Задача: Написать micro-framework для вычисления метрик использования online сервиса.

Описание задачи: На вход подаются три csv-файла: 1 - справочная информация о пользователях, 2,3 - отвечающие за события. Необходимо смешать события в один pandas-dataframe и обогатить их данными из первого csv-файла, после чего сгруппировать данные по пользователям и отсортировать их по времени. Метрики вычисляются по пользователям в час. Для того чтобы упростить жизнь аналитикам, необходимо предоставить framework в котором аналитик получает отсортированный лог (pandas-dataframe) событий за час по юзеру и вычисляет некую метрику или набор метрик, а потом эти метрики возвращает. По сути framework предоставляет возможность написать пользователями (аналитиками) небольшой модуль или функцию (которая, собтсвенно, и осуществляет аггрегацию). В неё будут переданы отсортированные данные событий за час конкретного пользователя и забраны метрики. В итоге, все метрики объединяются для пользователя как столбцы и записываются в выходной csv файл. Модули аналитики складывают в строго определённую папку из которой они динамически импортируются, при этом у модуля будет свойство включен/выключен чтобы импортировать только нужные. Модуль наследуется от общего родителя.

Функция модуля (маппер), которая расчитывает метркии для конкретного пользователя должна вернуть key-value структуру (например dictionary), соотвественно для набора модулей нужно конкатенировать эти возвращаемые значения по всем модулям.

Пример структуры модуля:

def **mapper**(each\_grouped\_pandas\_df):

#magic

return {‘metric1’: value1, …, ‘metricK’: valueK}

module\_x = ModuleBuilder(mapper = **mapper**, other\_data=other\_data)

Входные файлы:

User data:

| user\_id | sex | name | age | phone |

Page views:

| timestamp | user\_id | page\_name | referrer |

Backend Events:

| timestamp | user\_id | event\_name | parameters\_value |

Далее всё это смешивается, вызываются модули (функции) аналитиков, принимаются от них метрики, объединяем метрики и складываем в выходный файл

Выход:

Metrics:

| user\_id | datetime | metric\_1 | metric\_2| …… | metric\_n |

Пример:

Вход:

users.csv

|123| male | John | 35 | None |

|124| male | Martin | 40 | None |

Pageviews.csv

| 1469627670 | 123 | home.htm | <https://google.com> |

| 1469627675 | 123 | home.htm | <https://google.com> |

| 1469627676 | 124 | price.htm | <https://google.com> |

| 1469627677 | 123 | solutions.htm | <https://google.com> |

Backend.csv

| 1469627670 | 123 | get\_metadata | {“dashboard\_id”: 5} |

| 1469627678 | 124 | get\_metadata | {“dashboard\_id”: 6} |

| 1469627680 | 123 | get\_metadata | {“dashboard\_id”: 7, “filter”: 1} |

После первого шага данные должны выглядеть так:

| 1469627670 | 123 | home.htm | <https://google.com> | | | male | John | 35 | None |

| 1469627675 | 123 | home.htm | <https://google.com> | | | male | John | 35 | None |

| 1469627676 | 124 | price.htm | <https://google.com> | | | male | Martin | 40 | None

| 1469627677 | 123 | solutions.htm | <https://google.com> | | | male | John | 35 | None |

| 1469627670 | 123 | | | get\_metadata | {“dashboard\_id”: 5} | male | John | 35 | None |

| 1469627678 | 124 | | | get\_metadata | {“dashboard\_id”: 6} | male | Martin | 40 | None

| 1469627680 | 123 | | | get\_metadata | {“dashboard\_id”: 7, “filter”: 1} | male | John | 35 | None |

А одна из пачек что должна прилететь на обработку аналитическому модулю выглядит так:

| 1469627670 | 123 | home.htm | <https://google.com> | | | male | John | 35 | None |

| 1469627670 | 123 | | | get\_metadata | {“dashboard\_id”: 5} | male | John | 35 | None |

| 1469627675 | 123 | home.htm | <https://google.com> | | | male | John | 35 | None |

| 1469627677 | 123 | solutions.htm | <https://google.com> | | | male | John | 35 | None |

| 1469627680 | 123 | | | get\_metadata | {“dashboard\_id”: 7, “filter”: 1} | male | John | 35 | None |

На выходе получаем к примеру две метрики, колечество просмотренных домашних страниц и просмотренная страница solutions после домашки.

| user| datetime | home views | solutions\_after\_home |

| 123 | 2016-08-10 12:00:00| 2 | 1 |