브론즈에게 희망을

LoL api를 이용한 유저 데이터 분석

201714168 유호원
201714170 조해성
201714167 양현영
201714169 조영래

Requirements

< user requirements>

a. 기능 요구사항

1. 챔피언 선택

1.1 사용자 챔피언 선택

- 사용자가 챔피언을 선택하면 해당 챔피언의 승률을 보여준다.
- 승률이 높은 아이템 트리를 보여준다.
- 승률이 높은 룬을 보여준다.

1.2 상대 챔피언 선택

- 상대 챔피언을 선택하면 사용자가 선택한 챔피언과의 상대 승률을 보여준다.
- 상대 챔피언에 맞는 아이템 트리를 보여준다.
- 상대 챔피언에 맞는 룬을 보여준다.

1.3 랭크 선택

- 해당 챔피언으로 최근에 랭크를 승급 한 사람들의 아이템 트리를 보여준다.
- 룬을 보여준다.

1.4 카운터 챔피언 확인

- 해당 챔피언을 선택할 때 밴 해야 하는 챔피언이 무엇인지 보여준다.

2. 사용자 개인 분석

2.1 전적 검색

- 사용자가 자신의 소환사명을 입력하면 전적을 검색할 수 있다.

2.2 프로필 검색

- 사용자의 프로필 아이콘, 현재 속해 있는 리그를 보여준다.

Requirements

< user requirements>

a. 기능 요구사항

2. 사용자 개인 분석

2.3 챔피언 추천

- 사용자가 속한 리그, 리그 포인트, 승률 --?

2.4 아이템 트리 추천

- 사용자 개인의 챔피언 성향을 분석해 잘 어울리는 챔피언을 추천해준다.

2.5 행동강령 추천

- 사용자의 현재 매치를 분석해서 사용자의 행동강령을 추천해준다.

3. 오브젝트

- 각 랭크 별로 오브젝트(바론, 드래곤 등)를 어떻게 가져갔는지 보여주고 그에 따른 승률을 보여준다.

b. 비기능 요구사항

4. Performance

- 사용자가 전적을 검색하면 5초 내에 최신정보를 가져와야 한다.

5. Development Environment

- Riot api를 기반으로 데이터베이스를 구축해야 한다.

6. Goals

- 사용자의 티어를 향상시킨다.

Requirements

< system requirements>

c. 기능 요구사항

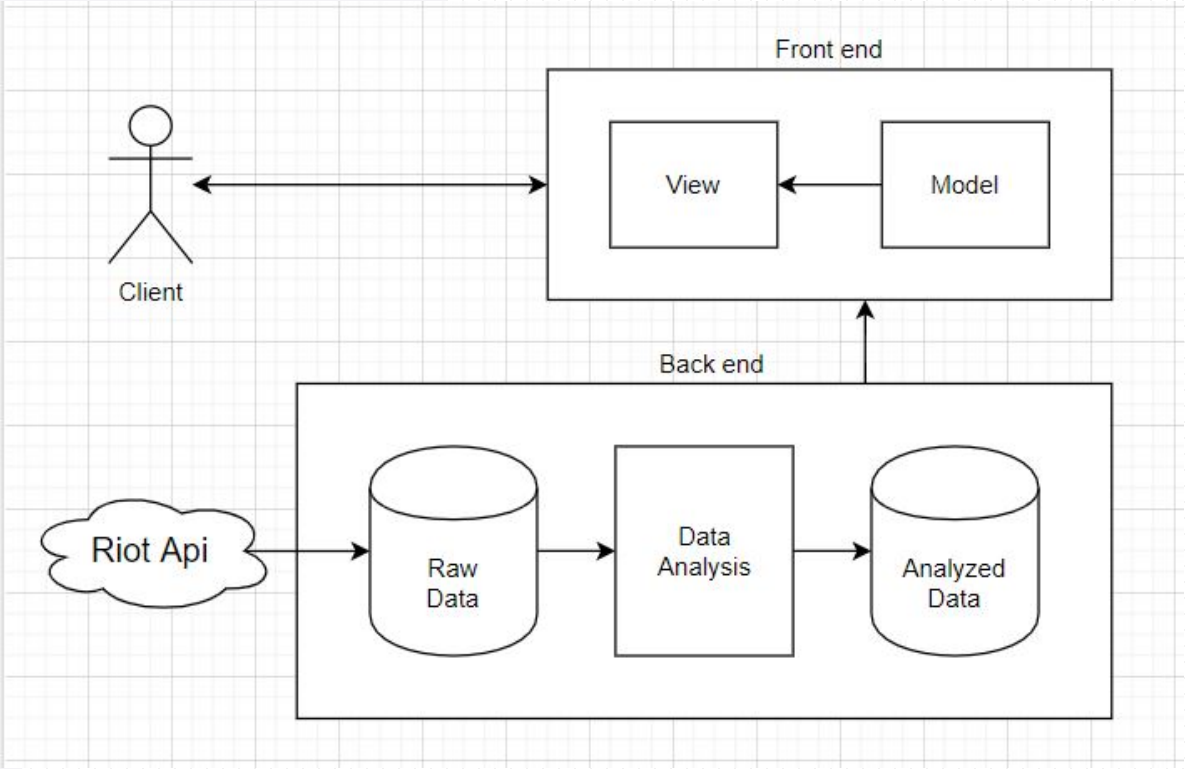
1. Raw 데이터 수집
   - Riot api로부터 필요한 데이터셋만 골라서 수집한다.
   - 잘못된 데이터를 제거하기 위해 데이터 전처리 작업을 수행한다.
   - 누락없이 데이터베이스에 저장시킨다.
2. 데이터 분석
   - Raw 데이터를 조합해 EDA를 수행한 후 머신러닝 또는 딥러닝을 수행한다.
   - 사용자가 보기 쉬운 데이터로 시각화 시킨다.
3. 데이터 저장
   - 분석한 데이터를 분석 데이터베이스에 저장시킨다.
4. 성능 측정 및 분석
   - 머신러닝 또는 딥러닝으로 학습 시킨 모델의 성능을 측정한다.

