Identifying geographic locations using OS based side channels

D. Sai Pavan CS14B041
S. Sai Teja Reddy CS14B051

Aim of the project

The aim of this project is to predict the city rendered in Google Maps using the process’ memory footprint.

The memory footprint is measured using the browser’s DRS (data resident size) aka RSS (resident set size). DRS value reflects the processes’ total size of its heap, stack, and mmap-allocated memory.

Background

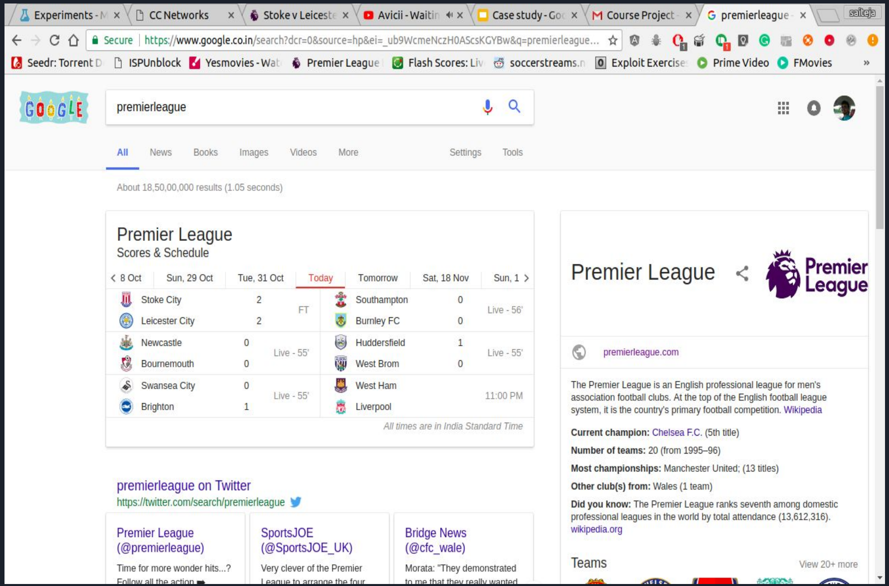
Consider two websites having very different layouts. They produce vastly different memory footprints.

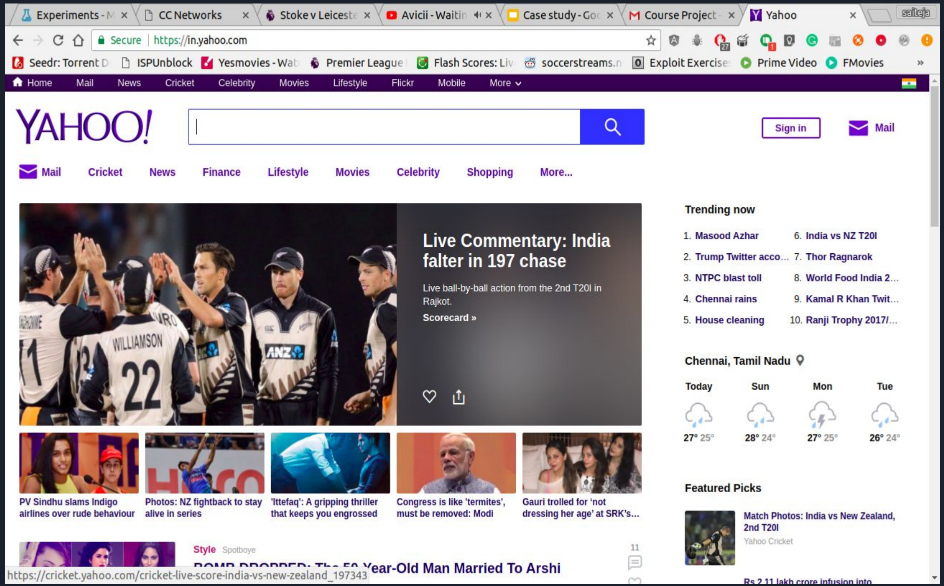
They can be classified using a matching algorithm based on a similarity index like Jaccard Index.

In Suman Jana and Vitaly Shmatikov’s 2012 paper, in a dataset of about 100,000 websites, they reported that about 30% of the websites are distinguishable.

