POLITECNICO DI MILANO

Data Mining

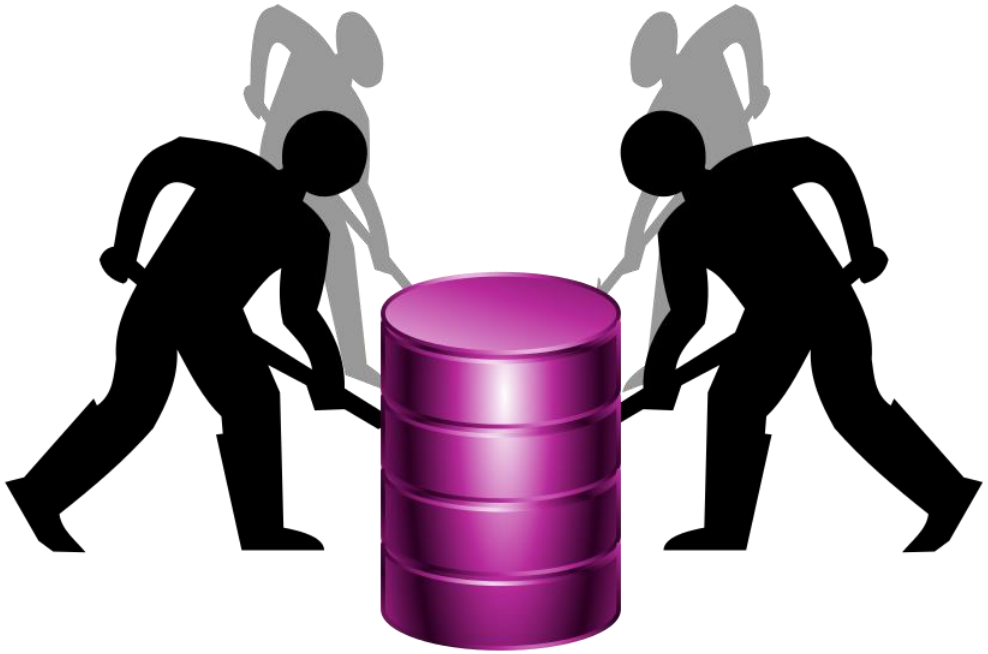
AY 2014-2015

Taxi Trajectory Prediction

Ettore Randazzo
Vittorio Selo
Paolo Simone
Benedetto Vitale

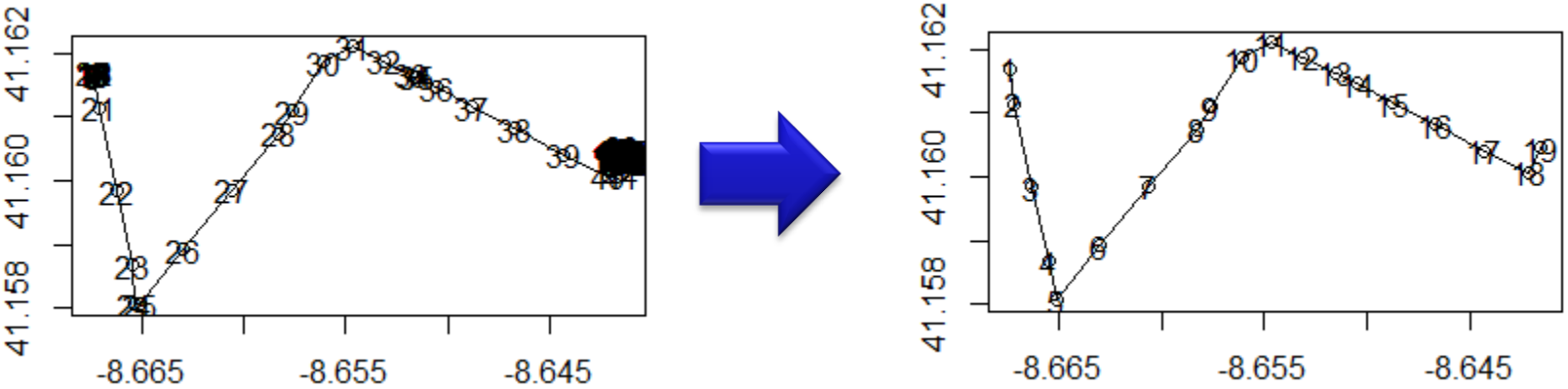
Summary

1. Preprocessing
2. Postprocessing
3. Prediction Models
4. Particular Cases

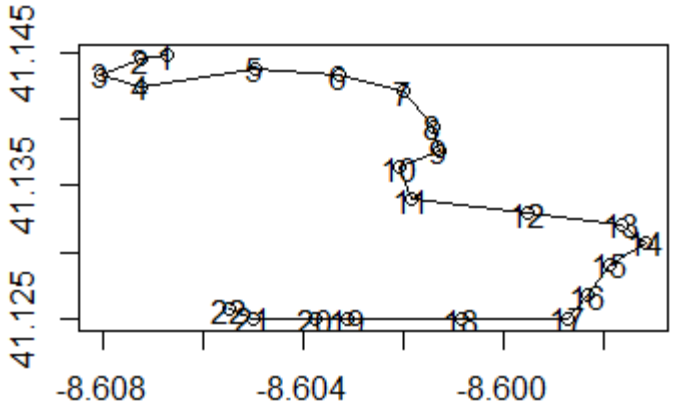


Preprocessing

- **Cleaning**: GPS errors & close points



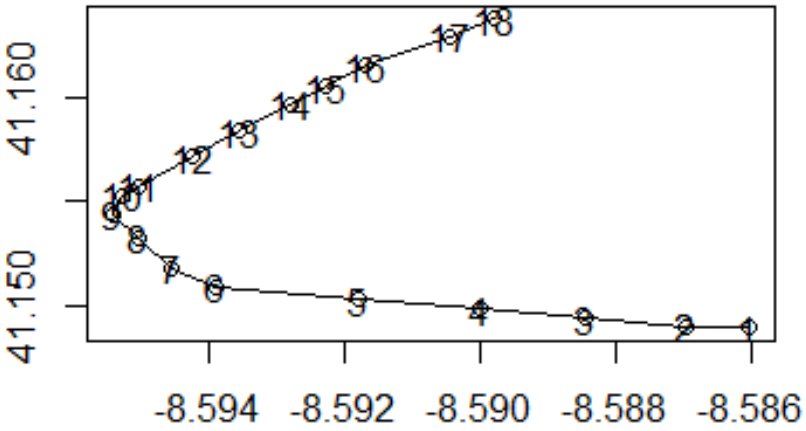
- **Disassemble**: obtain constant number of attributes