b. 비기능 요구사항

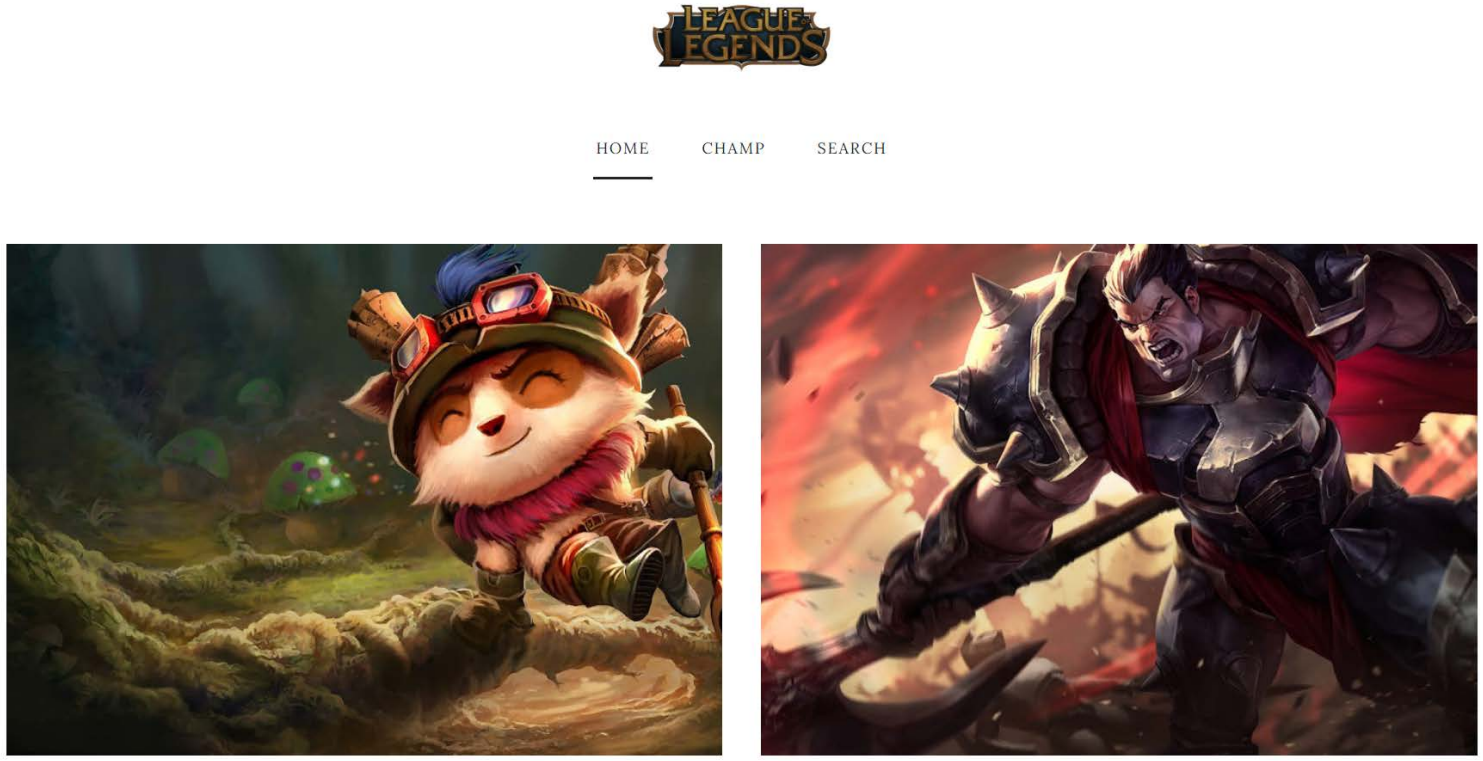
1. Api 제약
   - Riot api request는 1초에 20번, 1분에 100번으로 제한되며, 2020년 말까지 사용 가능하다.
2. 하드디스크 사양
   - 한국 서버 유저 330만 명, 각 유저 별 수백 개의 매치 데이터, 각 매치 별 수백개의 이벤트 데이터가 존재한다. 이 데이터들을 감당 가능한 하드디스크가 필요하다.
3. GPU 사양
   - 딥 러닝을 위해 CUDA를 사용한다. CUDA는 Geforce 8 시리즈급 이상에서 동작한다.