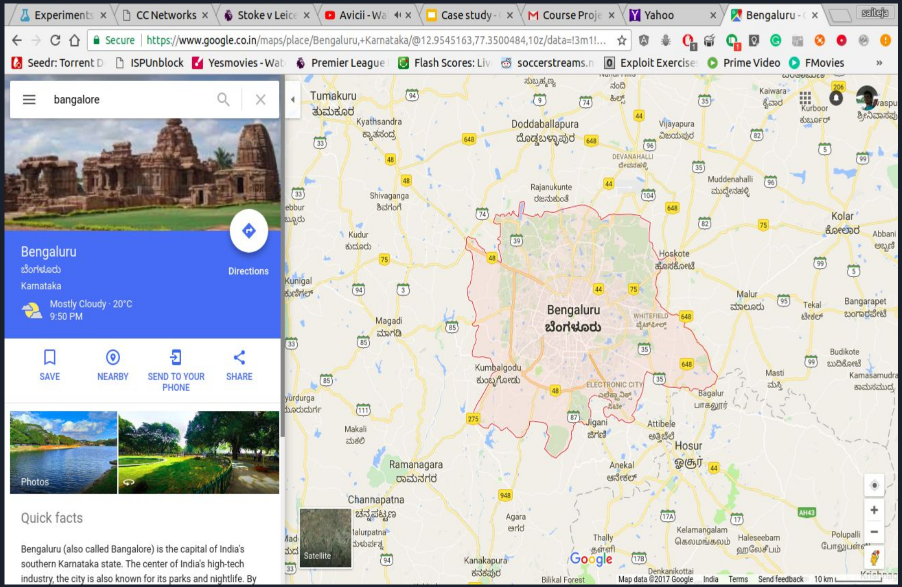
The Challenge

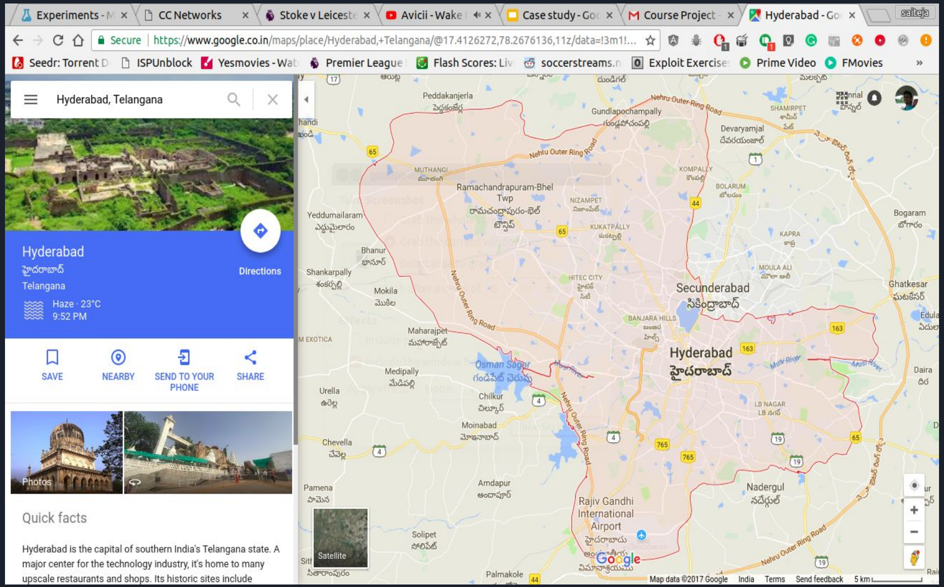
These have very low similarity while...





The Challenge
these have relatively high similarity.





Implementation

Collecting data

We used chrome to open the web sites.

Chrome uses a multi process architecture

- One main process
- One GPU process
- One process per tab, extension

We are interested in the Google map tab's process.

