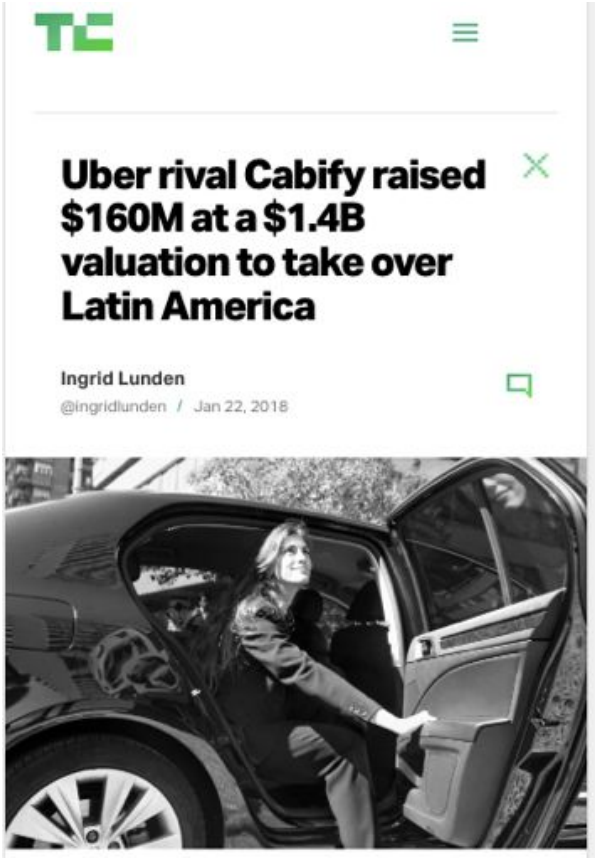
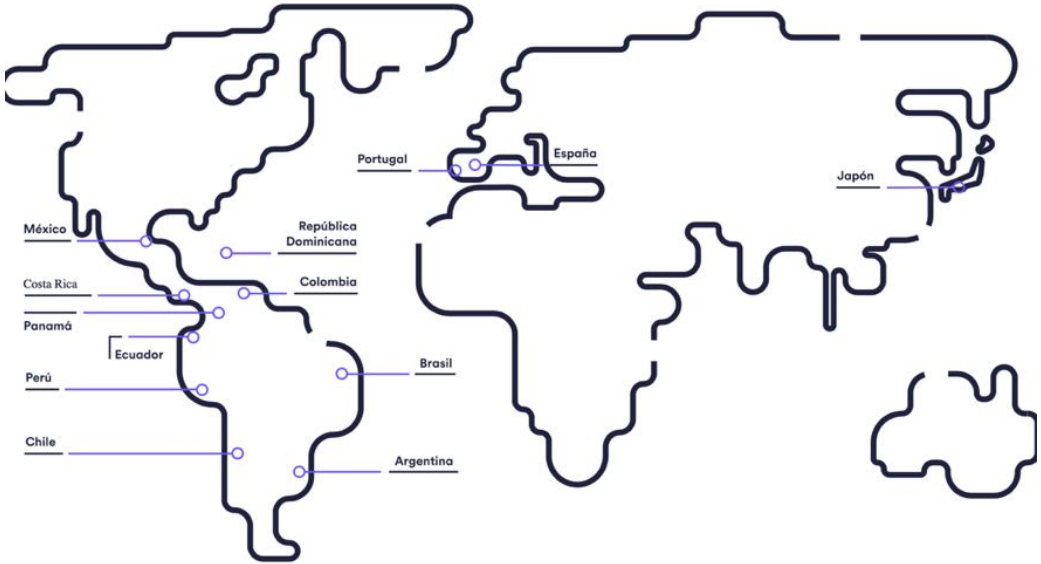
Driving in dataland

How data helps hypergrowth

cabify

About us (what everybody knows)

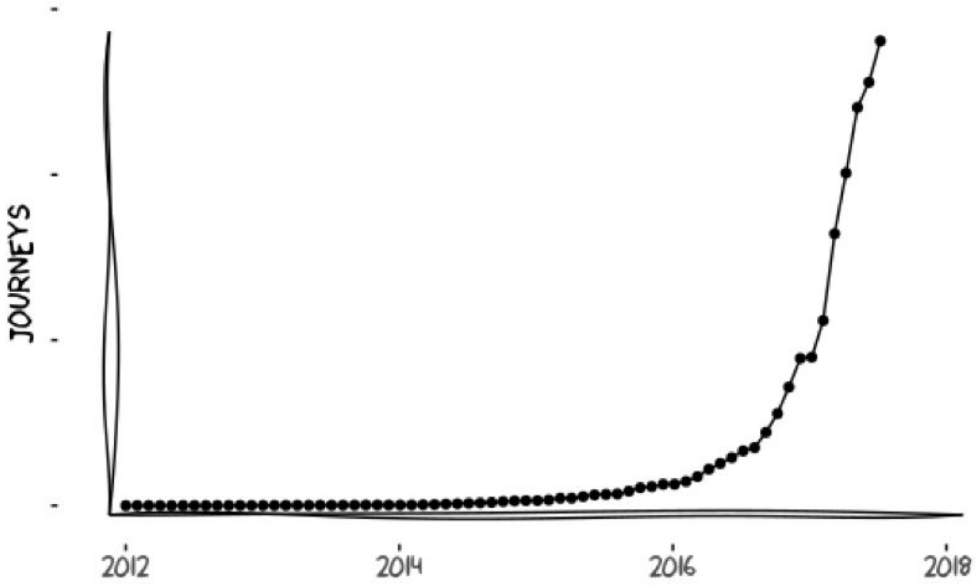




cabify

About us (what not everybody knows, but I can share with you without getting fired xD)

