CodeStates Project - CS2

Game Log Embedding

AI_05_최중훈

목차

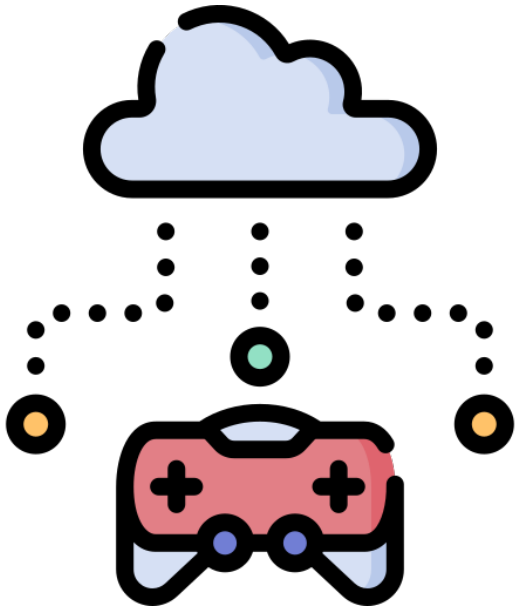
- 프로젝트 개요
- 프로젝트 설명 및 결과
- 개선방안
- 회고

프로젝트 개요

프로젝트 개요

프로젝트 배경 및 목적

- AFI는 게임서버 임대 서비스 '뒤끝'을 운영중
- 여러 게임사들로부터 축적되는 게임 유저들의 로그 데이터를 이용하여 General Embedding Tensor 를 생성
- 생성된 General Embedding Tensor를 유저 이탈, 구매예측, 불량유저 감지 등 Downstream Task에 이용



프로젝트 설명 및 결과

프로젝트 설명 및 결과

데이터 소개

- 2021년 12월 28일부터 2022년 1월 11일까지 총 191.47GB의 CRUD 로그데이터
- game_id, gamer_id, inDate, url, method, tableAndColum으로 구성

프로젝트 설명 및 결과

전처리 알고리즘

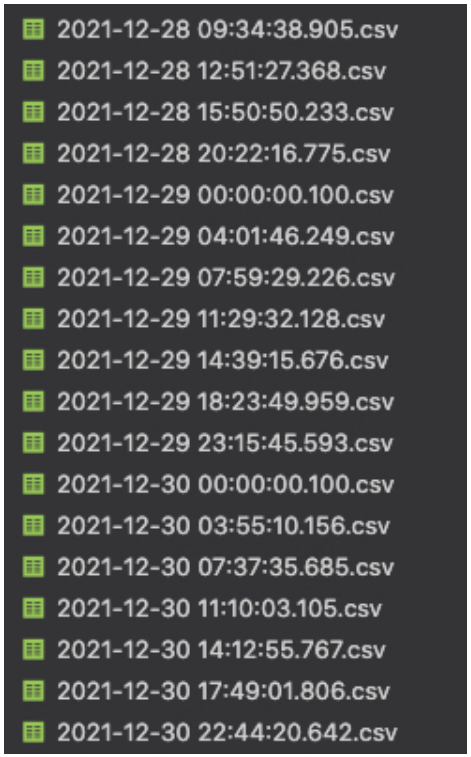
- 파일용량이 커 read_csv()로 한번에 불러올 수 없음
  - read_line()을 이용하여 csv파일의 데이터를 한줄씩 불러옴
  - 1000만행 단위로 전처리 후 분할저장

```
for i in range(10000000):
    line = f.readline()
    line = line.rstrip('\n')
    line = line.strip('"')
    line = line.strip('",')
    line = line.split('","')
    if line[0] == '':
        break
    df.append(line)
print(len(df))
print("csv name : ", csv)
```

프로젝트 설명 및 결과

전처리 알고리즘

- 시계열 데이터 특성상 시간, 순서 데이터가 중요
  - 파일단위로 inDate컬럼 기준 오름차순 정렬
  - 파일간 순서파악을 위한 파일이름 지정



프로젝트 설명 및 결과

전처리 알고리즘

- 게임 아이디가 중요하다고 판단
  - game_id와 url을 결합하여 사용자별 로그벡터 생성

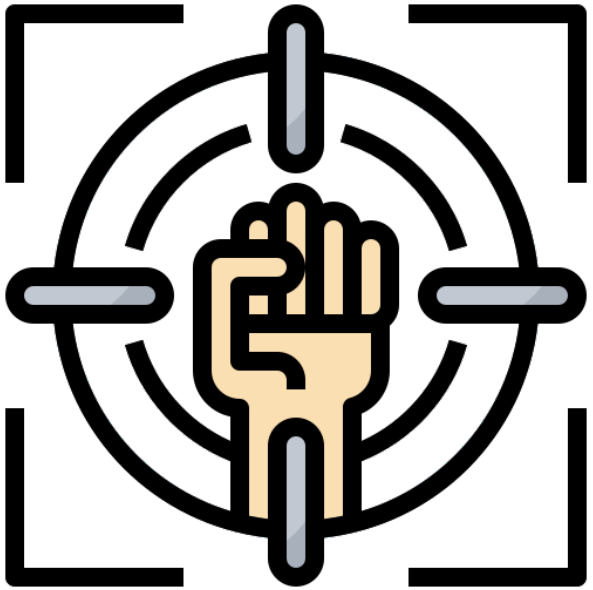
```
def transform(df):
    df = df.sort_values(by = 'inDate').reset_index(drop=True)
    # df['url2'] = df['game_id'].map(str) + df['url'].apply(lambda x : regex(x))
    df['url2'] = df['game_id'].map(str) + df['url']
    df = df.groupby(df['gamer_id'])['url2'].apply(list)
    df = pd.DataFrame(df)
    return df
```

개선방안

한계 및 개선방안

개선방안

- 정규표현식 적용
- Dense Vector
- 성능평가



회고

회고

돌아보기

- 현업 업무 간접경험
- 일정관리, 시간분배
- 대용량 데이터 다루는 경험
- 데이터의 특성을 고려한 전처리