

# 빅데이터 분석 및 시각화 개론 보고서

17조

201511046 김정훈

201511062 김호준

201511068 남정현

## 프로젝트 주제

저희 프로젝트의 최초주제는 League of Legends(LOL) 프로게이머 선수들의 머니볼 입니다.

## 주제 선정 이유

LOL은 자신 캐릭터와 팀원들의 캐릭터를 여러 방법으로 강화시켜 최종적으로 상대방의 중앙 구조물을 부수어 이기는 게임입니다. LOL은 경기의 승패를 예측하기가 매우 어려운 게임입니다. LOL 경기에는 플레이어 개인의 실력 외에도 사용하는 게임 캐릭터의 조합, 플레이어가 구매한 아이템, 추가적인 효과를 주는 필드 몬스터 등 게임안에서 여러가지의 전략이 가능하고 전략의 수가 굉장히 많기 때문에 일반적으로는 승패를 예측하기 어렵습니다. 그렇기 때문에 저희는 빅데이터를 분석함으로써 어떻게 해야 플레이어가 이길 확률을 높일 수 있는지 알아보고 싶었습니다. 또 각 플레이어마다 실력이 다른데 개인 전적 데이터를 이용해서 야구처럼 각 개인의 능력치를 수치화 할 수 있을 것이라 생각했습니다. 이 분석이 성공하면 프로게이머 경기에도 이를 적용해서 어떤 선수를 조합했을 때 최고의 성능을 내는 팀을 만들어 낼 수 있는지 또 각 프로게이머들이 어떠한 전략을 취해야 가장 승률이 높을 지를 예측할 수 있다고 생각했습니다.

## 처음 주제의 어려운 점

저희가 얻은 데이터는 모두 op.gg에서 가져온 데이터입니다. 하지만 op.gg의 데이터에는 플레이어가 게임 중에 선택한 전략에 대해 알 수 있는 요소가 매우 적었습니다. 그래서 게임 중 사용한 전략을 얻을 수가 없었습니다. 또 op.gg 사이트가 응답속도가 느리고 전적 데이터를 얻기 위해서 클릭해야하는 버튼이 많아 코드를 실행시키는데 많은 시간이 걸렸습니다.

## 주제 변경 & 이유

그래서 저희는 주제를 좀 더 수정해서 League of Legends(LOL) 프로게이머 선수들의 경기 승패 예측으로 변경하였습니다. 이전에 설명한 것처럼 게임 중 플레이어가 취한 전략에 대해서는 알

수가 없어서 게임이 끝나고 나서 나오는 게임 결과 데이터를 이용할 수 밖에 없었습니다. 게임 결과 데이터로는 프로게이머가 어떤 전략을 취해야 하는지는 알 수가 없어서 머니볼까지는 힘들다고 생각했고 다만 프로게이머 개인의 기량을 대략적으로 파악해서 팀의 총 실력을 파악하고 경기에서 승패를 예측해볼 수 있을 것이라 판단했습니다. 또 op.gg에서 프로게이머의 모든 개인 전적 데이터를 얻기 어려워서 최근 20경기만 활용해서 프로게이머의 기량을 판단한 수 있을 것이라 생각했습니다.

## 사용한 데이터

저희가 사용한 데이터는

1. 약 10000 경기의 LOL 상위 랭크 플레이어들의 개인 전적
2. 승패를 예측을 테스트 해보고자 하는 경기 26건
3. 각 경기에 참여한 플레이어들의 최근 개인 전적 20건입니다.

첫번째 데이터는 LOL 상위 랭크 플레이어들의 약 10000 경기에 달하는 개인전적 데이터입니다. 각 경기에는 10명이 참여하고 (Blue team : 5명, Red team : 5명) 10명의 플레이어들의 경기 전적이 적혀 있으므로 사실상 100000건의 개인전적이 적혀 있는 데이터입니다. 이 데이터의 row는 플레이어 한명한명을 나타내고 각 column은 {MatchNumber, GameResult, Champion, Nickname, Kill, Death, Assist, Damage, Buy\_Pward, Build\_ward, Destroy\_ward, CS, CSperMin, Baron, Dragon, Tower, KDA}로 총 99730 rows \* 18 columns 짜리 데이터입니다. 게임 전적에서 어떤 factor가 영향력이 큰지 알아보고 위한 데이터이고 동시에 머신러닝을 학습하기 위한 Train 데이터입니다.

