아니라 팀원들의 행동이 게임의 최종승패에 더 큰 영향을 준다. 한 팀에 속하는 플레이어의 수가 많아질수록, 다대다 게임에서 본인이 플레이하는 역할이 보조적일수록 이러한 경향은 더욱 심해질 수밖에 없다. 이러한 이유로 본인의 플레이 수준과 심한 괴리가 있는 리그에 오랫동안 머무르는 경향이 일대일 게임보다 다대다 게임에서 더욱 심하게 나타난다. 이러한 문제점을 감소시키기 위해 본 연구에서 제안한 방식을 다대다 게임에 적용할 수 있을 것이며, 향후 이를 검증하기 위한 추가적인 연구가 필요하다.

## REFERENCES

- [1] Bateman, C., & Boon, R., "21st Century Game Design (Game Development Series)", Charles River Media, Inc., 2005.
- [2] Charles, D., McNeill, M., McAlister, M., Black, M., Moore, A., Stringer, K., ... & Kerr, A., "Player-centred game design: Player modelling and adaptive digital games", Proceedings of DiGRA 2005 Conference, pp. 285-298, 2005.
- [3] Lazzaro, N., "Why we play games: Four keys to more emotion in player experiences", Proceedings of GDC, Vol. 306, pp. 1-8, 2004.
- [4] Google LLC., "Open Match", accessed Dec 14, 2021, <https://development.open-match.dev/>.
- [5] Elo, Arpad E., "The rating of chessplayers, past and present", Arco Pub., 1978.
- [6] Glickman, Mark E., "Parameter estimation in large dynamic paired comparison experiments", Applied Statistics, Vol. 48, No. 3, pp. 377-394, 1999.
- [7] Glickman, Mark E., "Dynamic paired comparison models with stochastic variances", Journal of Applied Statistics, Vol. 28, No. 6, pp. 673-689, 2001.
- [8] Herbrich, R., Minka, T. & Graepel, T., "TrueSkill™: a Bayesian skill rating system", Proceedings of the 19th International Conference on Neural Information Processing Systems, pp. 569-576, 2006.
- [9] Robertson, G., & Watson, I., "An improved dataset and extraction process for starcraft ai." The Twenty-Seventh International Flairs Conference, 2014.
- [10] Synnaeve, G., & Bessiere, P., "A dataset for StarCraft AI and an example of armies clustering", Eighth Artificial Intelligence and Interactive Digital Entertainment Conference, 2012.
- [11] Lin, Z., Gehring, J., Khalidov, V., and Synnaeve, G., "Stardata: A starcraft ai research dataset", Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference, 2017.
- [12] Ravari, Y. N., Bakkes, S., & Spronck, P., "Starcraft winner prediction", Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Vol. 12. No. 1, 2016.
- [13] Stanescu, M., Hernandez, S. P., Erickson, G., Greiner, R., & Buro, M., "Predicting army combat outcomes in StarCraft", Ninth artificial intelligence and interactive digital entertainment conference, 2013.
- [14] Hsieh, J. L., & Sun, C. T., "Building a player strategy model by analyzing replays of real-time strategy games", 2008 IEEE

- International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 3106-3111, 2008.
- [15] Weber, B. G., & Mateas, M., "A data mining approach to strategy prediction", 2009 IEEE Symposium on Computational Intelligence and Games, pp. 140-147, 2009.
- [16] Schadd, F., Bakkes, S., & Spronck, P., "Opponent Modeling in Real-Time Strategy Games", Proceedings of the GAME-ON, pp. 61-70, 2007.
- [17] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... & Silver, D., "Grandmaster level in StarCraft II using multi-agent reinforcement learning", Nature, Vol. 575, No. 7782, pp. 350-354, 2019.
- [18] Thompson JJ, Blair MR, Chen L & Henry AJ, "Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning", PLoS ONE 8(9): e75129, 2013.
- [19] Avontuur, T., Spronck, P., & Van Zaanen, M., "Player skill modeling in Starcraft II", Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Vol. 9, No. 1, 2013.
- [20] Lee, C. M., & Ahn, C. W., "Feature Extraction for StarCraft II League Prediction." Electronics, Vol. 10, No. 8, 2021.
- [21] Wolpert, D. H., "Stacked generalization", Neural networks, Vol. 5, No. 2, pp. 241-259, 1992.
- [22] Breiman, L., "Stacked regressions", Machine learning, Vol. 24, No. 1, pp. 49-64, 1996.
- [23] van der Laan, M., Polley, E. & Hubbard, A., "Super Learner", Statistical Applications in Genetics and Molecular Biology, Vol. 6, No. 1, 2007.
- [24] Ribeiro, M. T., Singh, S., & Guestrin, C., "'Why should i trust you?' Explaining the predictions of any classifier", Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135-1144, 2016.
- [25] Lundberg, S. M., & Lee, S. I., "A unified approach to interpreting model predictions", Proceedings of the 31st international conference on neural information processing systems, pp. 4768-4777, 2017.
- [26] Sundararajan, M., & Najmi, A., "The many Shapley values for model explanation", International Conference on Machine Learning, pp. 9269-9278, 2020.
- [27] García, M. V., & Aznarte, J. L., "Shapley additive explanations for NO2 forecasting", Ecological Informatics, Vol. 56, No. 101039, 2020.
- [28] Roth, A. E. (Ed.), "The Shapley value: essays in honor of Lloyd S Shapley", Cambridge University Press, 1988.
- [29] Hegtvedt, K. A., & Killian, C., "Fairness and Emotions: Reactions to the Process and Outcomes of Negotiations", Social Forces, Vol. 78, No. 1, pp. 269 - 302, 1999.
- [30] Maute, M. F., & Dubés, L., "Patterns of emotional responses and behavioural consequences of dissatisfaction", Applied Psychology, Vol. 48, No. 3, pp. 349-366, 1999.



김용우 (Kim, Yongwoo)

약력 : 2018 Universität Lüneburg, Data Science 석사  
2021-현재 한양대학교 기술경영학과 박사과정

관심분야 : 기계학습, 데이터마이닝, Esports



김영민 (Kim, Young-Min)

약력 : 2010 Paris-VI, 컴퓨터공학 박사  
2020-현재 한양대학교 기술경영학과 교수

관심분야 : 기계학습, 데이터마이닝