POC : Ad Recognition ML

Eléa Reggiani

Need :
Do not show the same ad twice or two competing ads in order to not damage the UX.

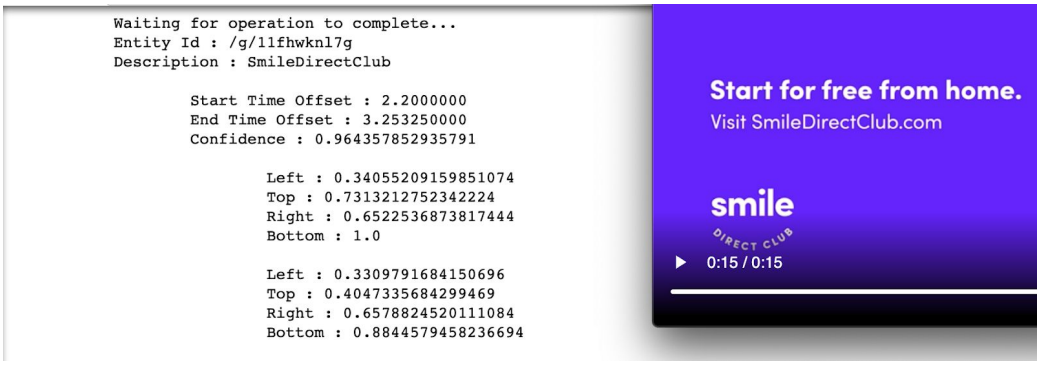
Road to success :
Create a program that retrieves urls, download videos from urls and use the Google API to recognize logos. The result is stored and cross-checked in a DB.
Two possible ways to go :
I am based on the ad before and if it's the same brand or the same category I do not display.
Or I could also check the time and compare in the same way (Ex : if there is the same ad/category twice in the last X minutes).

What do I need :
- Python 3
- Google Cloud Platform : (package: `google-cloud-storage`)
- API Google Videointelligence (package: `google-cloud-videointelligence`)
- URLs (CSV or Json or as an argument to the program)
- DB (MySQL)

Note : I chose Google Cloud Platform for this POC because of the freemium part and I'm more experienced with it.

Theoretical steps:
- Make a script which cleans all the urls
- Make a script that calls the Google API and the ML Videointelligence Google (we can also train our own model)
- Fill the DB with associated id(int auto_increment), url (varchar), brand (varchar), category (varchar), time (datetime), and displayed(tinyint).
- Choose X categories (car, make-up, sports, clothes…) and classify advertisers in categories (manually) or use a ML model depending on how many advertisers.
- Make a script which checks the DB with the current ad and set up the 'displayed' to 0 if the last ad is the same (or has the same category) as this one.

Practical steps:
- First step accomplished: Script is cleaning urls
- Second step accomplished : ML Google is recognising ad's logo



- Third step accomplished : Create DB on MySQL with id, brand, categories, url, time, displayed.
- Fourth step accomplished : Script is filling the DB
- Fifth step accomplished : Script is telling if we can display the ad or not (same ad)
- Sixth step : Create a table with advertiser and category in DB
- Seventh step : Script is telling if we can display the ad or not (depending on categories)
- Eighth step : Script is telling if we can display the ad or not (same ad or category in time lapse)

Next steps to get a real usable product :

- Do categories on a larger dataset (or use ML)
- Add a custom model with Google unknown brands
- Possible performance improvements for RTB :
    - Optimise with the datacenter regions, Edge...
    - Trim video, get frames from video
    - Compare hashed videos
    - Use every part of the project in the same environment : AWS logo rekognition ML or custom it + Sage Maker studio + RDS.