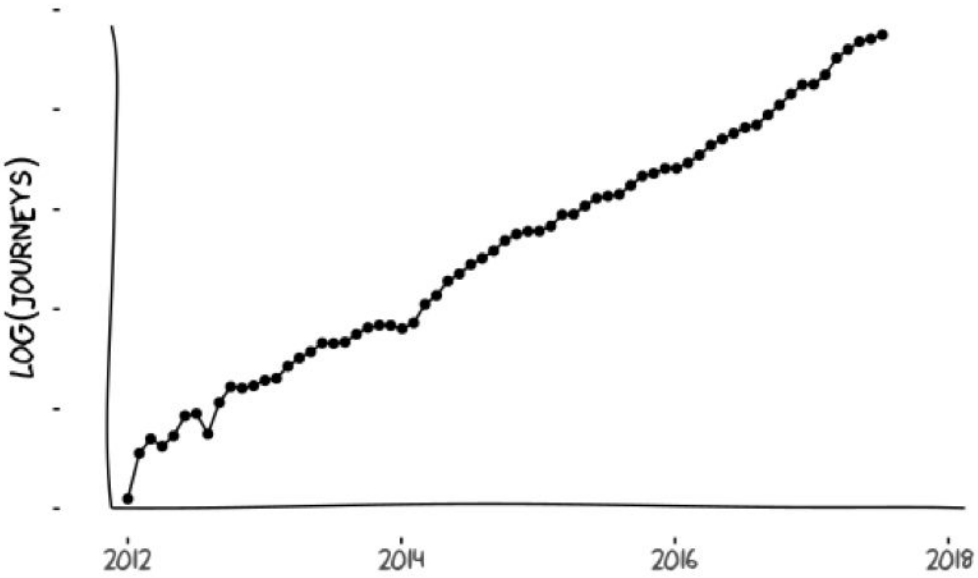
- We play in the most competitive industry nowadays (eg. in terms of VC funding). Being a unicorn, some of our competitors are 50x bigger
- All tech is built in MAD and SAO
- We grow exponentially, and we mean it
- We can understand cities through our own data



cabify

About us (what not everybody knows, but I can share with you without getting fired xD)

- We play in the most competitive industry nowadays (eg. in terms of VC funding). Being a unicorn, some of our competitors are 50x bigger
- All tech is built in MAD and SAO
- We grow exponentially, and we mean it
- We can understand cities through our own data



cabify


cabify

Non-mature industry = **Huge impact from *simple* data solutions**

cabify

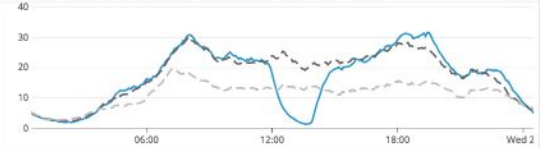
DATADOG

- Watchdog
- Events
- Dashboards
- Infrastructure
- Monitors
- Metrics
- Integrations
- APM
- Notebooks
- Logs

- Help
- Team
- carlos.herrera

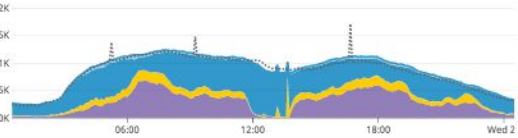
Drop off per minute (vs week before, month before)

1d



Fleet state

1d



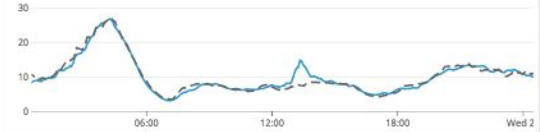
Connected drivers now

1d

338

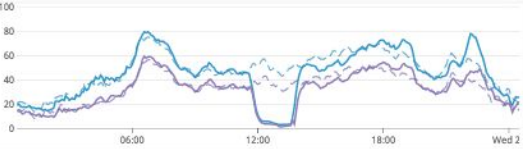
Median fleet pickup speed (km/h) vs previous week

1d



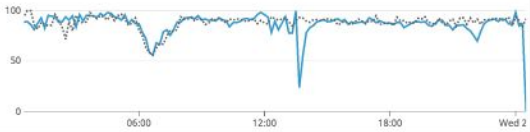
Load Factor & Effective Load Factor (%)

1d



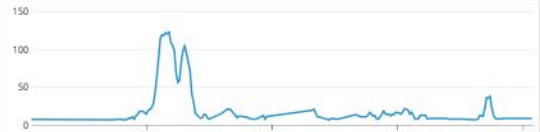
Success rate (vs previous week)

1d



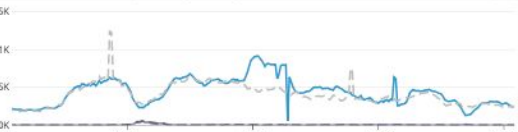
Max cell surge

1d



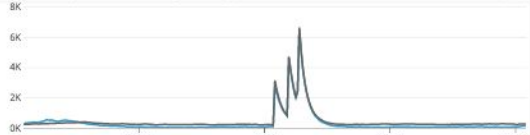
Available drivers vs queued journeys

1d



Timeouts per million (region vs global)

