Expedia Kaggle Competition

Anyway, in that competition, we worked with lots of customer behavior.

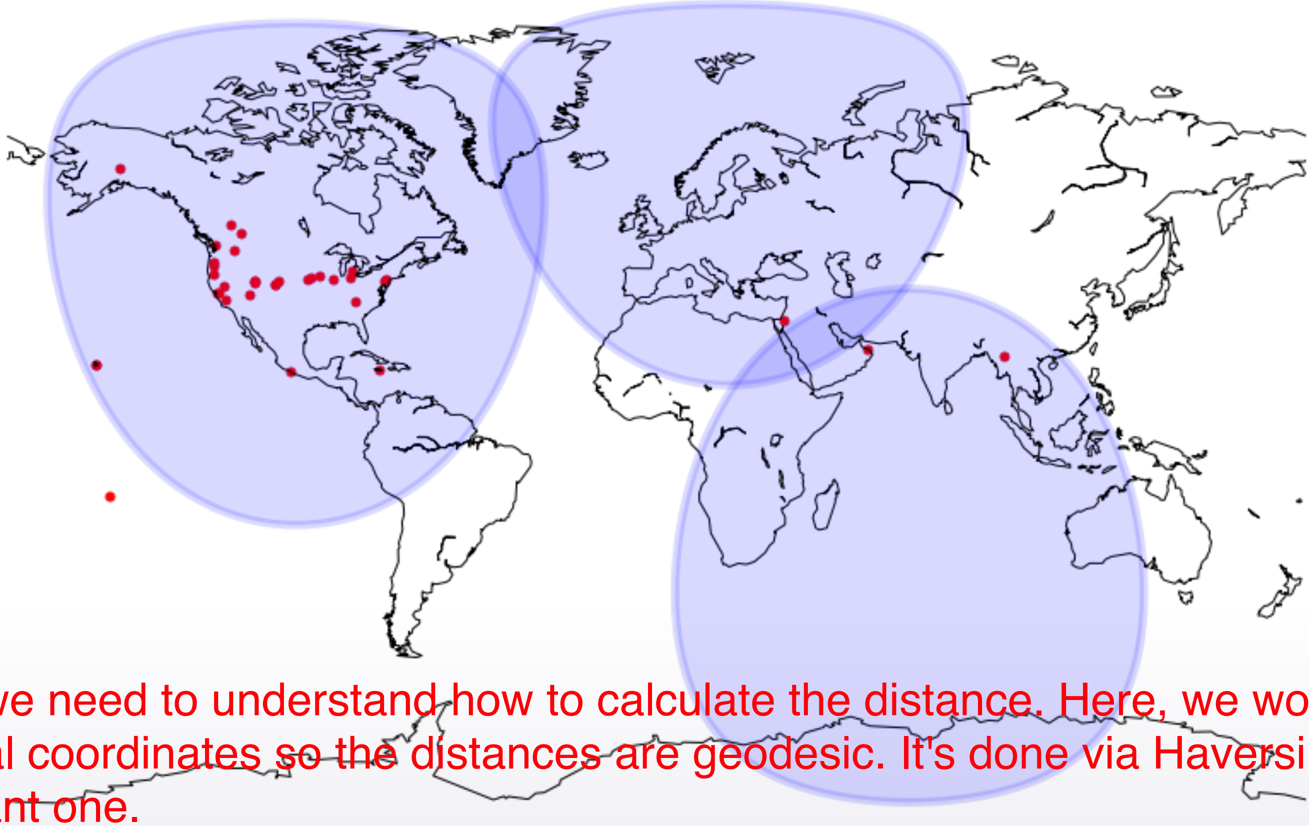
Important thing here is prediction target the hotel group. In other words, characteristics of actual hotel, remember it.