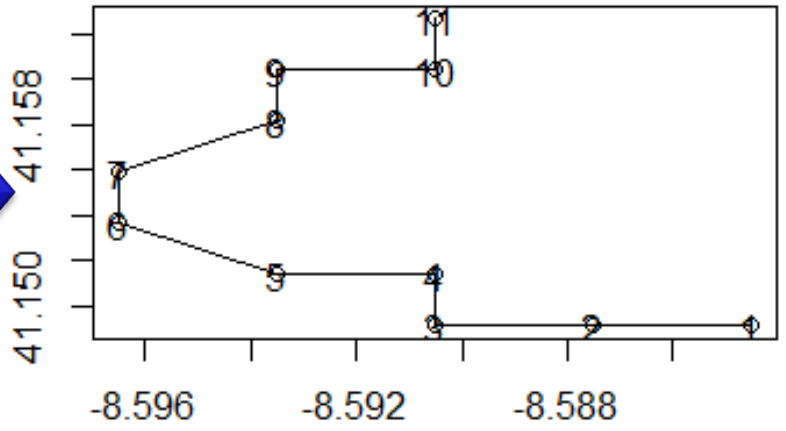


	LONG1	LAT1	LONG2	LAT2	LONG3	LAT3
1	-8.606709	41.14472	-8.607249	41.14444	-8.608041	41.14333
2	-8.607249	41.14245	-8.604954	41.14379	-8.603271	41.14327
3	-8.601993	41.14209	-8.601381	41.13948	-8.601282	41.13768
4	-8.602092	41.13640	-8.601840	41.13400	-8.599536	41.13295
5	-8.597637	41.13210	-8.597142	41.13075	-8.597862	41.12915
6	-8.598303	41.12677	-8.598717	41.12511	-8.600850	41.12500
7	-8.603118	41.12500	-8.603775	41.12501	-8.604990	41.12510