1d



Total drop off last 24 hours

1d

24417

Success rate

1d

86.3%

DO growth - last 7 days VS previous week

1w

7.1%

Total drop off 24h - 1 week ago

1d

25945

Success rate - week before

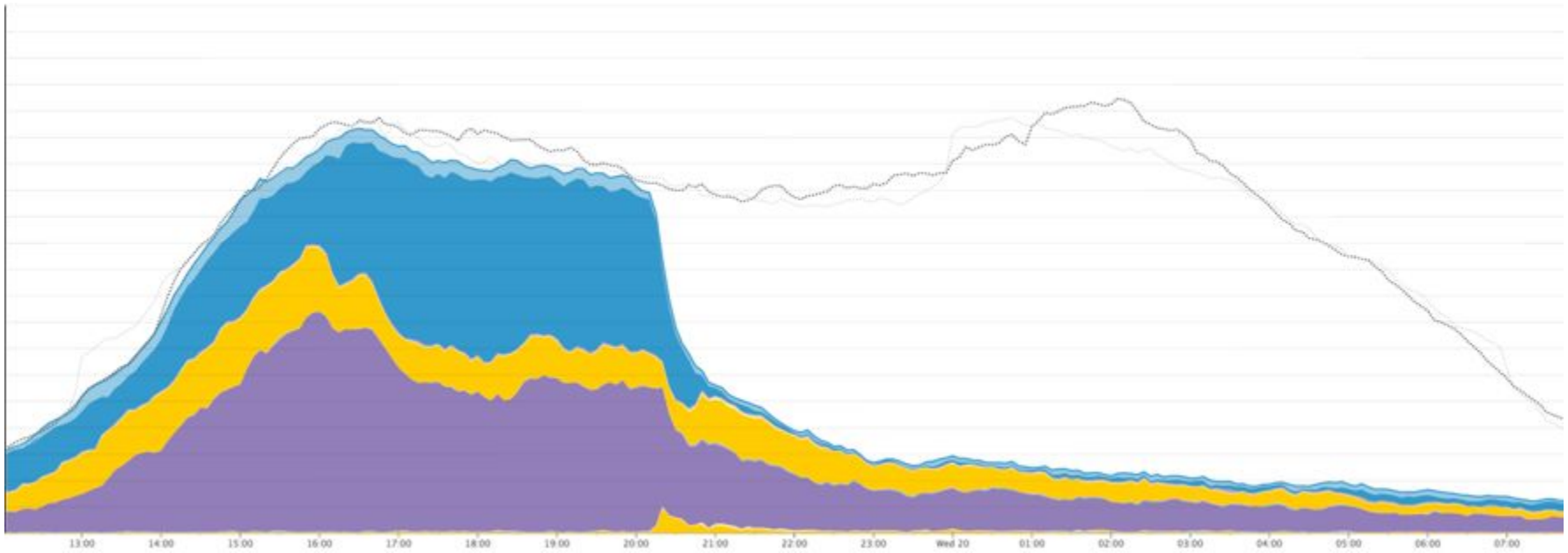
1d

87.2%

DO growth - Last 7 days VS 1 month ago

1w

88.5%



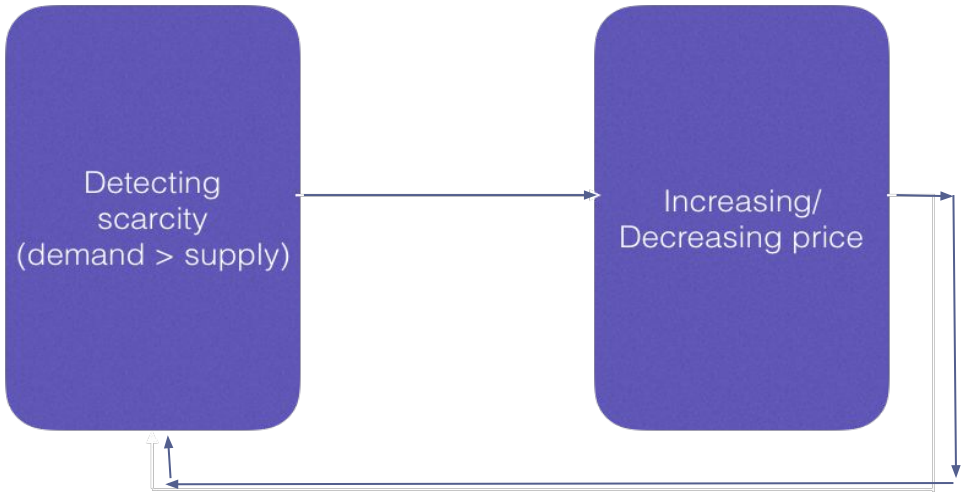
Mexico City, Sept 19 2017 (CEST)

cabify

Non-mature industry = **Huge impact from *simple* data solutions**

cabify

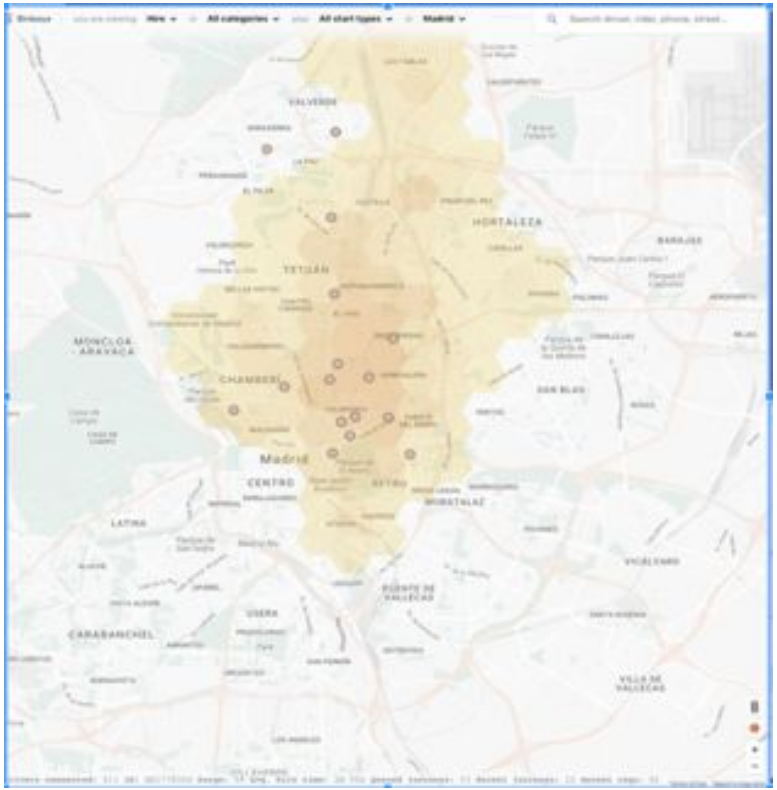
Exhibit A: Dynamic pricing



- Every 1 minute
  - We analyse up to 50K recent journeys, **specifically our service time**
  - Make a decision about 125K different hex cells
- We are simply nowcasting in a feedback loop, not forecasting at all
- We get a clear scarcity signal, temporally and spatially smooth