Data leakage

- destination_distance - user_city pair is a leak to true hotel location. A lot of matches between train and test.
- How to improve on that?
- Features based on counts on corteges of such nature
- Try to find the true coordinates

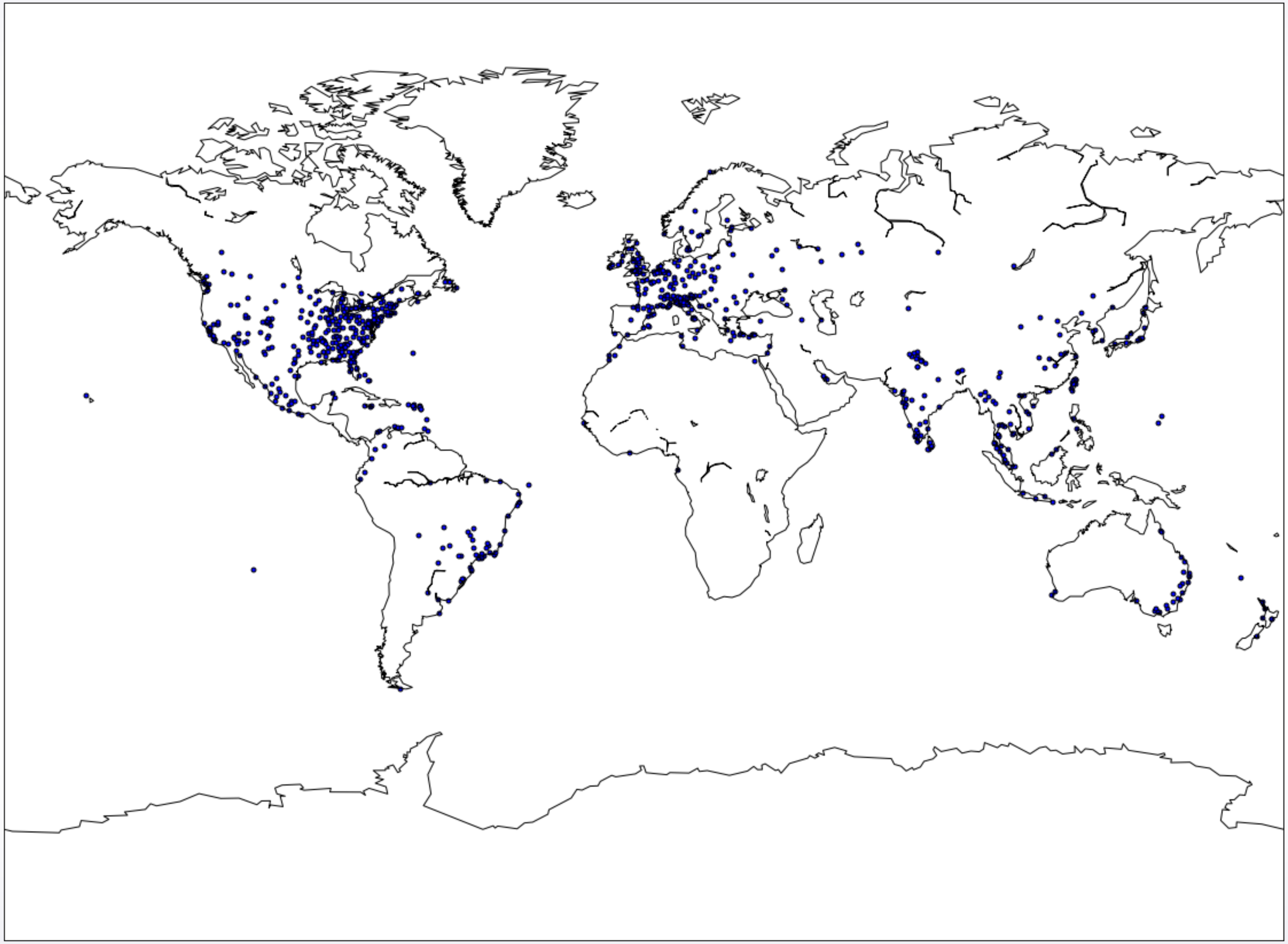
Spherical geometry

$$
\begin{aligned}
d &= 2r\arcsin\left(\sqrt{\operatorname{hav}(\varphi_2-\varphi_1)+\cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2-\lambda_1)}\right)\\
&= 2r\arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)+\cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)
\end{aligned}
$$

