# XITE Data Engineer assignment 2018

## Code logics

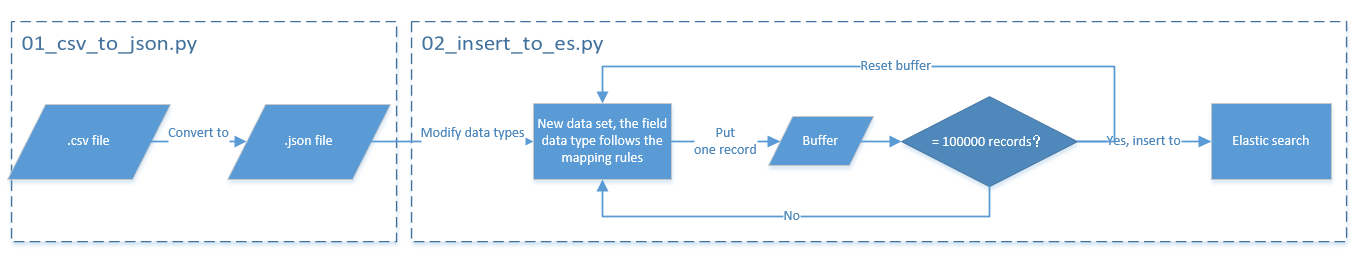


Figure 1 logics

## Usage

1. Open config.cfg, set up the configurations.
2. Run 01\_csv\_to\_json.py
3. Run 02\_insert\_to\_es.py

## Anomalies in data set

### Data pipeline abnormal

As shown in Figure 2, Figure 3, Figure 4, we can see that data is missing:

* + - from 2018-01-09 00:30 to 2018-01-09 01:30(Figure 2);
    - from 2018-01-13 23:30 to 2018-01-14 01:00(Figure 3);
    - from 2018-01-18 20:30 to 2018-01-19 01:00(Figure 4);

It seems like you were doing some operations during these periods and shut down the data pipeline.