cabify

Exhibit A: Dynamic pricing

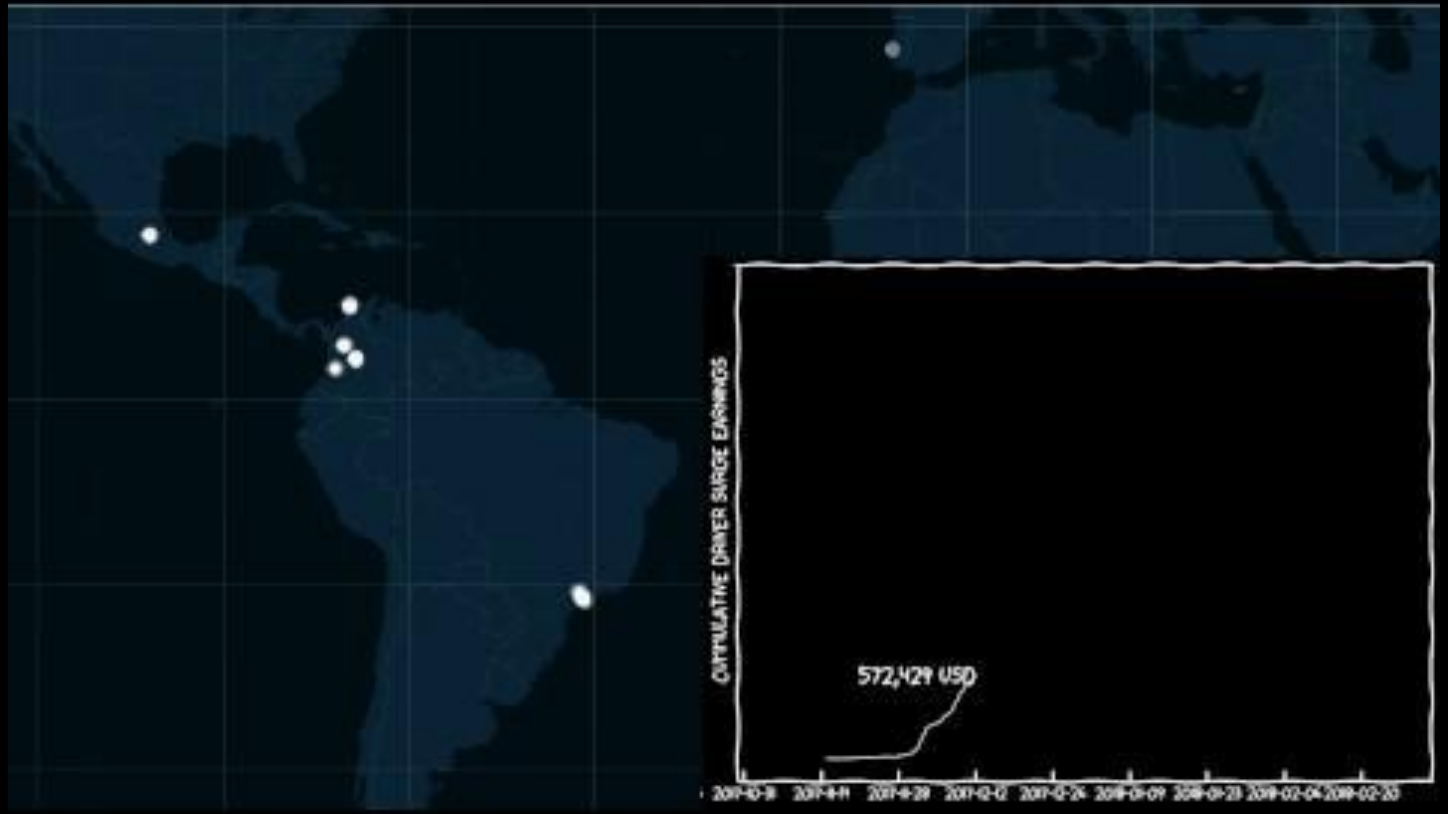


- Every 1 minute
  - We analyse up to 50K recent journeys, **specifically our service time**
  - Make a decision about 125K different hex cells

- We are simply nowcasting in a feedback loop, not forecasting at all

- We get a clear scarcity signal, temporally and spatially smooth

cabify



cabify

Impact of dynamic pricing

Aim: generating value for our drivers. Increase by 15-20% their earnings during peak hours

Methodology: if a driver makes a journey with demand supplement 20%, then is like 0.2 journeys additional journeys happened at Cabby

- First week in full deploy generated value equivalent to having done over 120K additional journeys.