각 column에 대해 상세한 설명을 드리자면 MatchNumber는 각 경기를 구분하기 위한 경기번호입니다. GameResult는 해당 경기가 이겼으면 1, 졌으면 0입니다. Champion은 각 플레이어가 경기에서 사용한 게임 캐릭터의 종류입니다. 각 캐릭터마다 서로 고유한 플레이방식을 가지고 고유한 플레이 전략을 요구합니다. Nickname은 플레이어가 게임에서 사용하는 이름입니다. Kill은 해당 경기에서 플레이어가 다른 플레이어를 죽인 횟수입니다. Death는 해당 경기에서 플레이어가 죽은 횟수입니다. Assist는 해당 경기에서 같은 팀의 플레이어가 다른 팀의 플레이어를 죽이는데 도움을 준 횟수입니다. Damage는 플레이어가 준 피해의 총량입니다. Buy\_Pward는 플레이어가 사용한 \*핑크 와드의 개수입니다. Build\_ward는 플레이어가 설치한 \*와드의 총 개수입니다. Destroy\_ward는 플레이어가 없앤 상대팀 와드의 총 개수입니다. CS는 플레이어가 잡은 \*미니언이나 \*몬스터의 개수를 의미합니다. CSperMin은 플레이어의 분당 CS를 의미합니다. Baron은 해당 경기에서 같은 팀이 잡은 \*바론의 개수입니다. Dragon은 해당경기에서 같은 팀이 잡은 \*드래곤의 개수입니다. Tower는 해당 게임에서 플레이어가 부순 \*타워의 개수입니다. KDA는 (Kill+Death)/Assist로 플레이어의 경기성적을 나타낼 때 LOL에서 주로 씁니다.

\* 와드 : LOL에서 플레이어는 자기 자신 주변과 팀원 주변 시야만 볼 수 있습니다. 와드를 설치하면 와드 주변의 시야 또한 볼 수 있습니다.

\* 핑크 와드 : LOL에서는 은신이라는 특수한 능력이 있는 Champion이 있습니다. 해당 Champion은 핑크 와드라는 은신 Champion을 탐지할 수 있는 와드 주변에 있을 때 탐지됩니다.

\* 미니언 : LOL에서 미니언은 맵에서 지속적으로 생성되는 아주 약한 유닛입니다. 미니언을 잡으면 골드와 경험치를 주며 골드를 이용해 아이템을 사서 자신의 캐릭터를 강화할 수 있습니다.

\* 몬스터 : LOL에서 몬스터는 맵에서 지속적으로 생기나 미니언보다 강하고 미니언 보다 드물게 생성되며 정글이라는 특수한 지역에만 생성됩니다. 미니언보다 더 많은 골드와 경험치를 줍니다.

\* 바론 : LOL에서 제일 강한 몬스터입니다. 잡으면 많은 골드와 경험치를 주며 팀 전체에게 이득을 주는 좋은 효과가 일정 시간 지속됩니다.

\* 드래곤 : LOL에서 두번째로 강한 몬스터입니다. 잡으면 많은 골드와 경험치를 주며 팀 전체에게 이득을 주는 좋은 효과가 일정 시간 지속됩니다. 잡으면 잡을수록 팀 전체에 주는 효과가 강화됩니다.

\* Tower : LOL에서는 상대방의 중앙 구조물을 파괴하면 팀이 이기며 상대팀이 항복하지 않는 이상 유일한 승리 방법입니다. 중앙 구조물로 가기 위해서는 각 길목마다 있는 Tower를 모두 부숴야만 접근할 수 있습니다. Tower는 많은 체력을 가지고 있고 Champion을 공격할 수 있습니다.

두번째 데이터는 승패 예측을 테스트 해보고자 하는 경기 \* 건입니다. 첫번째 데이터를 이용하여 학습한 머신러닝을 적용한 데이터이고 머신러닝이 제대로 학습되었는지 확인하기 위한 Test데이터입니다.

세번째 데이터도 승패 예측을 테스트 할 때 이용한 데이터입니다. 경기에 참여한 각 플레이어의 실력을 알기 위해 추출한 데이터입니다.

## 분석 방법

분석은 총 4가지 단계로 진행했습니다.

1. 약 10000건의 경기데이터에서 승패에 영향력이 큰 factor들을 찾기
2. 영향력이 큰 factor들만 골라 10000건의 경기데이터 머신러닝 (Linear Regression)
3. 실제로 진행된 임의의 경기를 학습된 머신러닝을 통해 승패 결과 예측 후 결과 확인
4. 3번과 비슷한 방식으로 프로 팀의 승패 결과 예측 후 결과 확인

먼저 첫번째 분석은 10000건의 경기데이터를 이용해 승패에 영향력이 큰 factor를 찾아보았습니다. 데이터의 각 경기에서 Blue team과 Red team의 각 factor의 평균값을 구한 뒤 평균값이 높은 팀이 이긴 비율(more win), 평균값이 낮은 팀이 이긴 비율(more lose), 평균값이 두 팀이 같은 비율(same)을 구했습니다. 이를 통해서 각 factor 중 어떤 factor가 승패에 영향력이 큰 지 알 수 있었습니다.