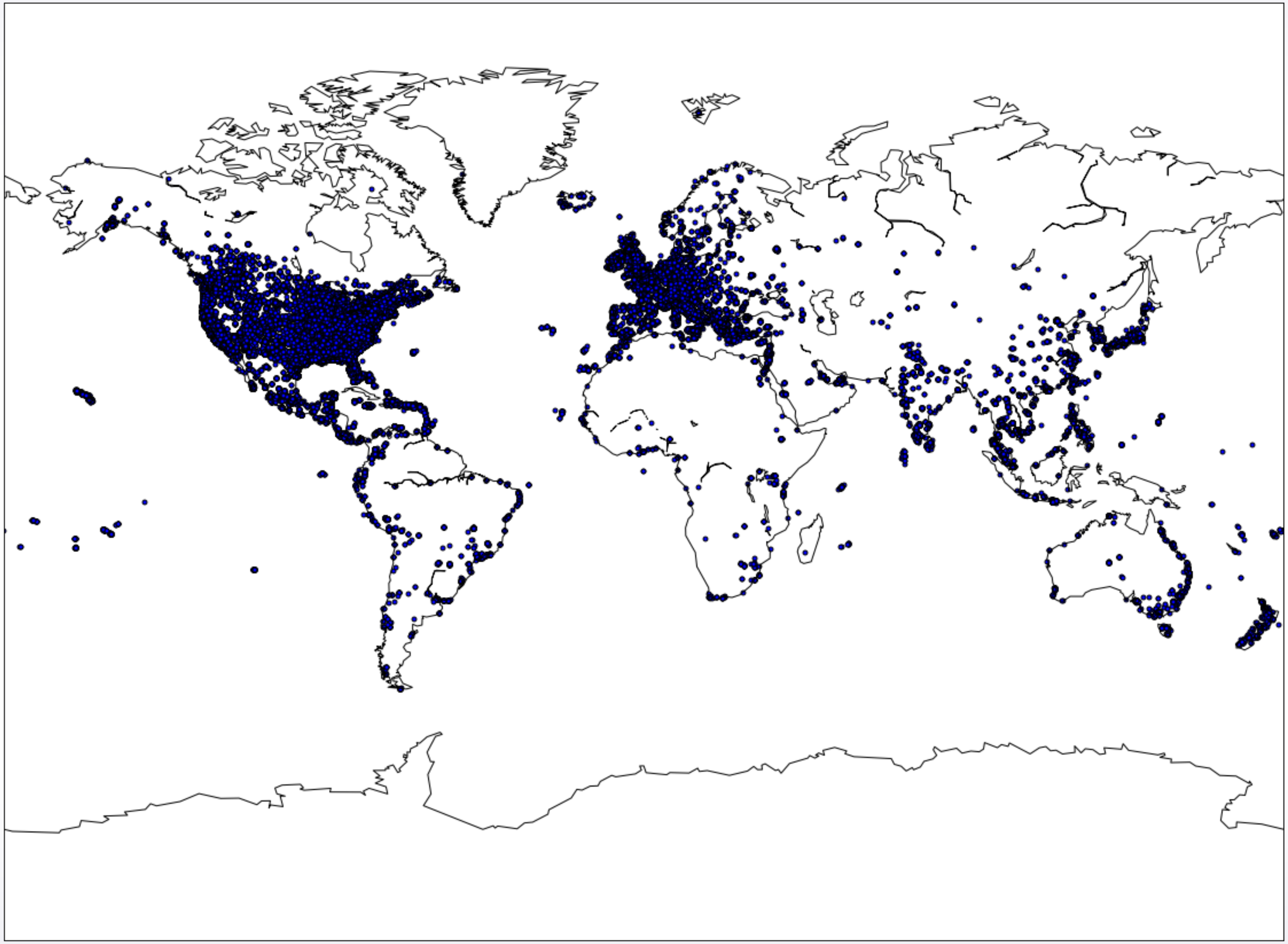


First of all, we need to understand how to calculate the distance. Here, we work with geographical coordinates so the distances are geodesic. It's done via Haversine formula, not a pleasant one.

