SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

# 08 PJT

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

# Django 에서 알고리즘 구현 및 성능측정

## 챕터의 포인트

- [도전] 알고리즘 구현 및 성능 측정
- 제출

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

# 알고리즘 구현 및 성능 측정

## 공통 요구사항

- Locust 테스트 시 활용한 Django 프로젝트를 그대로 활용하거나, 새로 만드셔도 무관합니다.
- 제공된 CSV 파일(test_data.CSV) 를 활용합니다.
- .gitignore 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 명시된 요구사항 이외에는 자유롭게 작성해도 무관합니다.

## 세부 요구사항

A. CSV 데이터를 DataFrame 으로 변환 후 반환

B. 결측치 처리 후 데이터 반환

C. 알고리즘 구현하기(평균 나이와 가장 비슷한 10명)

D. Locust 를 활용한 알고리즘 성능 측정

- A, B, C 요구 사항에 따라 URL 패턴과 이에 매칭되는 View 함수를 각각 구현합니다.

# A. CSV 데이터를 DataFrame 으로 변환 후 반환

- 제공한 데이터(data/test_data.CSV)를 Django 에서 읽어오도록 구현합니다.
  - 프로젝트 경로와 동일한 폴더에 data 폴더를 저장합니다.
- Numpy 혹은 Pandas 의 CSV 를 읽어오는 함수를 활용하여 완성합니다.
- DataFrame 생성 시 columns 옵션을 적절히 활용합니다.
- [참고] DataFrame 은 아래 방법을 사용하여 반환할 수 있습니다.

```
# records: 리스트 원소를 각각 하나의 레코드로 만들기 위해 주는 옵션
data = df.to_dict('records')

# JSON 형태로 응답합니다.
return JsonResponse({ 'dat': data })
```

## B. 결측치 처리 후 데이터 반환

- Pandas 라이브러리의 특정 값 반환 함수를 활용합니다.
  - 제공한 3.Pandas_Advanced.md 파일을 참고합니다.
- 비어 있는 값을 “NULL” 문자열로 치환 후 DataFrame 을 반환합니다.

# C. 알고리즘 구현하기(평균 나이와 가장 비슷한 10명)

- DataFrame 의 “나이” 필드를 활용합니다.
- 평균 나이와 가장 비슷한 나이인 10개 행을 새로운 DataFrame 으로 만들어 반환합니다.
- “나이” 필드에 대해 평균값을 구할 수 있도록 DataFrame 을 전역 변수로 선언하여 문제 해결에 활용합니다.
- “나이” 필드의 데이터 중, 결측치를 제외한 데이터들에 대하여 평균을 계산합니다.

# D. Loucst 를 활용한 알고리즘 성능 측정

- C 번에서 구현한 함수를 활용합니다.
- C 번의 함수를 테스트 할 수 있도록 Locust 스크립트 파일을 작성합니다.
- 총 접속자 수와 동시 접속자 수를 변경하며 여러 번 성능 테스트를 시도합니다.
  - README.md 파일에 “총 접속자, 동시 접속자, 평균 RPS, 응답 시간“ 을 기록합니다.
- 다른 사람들이 구현한 알고리즘과 나의 알고리즘의 성능을 비교해봅니다.
  - PC 성능에 따라 다른 결과가 나올 수 있으므로, 코드를 전달 받도록 합니다.
  - 내 PC 에서 다른 사람들의 알고리즘을 동작 시켜보고, 결과를 기록합니다.
- 나의 알고리즘과 성능 차이가 나는 이유를 분석하여 README.md 에 작성합니다.

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

# 구현 시 참고사항

## 데이터 형식 - CSV

- 몇 가지 필드를 쉼표(,)로 구분한 텍스트 데이터 및 텍스트 파일
- 일반적으로 표 형식의 데이터가 일반 텍스트 형태로 저장됩니다.
- 엑셀과 비교하였을 때, 일반 텍스트로 저장되므로 저장 및 전송하고 처리할 수 있는 프로그램이 다양하다는 큰 장점이 있습니다.

# 데이터 형식 - CSV

- 표 형식의 데이터

| 이름 | 생년 | 월 | 일 | 성별 | 직업 | 사는 곳 |
|---|---|---|---|---|---|---|
| 홍길동 | 1992 | 7 | 17 | 남 | 강사 | 서울 |
| 희동이 | 1997 | 4 | 3 | 여 | 학생 | 대구 |
| 금땡구 | 1988 | 2 | 15 | 남 | 개발자 | 광주 |

- csv 로 표현하면 아래와 같습니다.

```
이름,생년,월,일,성별,직업,사는 곳
홍길동,1992,7,17,남,강사,서울
희동이,1997,4,3,여,학생,대구
금땡구,1988,2,15,남,개발자,광주
```

## JSON vs CSV

- 둘 모두 서로 다른 시스템이나 디바이스 간에 데이터를 쉽게 교환할 수 있도록 도와줍니다.

| 특징 | JSON | CSV |
|---|---|---|
| 장점 | - 모양과 규칙 자체가 단순해서 타 언어에서도 구현하기가 쉽다. | - 용량이 가장 작다.<br>- csv는 용량이 작기 때문에 변하지 않는 많은 양의 데이터를 제공할 때 주로 이용이 가능하다. |
| 단점 | - 콤마가 누락되거나 중괄호가 잘못 닫히는 등 문법 오류에 취약하다. | - 데이터의 의미를 표시할 수 없기 때문에 데이터가 많아지면 어떤 데이터가 항목을 나타내는지 직관적인 이해가 어렵다. |
| 주요 사용처 | - 서버 통신 REST API를 사용할 때 가장 많이 사용 | - 간단한 테이블 작성 또는 읽는 속도가 중요한 부분에서 사용 |

## 알고리즘 구현 시 활용할 수 있는 라이브러리

- 데이터 사이언스 3종 패키지
  - Numpy, Pandas, Matplotlib

# Numpy

- 다차원 배열을 쉽게 처리하고 효율적으로 사용할 수 있도록 지원하는 파이썬 패키지
- 장점
  - Numpy 행렬 연산은 데이터가 많을수록 Python 반복문에 비해 훨씬 빠르다 !
  - 다차원 행렬 자료 구조를 제공하여 개발하기 편리합니다.
- 특징
  - CPython(공식 사이트의 Python) 에서만 사용 가능
  - 행렬 인덱싱(Array Indexing) 기능 제공

## Pandas

- Numpy 의 한계
  - 유연성(데이터에 레이블을 붙이거나, 누락된 데이터로 작업)이 부족함
  - 그룹화, 피벗 등 구조화가 부족함
- Pandas 는 마치 프로그래밍 버전의 엑셀을 다루듯 고성능의 데이터 구조를 만들 수 있음
- Numpy 기반으로 만들어진 패키지로, Series(1차원 배열) 과 DataFrame(2차원 배열) 이라는 효율적인 자료구조 제공

## 실습 파일

- 파일
  - Numpy_Basic.ipynb - Numpy 기초 코드
  - Pandas_Baisc.ipynb - Pandas 기초 코드
  - Pandas_Advanced.ipynb - Pandas 응용 코드

- Pandas 학습 - 테스트 데이터
  - data/test_data.CSV - 결측치 미포함 테스트 데이터 (50개)
  - data/test_data_has_null.CSV - 결측치 포함 테스트 데이터 (50개)

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

# 제출

# 제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
  - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- https://lab.ssafy.com/ 에 프로젝트를 생성하고 제출합니다.
  - 프로젝트 이름은 '프로젝트 번호 + pjt' 로 지정합니다. (ex. 01_pjt)
- 반드시 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.