Reqd DRS value is in the 2nd column of /proc/<pid>/statm

Collecting data

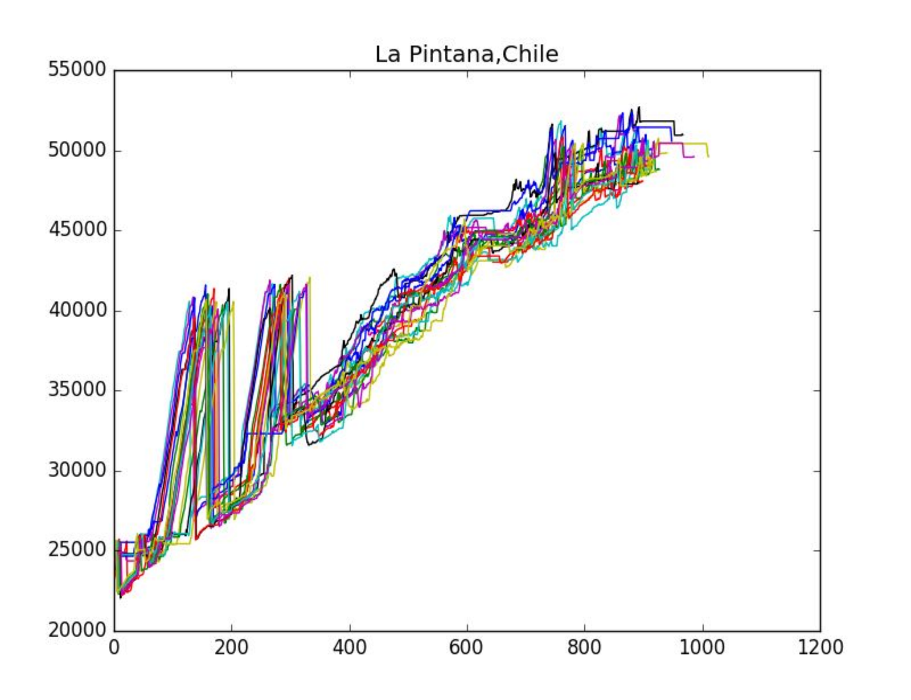
From a dataset of 23,000 cities we chose 50 cities randomly.

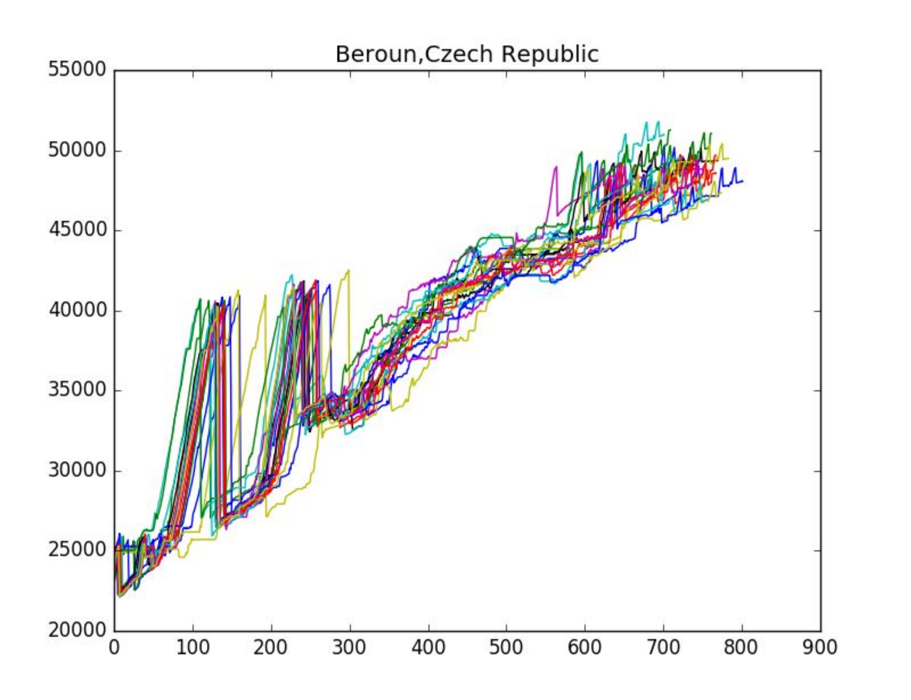
This was done to avoid selecting a biased metropolitan dataset.

We then proceeded to collect the DRS value for each city 20 times using an automated script written in python.