- To do so, our drivers would typically need to drive for 2.8 million km and 100K hours

cabify

Non-mature industry = **Huge impact from *simple* data solutions**

cabify

Exhibit B: Matching system

- It is the main tech advantage ride hailing (RH) system brings to the world

- A taxi driver needs to match in time and space with perspective rider => they are busy only 30% of their time

- A RH driver “is found by the job” => they are busy up to 55% of their time

cabify

Before data team

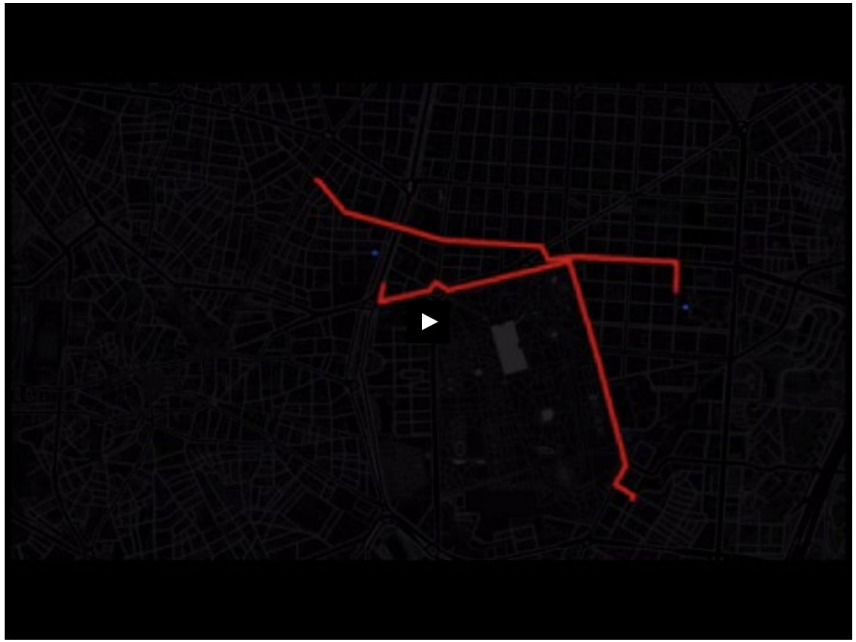
- Matcher assigns the ride to the nearest driver (bird distance)
- Ride is not assigned if nearest driver further than X km straight line (X being manually set up by zone, time, kind of vehicle...)
- FCFS basis (greedy approach) => wild goose chase under heavy load :-(

After data team

- Matcher assigns the ride to the driver that will arrive faster (ETA estimation)
- Ride is not assigned if driver would take longer than Y minutes (isochorone, automatic)
- Solve the entire city at a time (hungarian algorithm AKA munkres)

cabify

Before data team

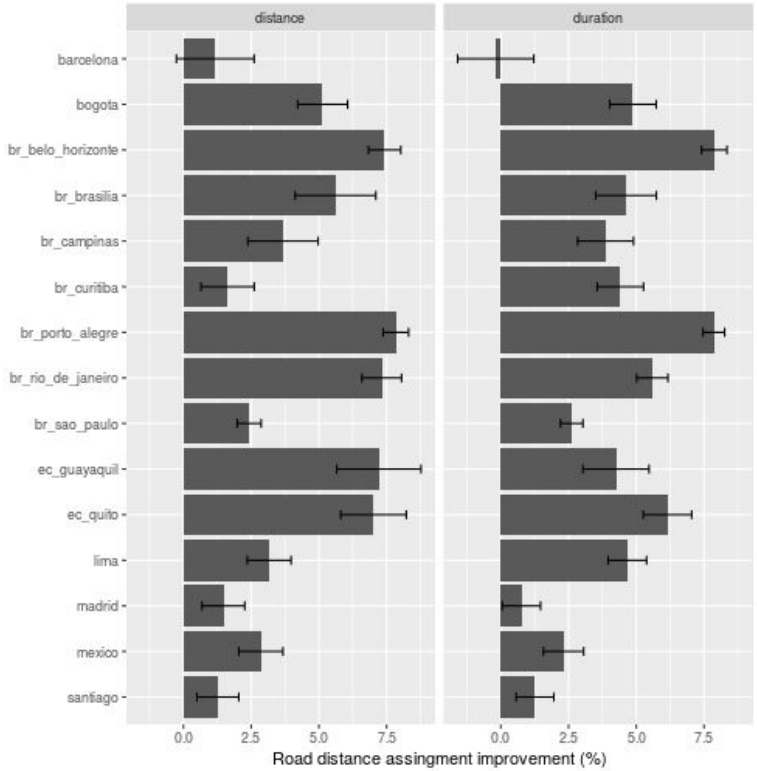


After data team

cabify

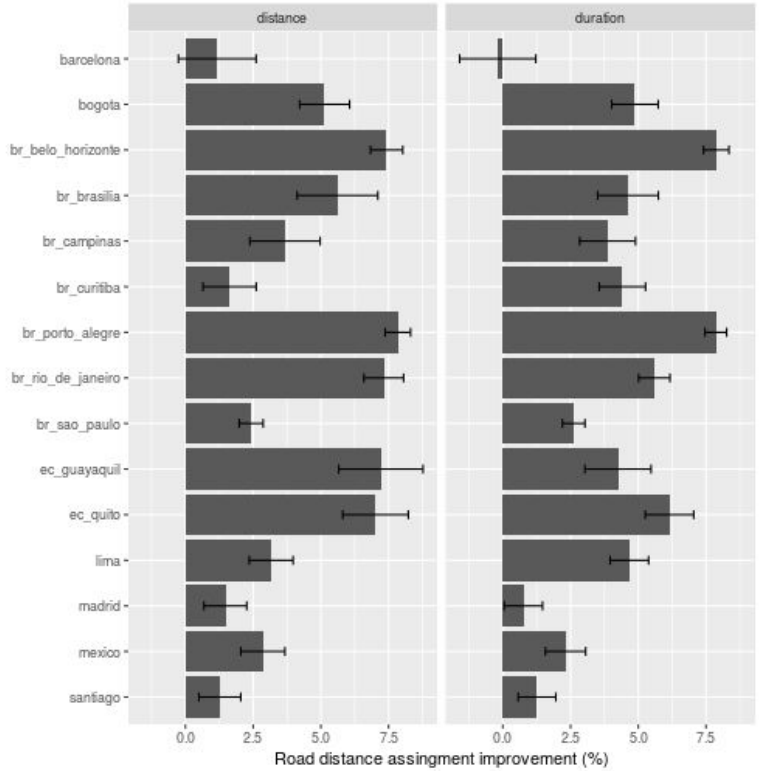
Matching system 2.0 impact

- Measured in a 13 min + 13 min experimental configuration to control for seasonality, localization...
- Our pickups are now shorter and faster
- We estimate our drivers are **25K-30K saving working hours and 500,000 km per week**



Challenge: scaling up matching 2.0

- We are a gold mine for commercial route providers
- We request up to 2-3K optimal routes per second, scaling up with square of business size (CFO not happy at all)
- Commercial results are not exactly what we are looking for
- Can we build our own ETA estimation system?



