Hey reader,

I was in charge of merging the 4 datasets found in the unprocessed folder into a single comprehensive dataset.

I removed extra information about our 50 customers and put that into the clientInfo csv file and all their clickstream details in finalAggregation.csv

The unprocessed folders were not in homogenous form, the json files were not even in proper json format which disallowed for proper analysis.

I mainly used the R program to endeavor the changes, but in the middle of summer, I was recommended to switch to python because there were more opportunities to use the various packages available there.

So a lot of the early work was done in R, later on, I switched to iPython notebook (which somehow allows for R, and matlab together!)

Analysis

So in the finalAggregation.csv file (I will hereon refer to only this file)

The only two possible ways of garnering analysis was to use the timestamp rather than the referrer\_id, page\_id columns. I was never really able to decipher the implementation of the original tracking extension that wrote the latter two columns and was never able to properly implement a method of analysis using the two columns because it was a bit difficult to understand. The references are mumble jumble and I wish you luck in finding out.

(switch to ipython notebook as a medium)

In using the timestamp, it is possible to define “Clickstreams” as sequences of clicks where juxtaposed clicks are no longer than X minutes apart.

I did analysis of said clickstreams as they could be clustered based on their time length (I was able to measure time between first and last click to get a general idea of how long they were clicking for).

I also pulled youtube data using their api to look at what they were watching.

Most simply, the youtube api offered by the pafy package in python allowed me to grab information like the videos title, description, category, and tags to allow for analysis.

My goal was to cluster videos by the topics that they encapsulated, I tried to use Latent Semantic Analysis on each video’s description, title, and tag data, but that became a time hassle, so I ended up just grouping the youtube videos by their category data.

Note that when you run some of the ipython code, the compiling time could be as much as 3-5 hours because of how youtube delays pull requests as not to cause a DDOS on their servers.