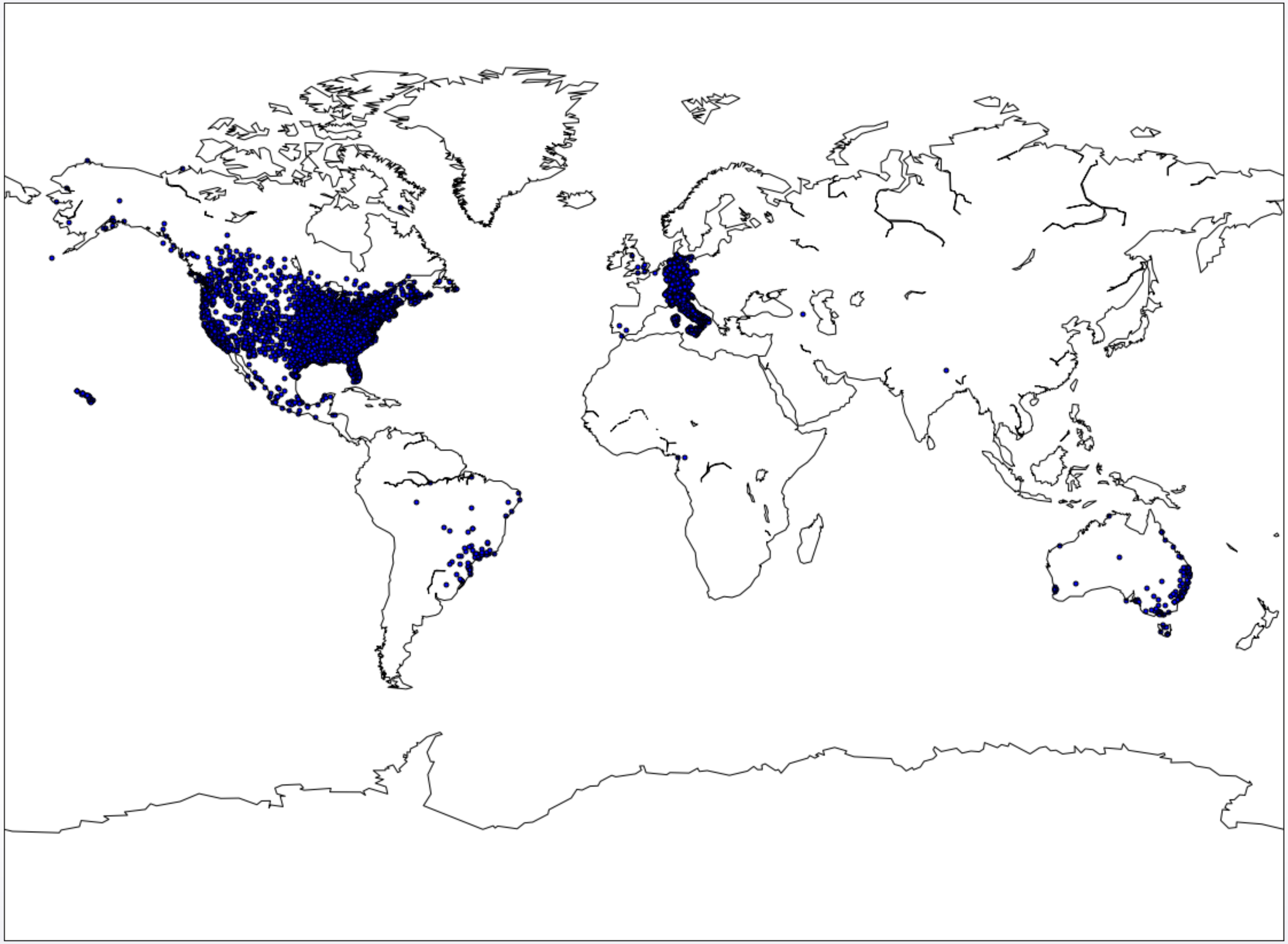
Hotel cities. Old version



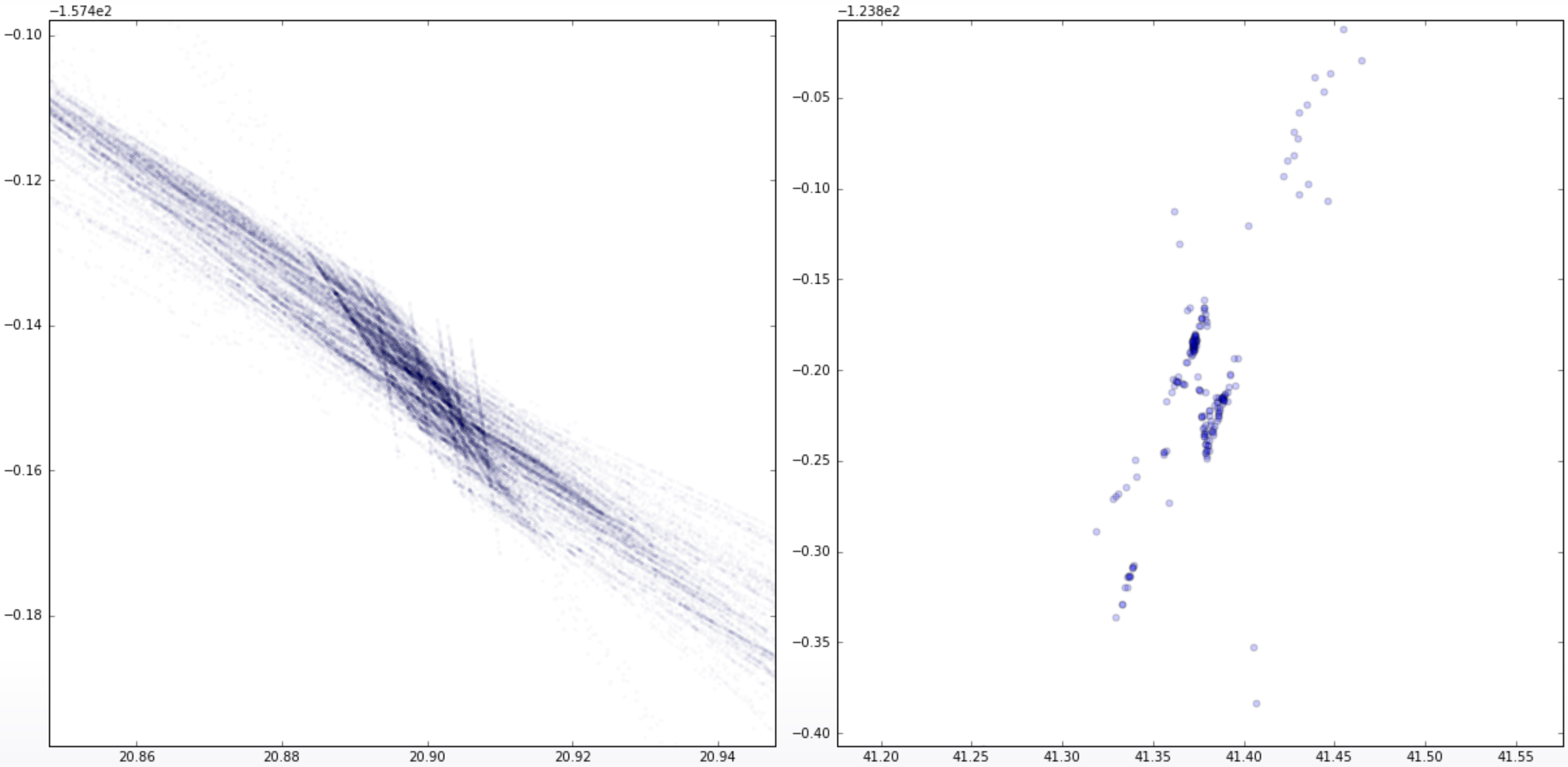