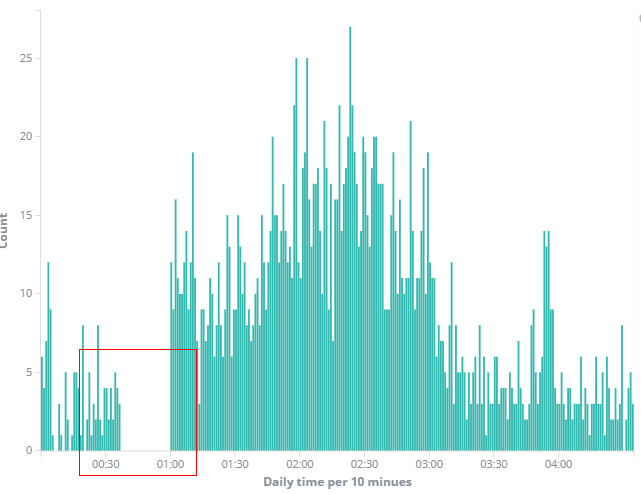


Figure 2 2018-01-09 00:00:00 to 2018-01-09 04:30:00

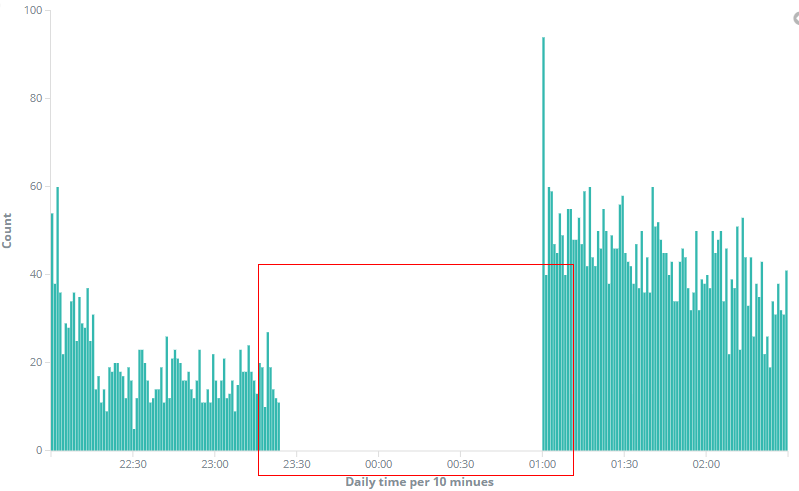


Figure 3 2018-01-13 22:00:00 to 2018-01-14 02:30:00

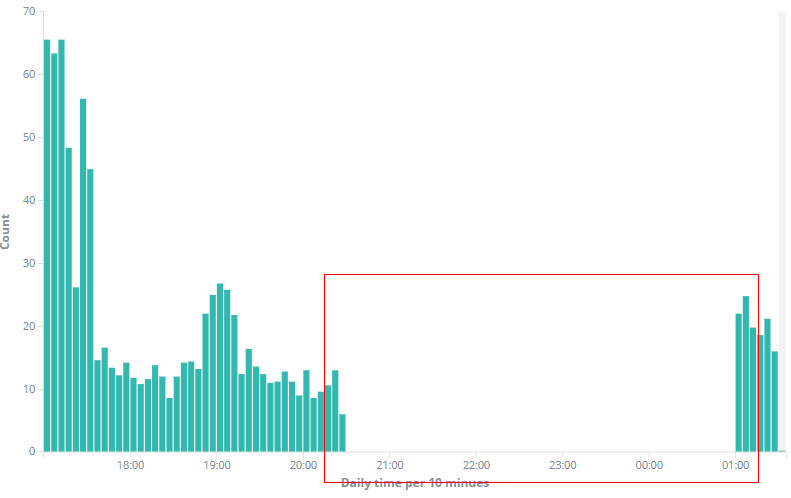


Figure 4 2018-01-18 17:00:00 to 2018-01-19 02:30:00

### Function abnormal or data missing

##### “isSearch” field

The “isSearch” field is empty for all records in January (Figure 5).

It seems like either the search function does not work or the data is missing.

## 

Figure 5 Results for “isSearch” field

##### “actionType” field

The “action Type” field is empty from 2018-01-01 to 2018-01-15 (Figure 7).

It seems like

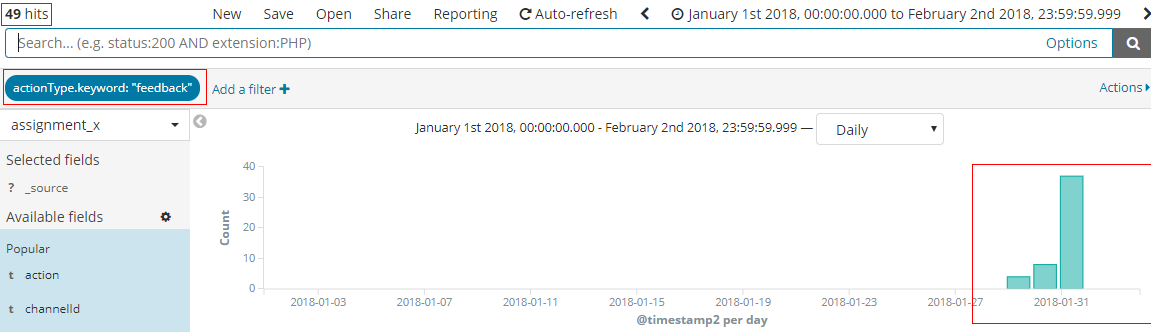


Figure 6 “action Type”:”feedback” overview in January

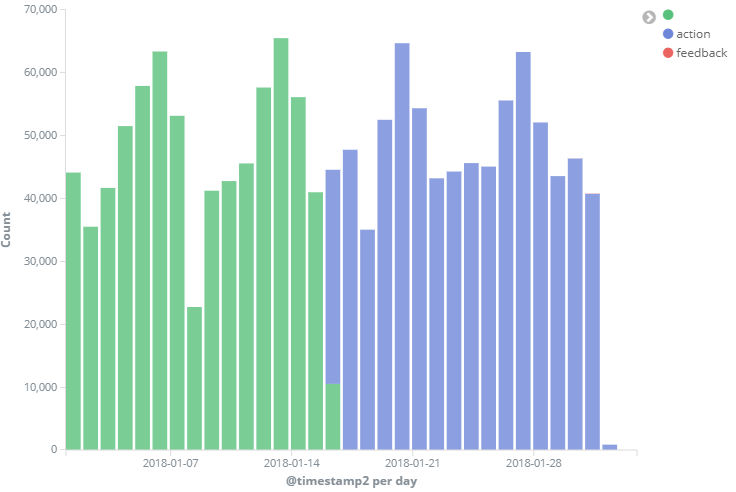


Figure 7 “action Type” field overview in January

##### “channelid” field

There are 44862 records without “channelid” value.