Architecture Diagram

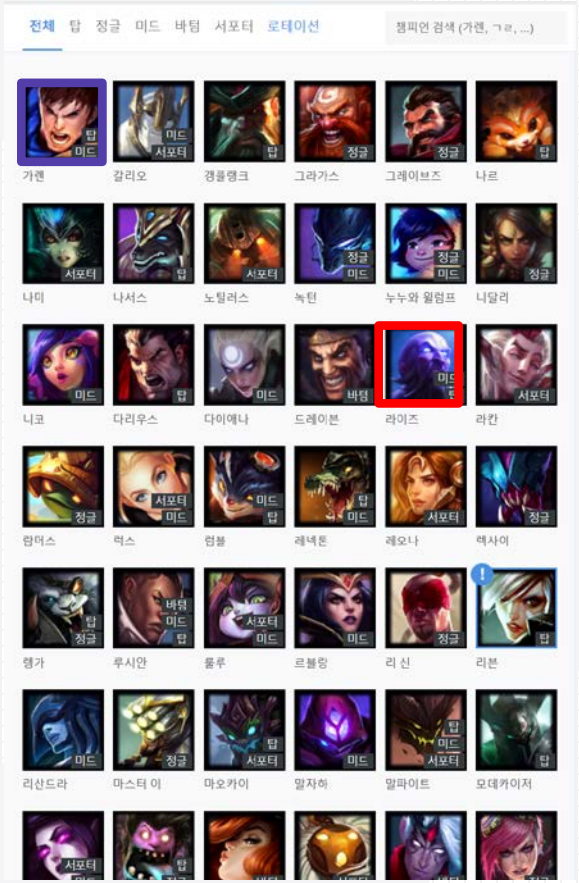
- Back-end 서버: Raw 데이터를 저장, 유저 데이터 분석 및 분석된 데이터를 저장한다.
- Raw Data DB: Riot api를 통해 가져온 데이터가 저장된다.
- Analyzed Data DB: Raw Data DB에 저장된 데이터는 분석 과정을 거친 후 Analyzed Data DB에 저장된다.
- Front-end 서버: 사용자와 직접적인 통신을 담당하는 웹서버이다.
  - View는 사용자에게 출력되는 페이지를 보여준다.
  - Model은 사용자에게 출력될 데이터를 저장한다.