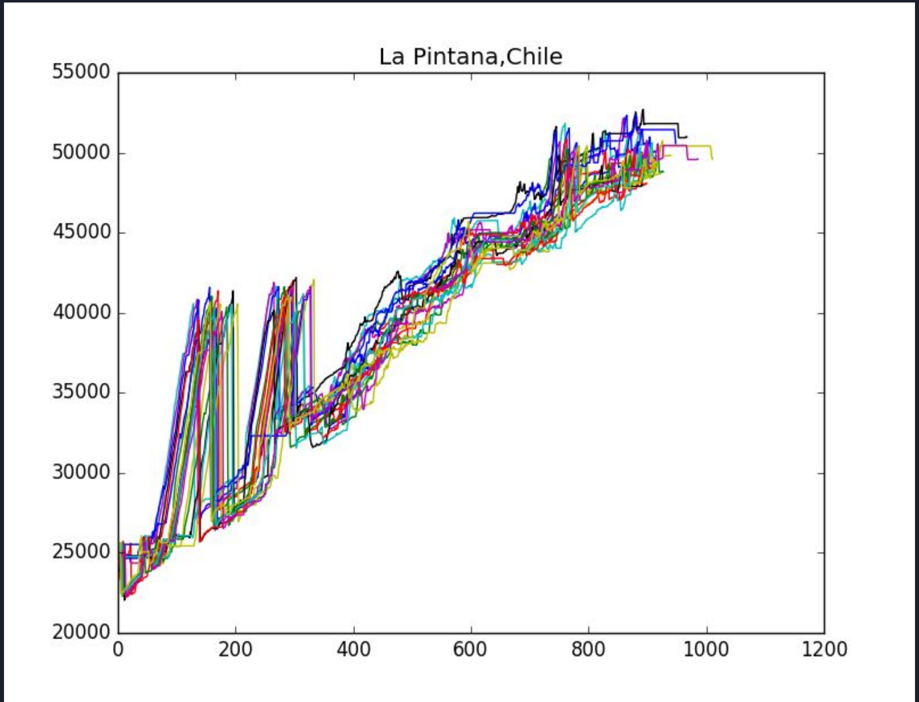
The DRS value was collected continuously until the Google Maps page finished loading.

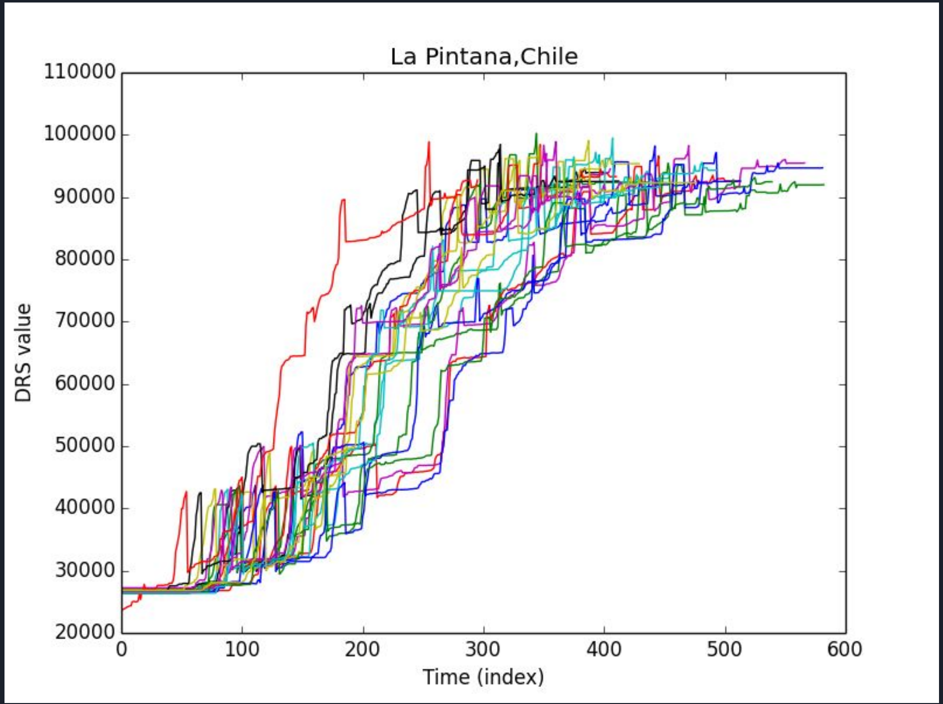
DRS value vs time for two different cities (on the same machine)





DRS value vs time for the same city on different machines





Analyzing data

We used the collected data to build a Random Forests machine learning model.

The feature vector for a city consisted of all the DRS values collected.

Cities with smaller features vectors were padded with their last obtained DRS values so as to have equal number of features.

We used Azure ML studio for this purpose.

Results

Using the obtained trained models we obtained an accuracy of 30.24% on validation data.

On the corresponding test data we obtained an accuracy of 29.59% .

Thus we were able to match the accuracy reported in Suman Jana and Vitaly Shmatikov’s 2012 paper for a relatively more difficult problem.

Improvements

The obtained results can be improved by

- Better data collection techniques
- Further fine tuning the Random Forest parameters.
- Using other techniques like RNNs.

References

Suman Jana and Vitaly Shmatikov's 2012 paper:

Memento: Learning Secrets from Process Footprints

http://ieeexplore.ieee.org/document/6234410/

Thank You