Hotels cities. New version



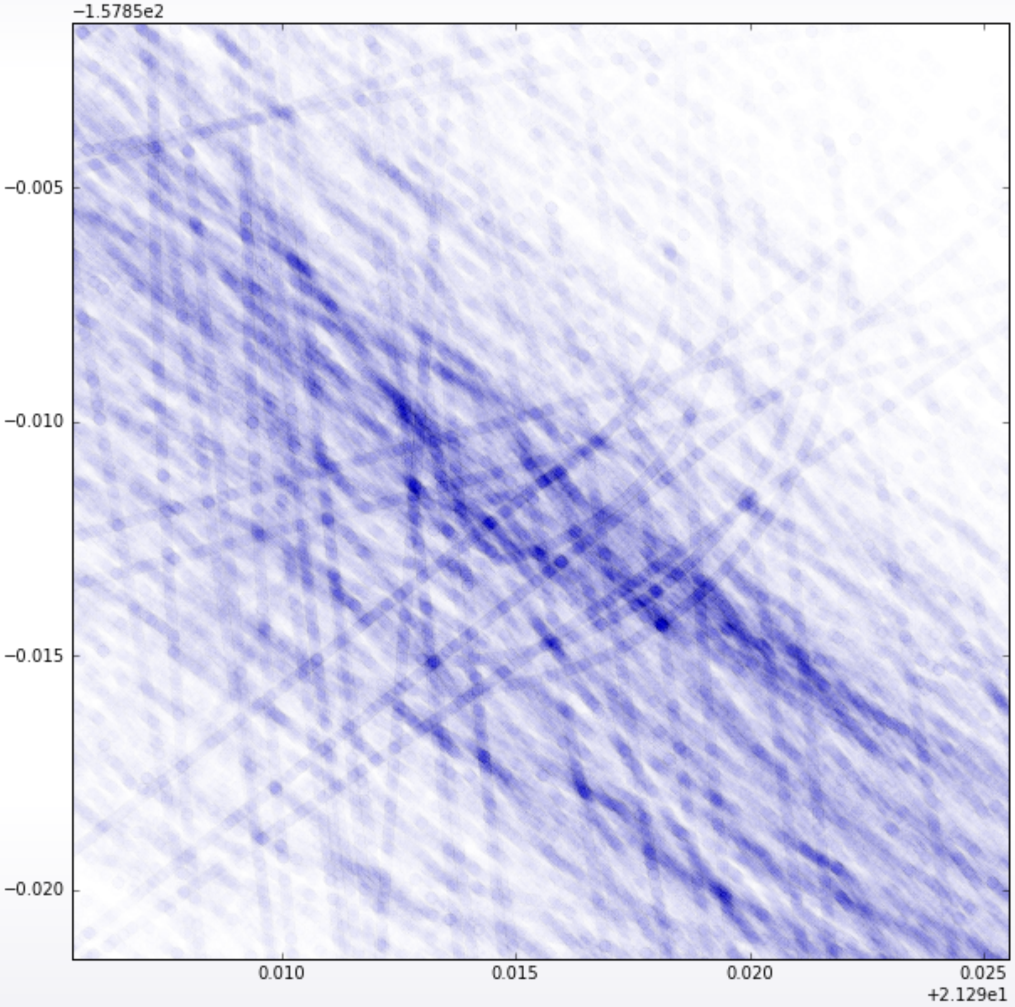
User cities. New version