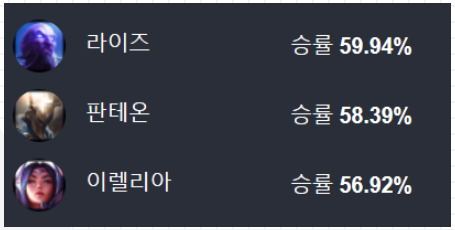


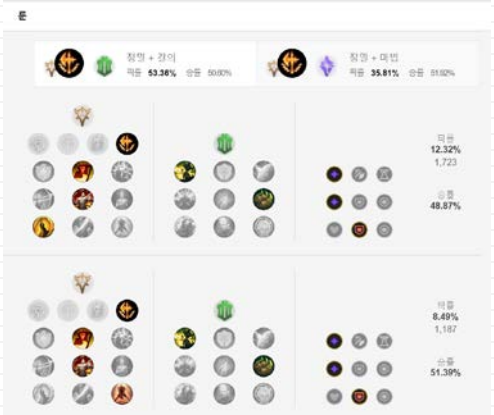
Prototype



Prototype





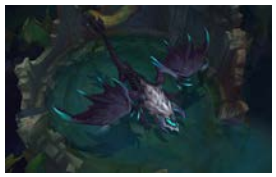


Prototype

오브젝트 분석



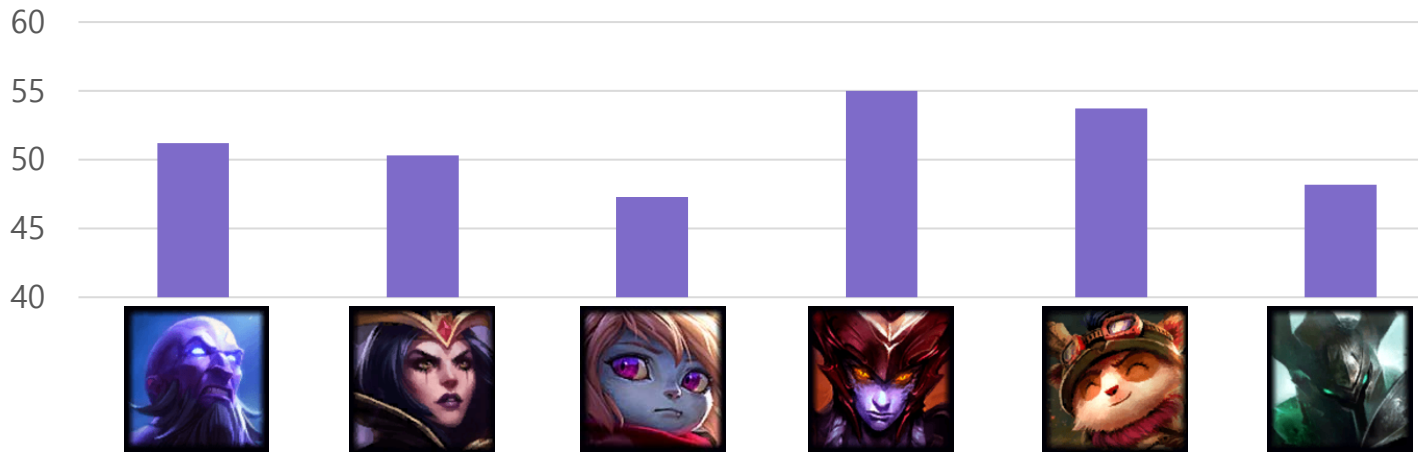


1	0	0	0	1	1

승률: 51.2%

챔피언별 승률



User Test Case

NUM	REQUIREMENTS	TEST CASE
a.1.1	사용자 챔피언 검색	사용자가 선택한 챔피언의 정보가 정확하게 보여진다.
a.1.2	상대 챔피언 검색	사용자가 선택한 상대 챔피언의 정보가 정확하게 보여진다.
a.1.3	랭크 선택	사용자가 선택한 랭크와 챔피언에 해당하는 정보가 정확하게 보여진다.
a.1.4	카운터 챔피언 확인	해당 챔피언을 선택할 때 밴 해야 하는 챔피언이 무엇인지 보여준다.
a.2.1	전적 검색	사용자가 자신의 소환사명을 입력하면 해당 소환사의 게임 전적 정보를 보여준다.
a.2.2	프로필 검색	사용자가 자신의 소환사명을 입력하면 해당 소환사의 프로필 아이콘, 현재 속해 있는 리그 정보를 보여준다.
a.2.3	챔피언 추천	사용자가 자신의 소환사명을 입력하면 해당 소환사 개인의 챔피언 성향을 분석해 잘 어울리는 챔피언을 추천해준다.
a.2.4	아이템 트리 추천	사용자가 자신의 소환사명을 입력하면 사용자의 현재 매치에서 상대 챔피언들을 분석해 사용자의 아이템 트리를 추천해준다.
a.3.1	오브젝트 분석	오브젝트 분석 탭을 클릭하면 각 랭크 별로 오브젝트(바론, 드래곤 등)를 어떻게 가져갔는지 보여주고 승률을 보여준다.

System Test Case

NUM	REQUIREMENTS	TEST CASE
c.1	Raw 데이터 수집	Riot api로부터 필요한 데이터셋을 수집해서 누락 없이 데이터베이스에 저장시킨다.
c.2	데이터 분석	Raw 데이터를 조합해 머신 러닝, 딥 러닝 수행 후 보기 쉬운 데이터로 시각화 시킨다.
c.3	데이터 저장	분석된 데이터를 누락없이 잘 저장시킨다.
c.4	성능 측정 및 분석	학습한 모델의 성능을 측정한다.