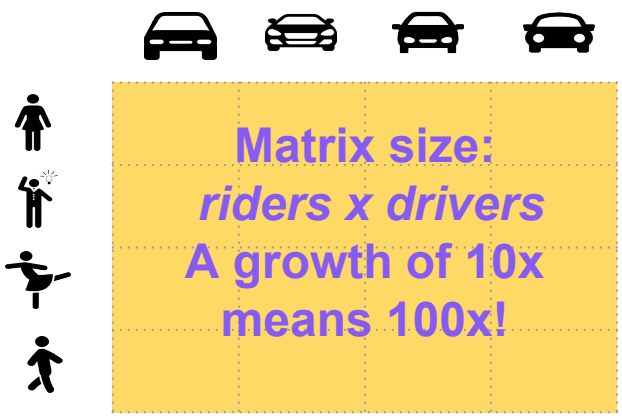
ETA estimations

cabify

The problem

High dependency on Commercial API ETA predictions:

- Show a price to the rider
- Match drivers with riders
- Simulations and experiments



cabify

The objective

- Make predictions about the time required for going from A to B based on historical **Data**
- Assume that the *route* of the trip will not be available in *evaluation time*
- Distinguish between going *to pickup* and to *destination* states

2 minutes

cabify

The data

- Journeys from Jan 2017 to May 2018
- For Madrid 11M of examples, randomly chosen.
- Extended data set with more than 1 billion of examples (some tests)

from_lat	from_lon	to_lat	to_lon	start_at	end_at	state	time_diff
40.467	-3.58071	40.468	-3.57002	2018-04-17 06:33:27	2018-04-17 06:37:22	hired	234
40.447	-3.69163	40.4376	-3.69354	2018-05-08 17:58:13	2018-05-08 18:04:42	hired	389
40.4509	-3.60403	40.4913	-3.59458	2018-04-08 12:50:17	2018-04-08 12:59:34	hired	556
40.4363	-3.69125	40.4333	-3.69149	2018-05-14 20:50:21	2018-05-14 20:53:55	hired	214
40.4314	-3.67514	40.4326	-3.68163	2018-05-08 16:08:55	2018-05-08 16:13:53	hired	298
40.4668	-3.66895	40.4693	-3.64337	2018-05-11 13:23:35	2018-05-11 13:32:36	hired	541
40.436	-3.68327	40.4274	-3.68488	2018-05-09 07:54:18	2018-05-09 08:09:02	hired	884
40.4525	-3.66802	40.4511	-3.66736	2018-05-09 07:39:38	2018-05-09 07:48:42	hired	544
40.4673	-3.7172	40.4606	-3.70944	2018-05-04 07:10:51	2018-05-04 07:15:20	hired	269
40.4272	-3.68401	40.4289	-3.68581	2018-05-11 16:49:06	2018-05-11 16:53:09	hired	242

The basic features

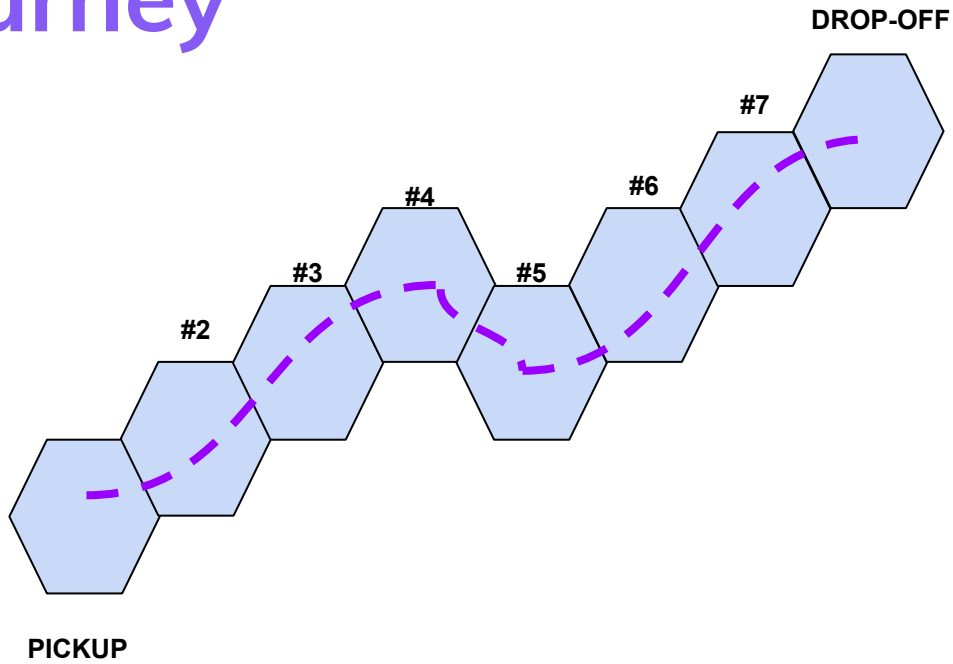
- Distinguish between going *to pickup* or *to destination*
- Add some heuristics like the *haversine* distance and the *manhattan* distance
- The model performs pretty bad

state	haversine	manhattan_lon	manhattan_lat	manhattan	day_hour	week_day	time_diff
hired	911.275	111.589	904.41	1016	06	Tuesday	234
hired	1059.73	1047.36	161.466	1208.83	17	Tuesday	389
hired	4553.73	4482.92	799.686	5282.6	12	Sunday	556
hired	334.093	333.479	20.2478	353.727	20	Monday	214
hired	565.473	133.625	549.453	683.078	16	Tuesday	298
hired	2183.38	287.741	2164.29	2452.03	13	Friday	541
hired	970.054	960.392	136.577	1096.97	07	Wednesday	884
hired	164.925	155.318	55.4662	210.784	07	Wednesday	544
hired	998.416	752.379	656.356	1408.73	07	Friday	269
hired	241.482	187.051	152.724	339.776	16	Friday	242

cabify

The poetry of a journey

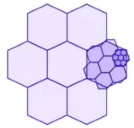
- For one journey, we do not only know information about pick up and drop of



