At Terragon we analyse behavioural, transactional, and demographic data to calculate propensity of profiles to respond to an ad served. Our Machine learning algorithms can forecast the likelihood that customers will purchase, install, unsubscribe, sign up or confirm a loan by finding patterns. By more intelligently analysing the intricacies and customer journeys of previous customers, we can better understand how we acquired leads and customers in the past, and who didn't convert in the end. Then, through personalised targeting, we can determine which prospects are similar to current customers and thus likely to convert in the end.

Problem statement

Given the demographic and behavioural patterns of profiles during a period and the past behaviours of profiles, **calculate the propensity of profiles to click** on an ad served. A model of this decision problem would allow us to know which customers to target while serving an ad and increase the Click Through Rate (CTR) of campaigns.

Such a solution would need to be deployed and serve predictions in realtime. Using any AWS/ Azure service or Docker deploy the model and serve the responses in less than 100ms.

This solution should be able to handle large requests, and resilient to failure.

Sample Docker API on port 3400.

HTTP POST: http://34.250.58.249:3400/model/api/{version}/?

INPUT

CSV/JSON with columns clearly named

OUTPUT

Sample output

```
1 ▾ {
2 |      "user_4": 3689,
3        "user_9": 3690,
4        "user_10": 6311,
5        "user_11": 6311,
6        "user_12": 6311,
7        "user_13": 6311,
8        "user_1": 14298,
9        "user_3": 14301,
10       "user_8": 14301,
11       "user_6": 16193,
12       "user_2": 16291,
13       "user_7": 16291,
14       "user_5": 983688
15 }
```

The output should be a score between 0 and 1.

Evaluation

For orienting and to provide an initial goal, a **micro-precision score of > 75% will be good, and > 85%** will be great. That's just the minimum threshold we expect, but a good report/coding practice can outweigh the final score in our internal evaluation.
If you want to use neural networks that's fine, but a simpler solution works as well.

Also the API must **handle bad requests** without breaking.

We expect a weekly update on progress made.

If any questions send an email to **datasquad@terragonltd.com**

Best of Luck.