It seems like the data wasn’t collected.

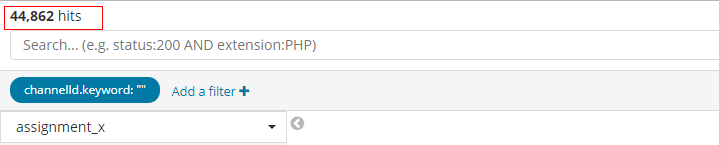


Figure 8 Result for “channelid” is null

## Visualization

Table 1 items that are useful for business purpose

|  |  |  |
| --- | --- | --- |
| Items | Usage | example |
| action | To assess if users like provided video. | Figure 9 |
| timestamp | Check the degree of activity | Figure 10 |
| Action, video id | Find the popular and unpopular videos | Figure 11 |
| Duration | Find the best time to place ads, recommendations, etc. | Figure 12 |
| Duration, device id | Find active users | Figure 13 |

For the following visualization and analysis, I exclude all the test account activities.

### User activity types overview

In Figure 9, we can see the ratio of skip, play and expired actions in the entire month. Therefore we can have a rough overview about if the users enjoy the provided videos or not.

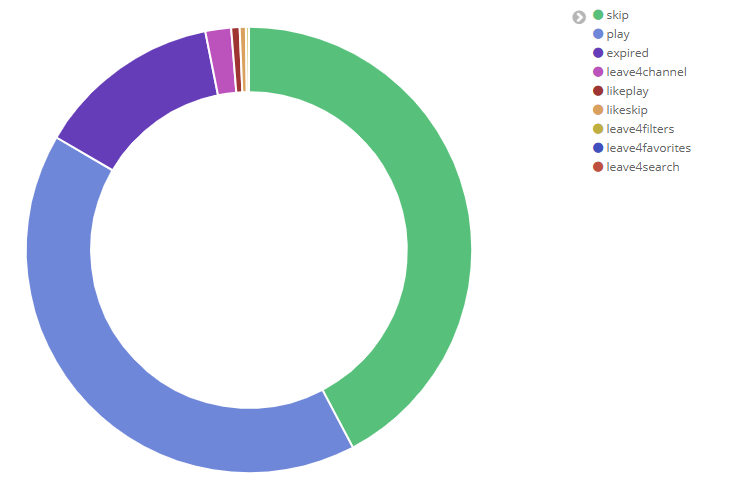


Figure 9 User activities overview in January

### Degree of activity regarding to daily hours

In Figure 10, x axis shows the daily hours from 01 to 24, y axis shows the count of records within each daily hour in January.

From Figure 10, we can see that most uses tend to watch videos between 10:00 to 18:00.

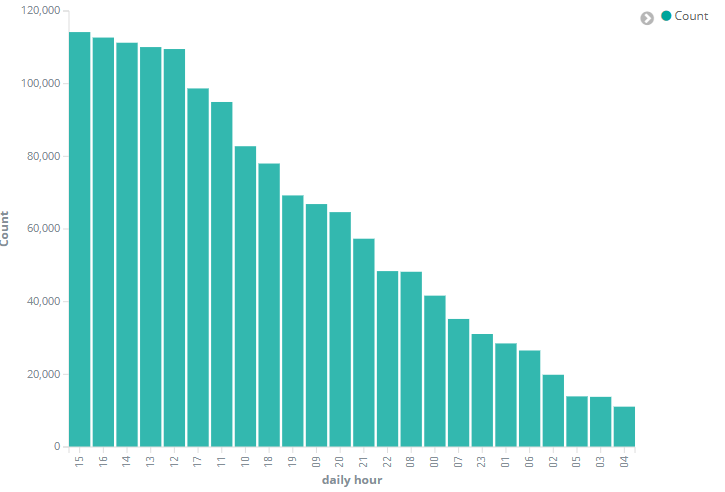


Figure 10 user activity regarding to daily hours in Jan.

### Top hit videos in January

Figure 11 shows the top hit videos that have been provided in January. From color shading between ‘play’ and ‘skip’ we can see which videos are popular and which videos are unwanted.