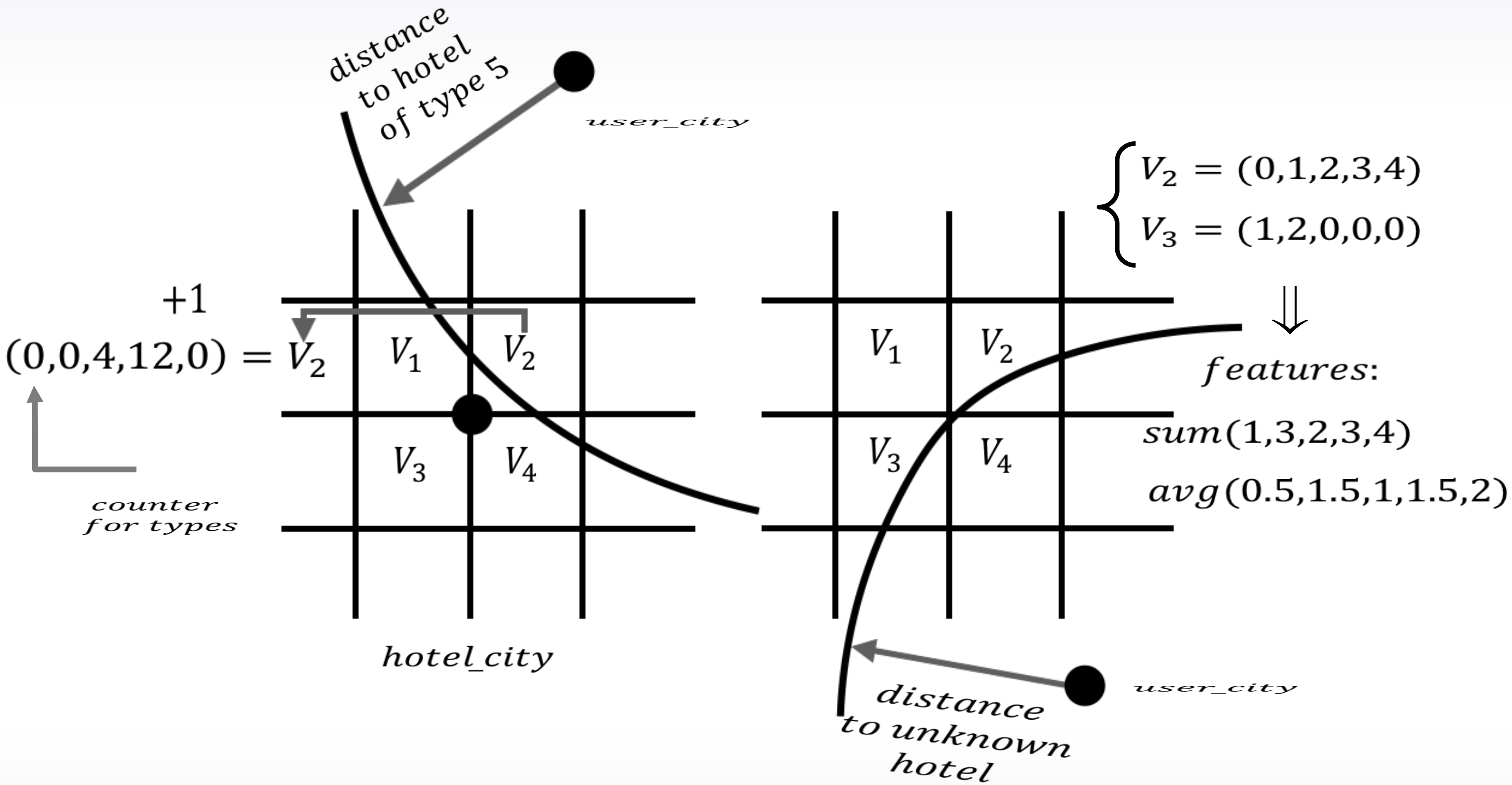
Trying to find the true coordinates of hotels (fail?)