두번째 분석은 첫번째 분석을 통해 골라진 영향력이 큰 factor들을 통해 10000건의 경기 데이터를 학습시켰습니다. 각 경기마다 승패 여부를 통해 factor마다 weight와 bias를 구했습니다. 구한 weight와 bias는 후에 세번째 분석에서 승패결과를 예측할 때 이용되었습니다. 머신 러닝 방법은 Linear Regression을 이용하였습니다.

세번째 분석은 두번째 분석에서 학습된 머신러닝을 바탕으로 승패 결과를 예측한 것입니다. 전적사이트에서 임의의 경기를 골라 경기에 참여한 플레이어들을 확인하였습니다. 경기의 참여한 플레이어들의 최근 20 경기의 전적을 크롤링한 뒤 평균을 내어 플레이어마다 최근 기량을 파악했습니다. (두번째 데이터에서 사용한 factor와 같은 factor를 플레이어마다 가지도록 함.) 이를 학습된 머신 러닝에 넣어 프로그램이 factor마다 구한 weight와 bias를 이용해 경기의 승패 여부를 예측하도록 했습니다.

네번째 분석 또한 세번째 분석과 같은 방식으로 분석을 진행했지만 다른 점이 있다면 네번째 데이터는 실제 프로게이머들의 최근 공식 경기의 승패를 예측하였습니다. 프로게이머의 최근 기량 파악도 세번째 분석과 마찬가지로 프로게이머의 최근 20경기 전적을 이용하였습니다.

## 분석 결과

첫번째 분석의 결과는 다음과 같습니다.

	more win	more lose	same
KDA2	95.91%	4.09%	0%
KDA	95.04%	4.96%	0%
Kill	94.89%	3.79%	1.32%
Assist	93.03%	6.32%	0.65%
Damage	87.61%	12.39%	0%
CS	81.11%	18.29%	0.59%
Tower	96.86%	1.43%	1.70%
Dragon	62.26%	12.38%	21.35%
Baron	54.38%	7.11%	38.50%
Des_ward	59.36%	36.10%	4.53%
Build_ward	56.89%	40.56%	2.54%
Pink_ward	54.20%	39.90%	5.91%
1 * Dragon + 3.5 * Baron	77.02%	12.33%	10.64%
1.85 * Des_ward + 1 * Build_ward + 0.12 * Pink_ward	72.72%	27.28%	0%

승패에 영향이 큰 factor들로 (a) {Kill, Death, Assist,  $(1.85 * \text{Des\_ward} + 1 * \text{Build\_ward} + 0.12 * \text{Pink\_ward})$ } 를 선정하였습니다. 두번째로 (b) {KDA2, Damage,  $(1.85 * \text{Des\_ward} + 1 * \text{Build\_ward} + 0.12 * \text{Pink\_ward})$ }를 선정하였습니다.

두번째 분석은 각 factor들을 Linear regression으로 학습시켰고 weight와 bias는 알 수 없습니다.

세번째 분석을 통해 승패를 예측하였습니다. 총 26건의 경기 예측 결과는 다음과 같습니다.

(a) :

예측 성공	예측 실패	예측 불가	성공 비율
14	7	5	53.85%

(b) :

예측 성공	예측 실패	예측 불가	성공 비율
14	6	6	53.85%

예측 성공 비율이 높진 않았지만 유의미한 예측률을 가진다는 것으로 볼 수 있습니다. 또 같은 방식으로 프로게이머 팀들의 경기 결과를 예측했을 때 결과는 다음과 같았습니다.

	팀1	팀2
SG vs SKT	1	0
SG vs KT	1	0
SKT vs KT	0	1
LG vs KT	1	0
LG vs SG	0	1
LG vs SKT	1	0

실제 경기 결과와 예측결과가 맞는 지는 확인하지 못하였습니다.

## 토의 및 향후 발전 방향

위 분석결과를 통해서 예측 결과가 아주 좋진 않지만 프로게이머 팀의 경기 결과를 예측하는데 도움이 될 것이라고 생각합니다. 추후에 하고 싶은 것들은 다음과 같습니다. 첫번째 데이터의 모든 factor를 경기 결과 예측에 사용하고 싶었는데 그러지 못해 추후에는 모든 factor를 머신 러닝에 학습시켜 경기 결과 예측을 해보고 싶습니다. 그리고 factor가 높은 쪽의 경기 승패 비율을 통해 경기에서 중요한 factor를 뽑아냈는데 좀 더 다양한 방법으로 뽑아내면 좋을 것이라 생각합니다. ANOVA나 머신러닝의 tree를 이용해서 추후에 이를 해보고 싶습니다. 또한 이 프로젝트에서는 경기별로 팀 당 factor 평균값을 구하여 예측에 사용하였는데 플레이어 별로 예측에 사용하면 좀 더 정확한 결과가 나올 것이라고 예상합니다.