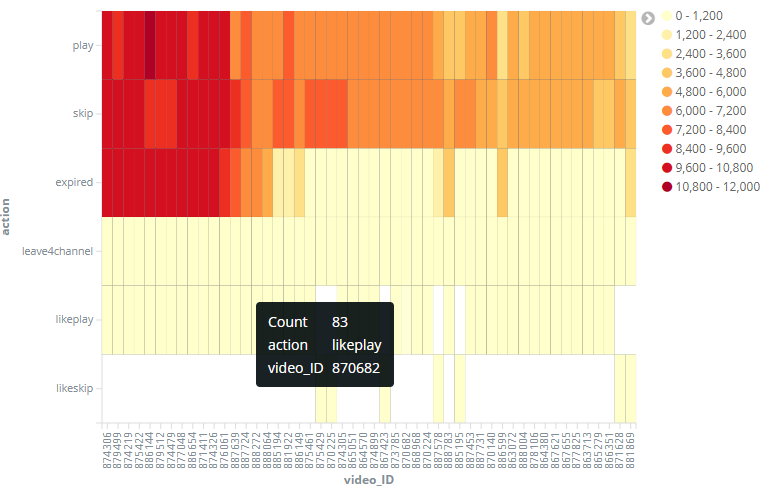


Figure 11 Top hit videos in Jan.

### User activity regarding to daily hours

From Figure 12, we can see that users tend to skip the video before 10 seconds and at around 32 seconds.

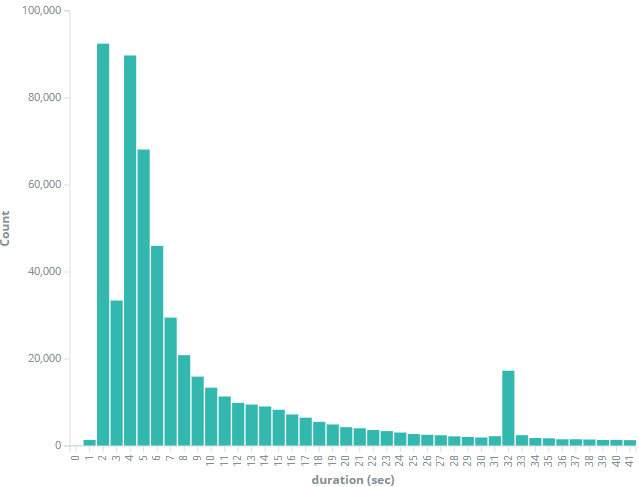


Figure 12 moments of skip action performed

### Active users

Figure 13 shows which devices spend most of time in January. Device id is hidden on x axis.

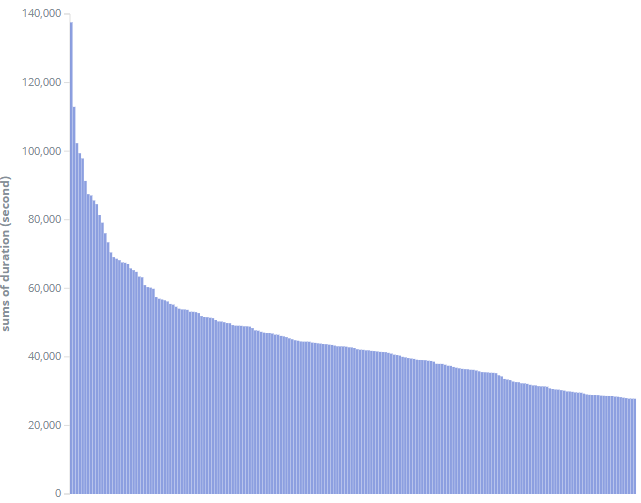


Figure 13 the sum duration of each device (device id is hidden on x axis)

## Proposal

### Data source

I checked the first 10000 records when I received the data set. I plotted them using row number as x axis and time stamp as y axis. I expected to have a near linearity plot, but what I got is a mess. So I assume these data may be collected from data warehouse, instead of live streaming pipe line.

I suggest we should be able to collect data from live streaming pipe line and visualize them in real time to monitor user’s activity and react to them.

### Data pipe line architecture

To increase the reliability of the data pipe line:

I don’t know Google cloud, so I use Amazon web service as an example

1. Use bootstrap scripts to install apache Kafka and set up configurations on multiple EC2 machines in different availability zones. Let’s take 3 EC2 machines as an example.
2. The setting for Kafka topics should include 3 replicas.
3. Put these EC2 machines behind Elastic load balancer and set up correct settings. If one or two EC2 machines go down. Elastic load balancer will automatically pull up new EC2 machines and use bootstrap scripts to install apache Kafka and set up configurations, so that there will always be 3 EC2 running. The data pipe line will be more reliable.