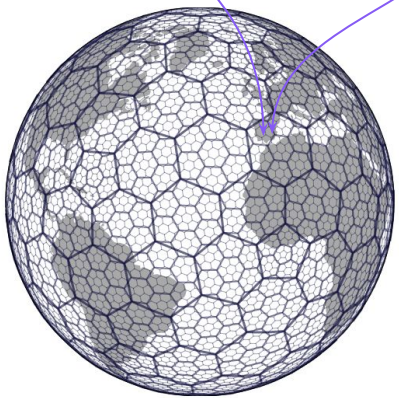
[pickup #2 #3 #4 #5 #6 #7 drop-off]

cabify

Enhanced data



from_lat	from_lon	to_lat	to_lon	from_h8	from_h9	from_h10	to_h8	to_h9
40.467	-3.58071	40.468	-3.57002	88390cb539fffff	89390cb5387ffff	8a390cb53847fff	88390cb517fffff	89390cb516bffff
40.447	-3.69163	40.4376	-3.69354	88390cb19bfffff	89390cb19bbffff	8a390cb19b97fff	88390cb191fffff	89390cb190bffff
40.4509	-3.60403	40.4913	-3.59458	88390cb527fffff	89390cb5277ffff	8a390cb52767fff	88390cb555fffff	89390cb5547ffff
40.4363	-3.69125	40.4333	-3.69149	88390cb197fffff	89390cb1973ffff	8a390cb19707fff	88390cb197fffff	89390cb197bffff
40.4314	-3.67514	40.4326	-3.68163	88390ca269fffff	89390ca268fffff	8a390ca268efff	88390ca269fffff	89390ca269bffff
40.4668	-3.66895	40.4693	-3.64337	88390cb0bdfffff	89390cb0bcbffff	8a390cb0bca7fff	88390cb0b3fffff	89390cb0b37ffff
40.436	-3.68327	40.4274	-3.68488	88390cb193fffff	89390cb192bffff	8a390cb1929fff	88390ca26dfffff	89390ca26c3ffff
40.4525	-3.66802	40.4511	-3.66736	88390ca249fffff	89390ca248fffff	8a390ca248f7fff	88390ca249fffff	89390ca248fffff
40.4673	-3.7172	40.4606	-3.70944	88390cb1d9fffff	89390cb1d83ffff	8a390cb1d817fff	88390cb1d1fffff	89390cb1d07ffff
40.4272	-3.68401	40.4289	-3.68581	88390ca26dfffff	89390ca26c3ffff	8a390ca26c2ffff	88390ca26dfffff	89390ca26d3ffff



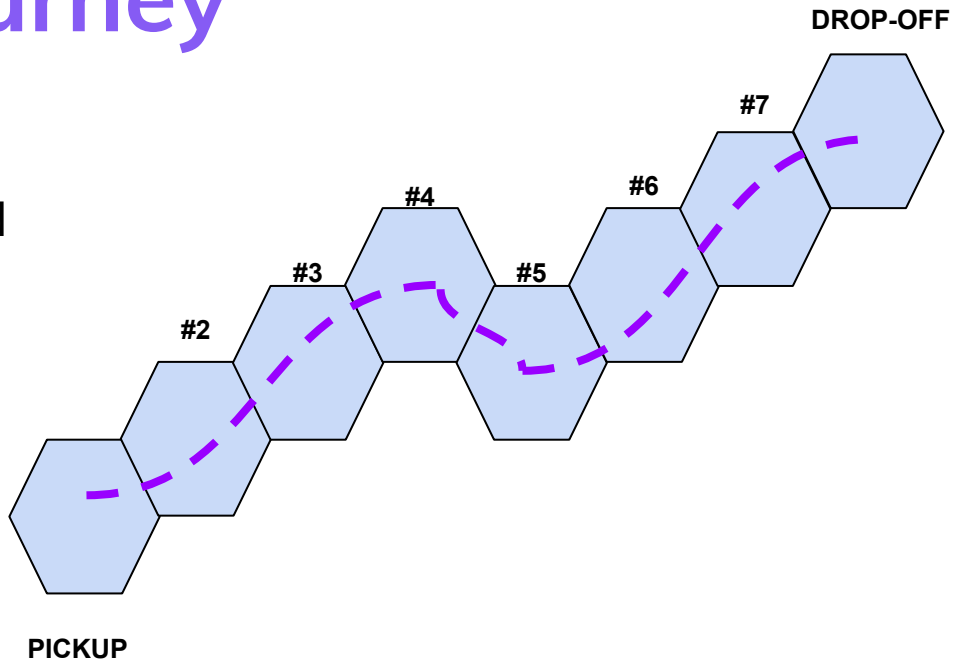
- Huge number of hex combinations
- One-hot encoded: one feature for each hex of a level, resulting in orders of millions of features
- For the algorithm, each hex is exactly the same as the others

cabify

The poetry of a journey

- For one journey, we do not only know information about pick up and drop of

- The magic: cells in journeys behave statistically similar to words in sentences
  - Zipf law: Common words (Why), common cells (train station)
  - Co-occurrence: Madrid is often close to city, just like cells in a road are often traversed together
  - Do NLP techniques such as **word2vec** have a chance?