Preprocessing 2.0

- **Square clustering**: uniform input points





… also known as "**Snake preprocessing**" :-)

Postprocessing

- **Extract prediction**: destination predicted as factor to exploit correlation

- **Weightened Average**: last segments are more meaningful

Prediction models

First attempts:

- Classification trees (rpart)
- Regression trees (anova)
- K Nearest Neighbours (knn)
- Naive Byas (e1071)
- …

Until…

- Random Forest!

Except… we don't have 10^9 **TB** of RAM ☹

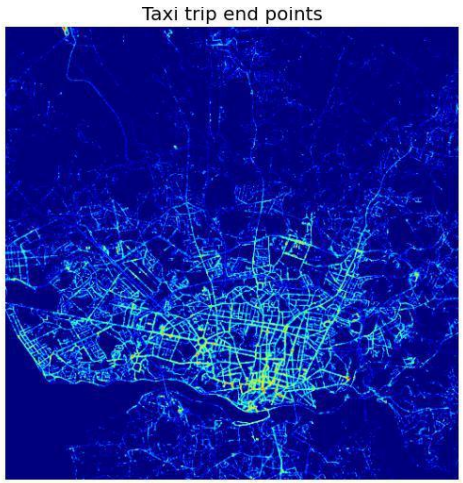
State of the art

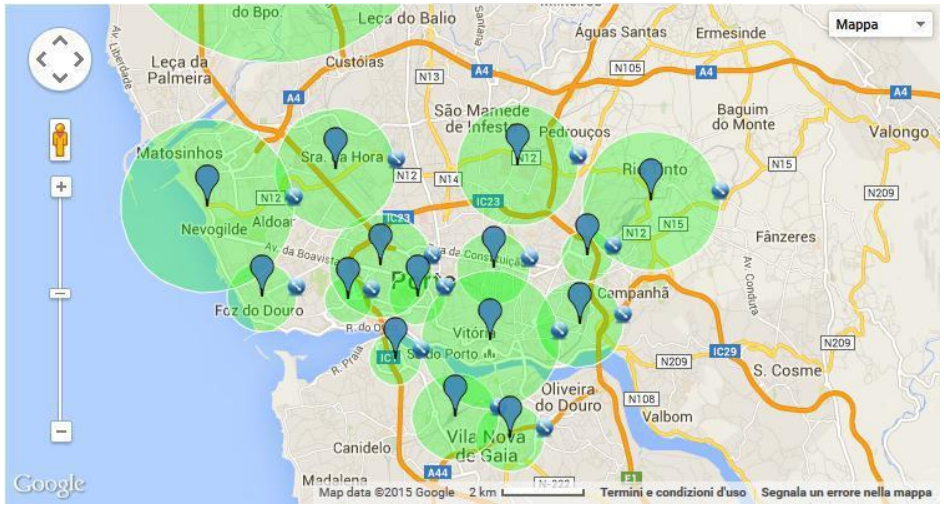
1. Generate **chunks** of small size by sampling the dataset
2. **Preprocess** each new training set
3. **Train** a different forest on each chunk:
   - Trip type
   - Day of the week
   - Day phase
   - Segment coordinate
   - Clusterized destination as factor (Classification)
4. Extract the **prediction** of each forest
5. Take the prediction that minimize the **SSE** wrt the others

Sneaky tricky trips

If the trip doesn't contain enough points to be processed…

- **Clustering**





- **Linear Interpolation**

Q&A