Trying to find the true coordinates of hotels (fail?)



Counters in grid cells



For every city, let's create a grid around its center. Something like 10 kilometers times 10 kilometers with step size of 100 meters.

TRAIN

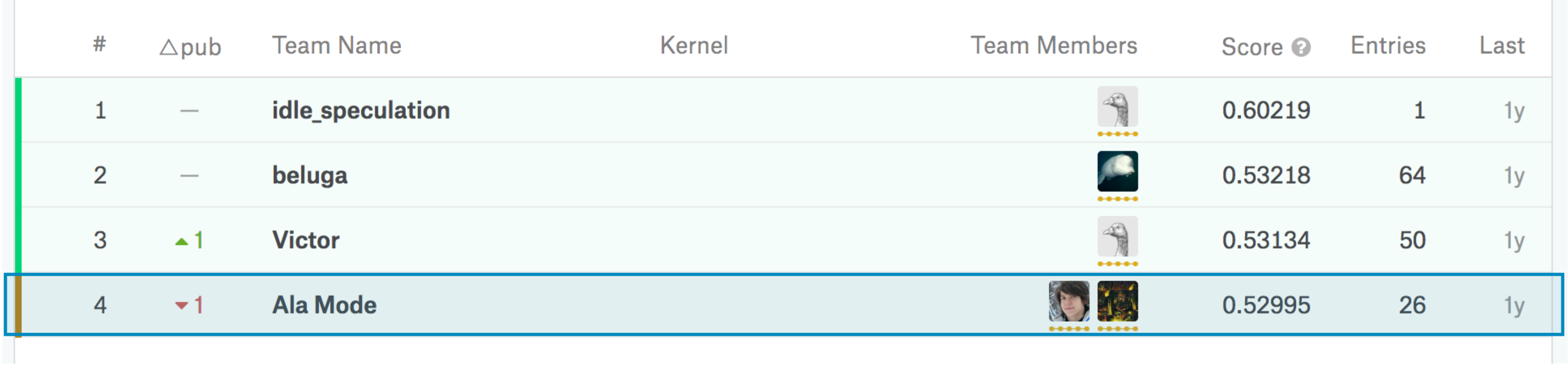
INFERENCE

Final model

- Out-of-fold feature generation. 2013<->2014
- Xgboost
- 16 hours of training

Results

- Public – 3rd
- Private - 4th



#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	idle_speculation			0.60219	1	1y
2	—	beluga			0.53218	64	1y
3	▲1	Victor			0.53134	50	1y
4	▼1	Ala Mode			0.52995	26	1y