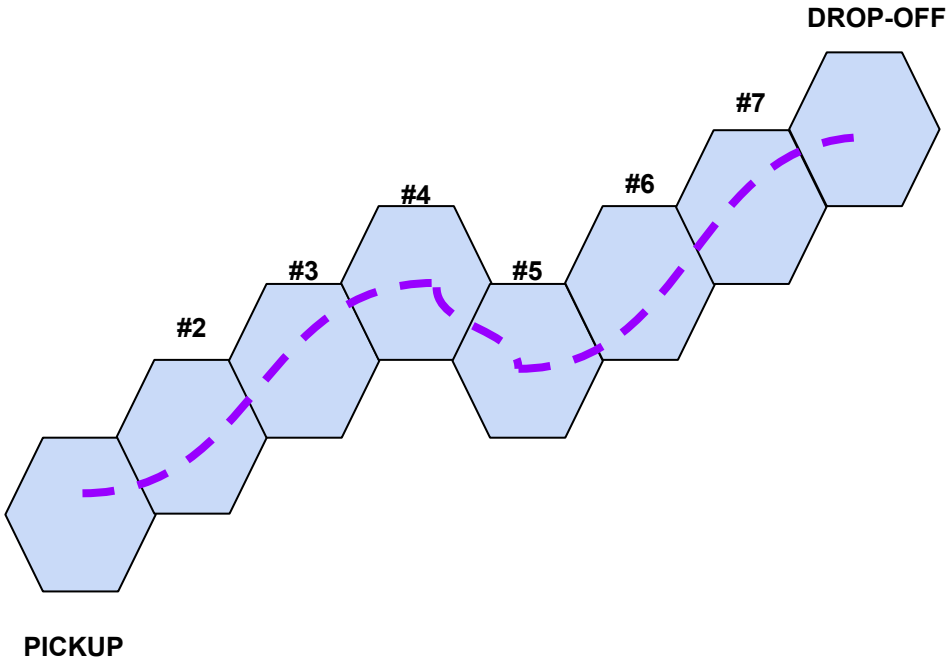


[pickup #2 #3 #4 #5 #6 #7 drop-off]

cabify

Embeddings

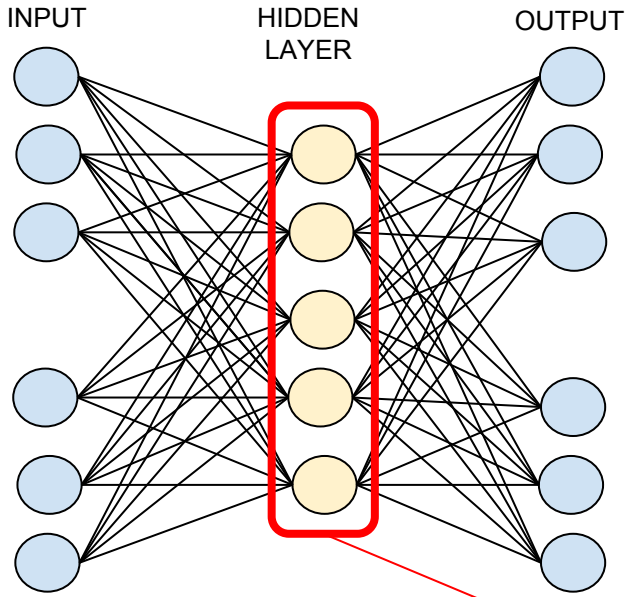
- Use the data of the trajectory of the journey: available in *training time* but not in *evaluation time*
- Consider each trajectory as a word of h3 of a concrete level
- If you can go for con hex to another fast, their embeddings should be close (in euclidean distance)



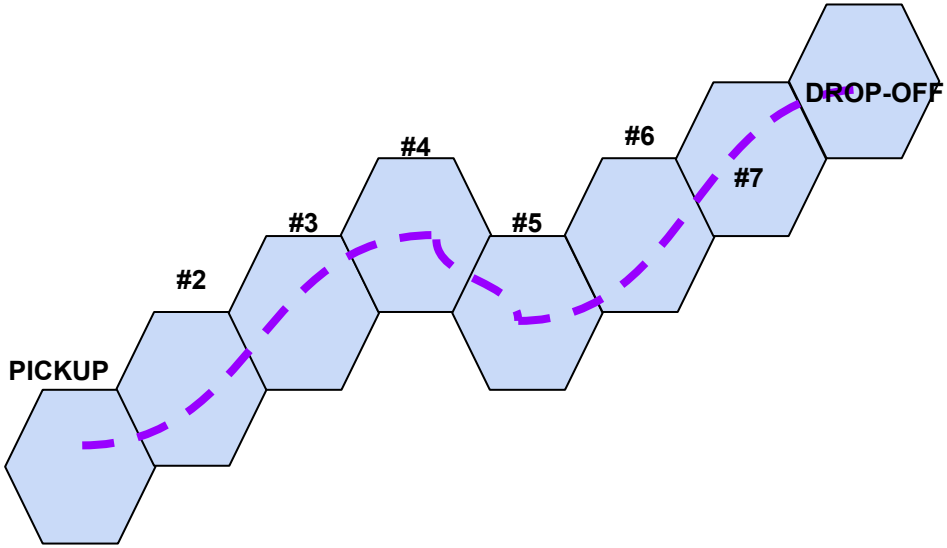
[pickup #2 #3 #4 #5 #6 #7 drop-off]

cabify

Embeddings



#PICK-UP	#2
#PICK-UP	#3
#PICK-UP	#4
#2	#3
#2	#4



[pickup #2 #3 #4 #5 #6 #7 drop-off]

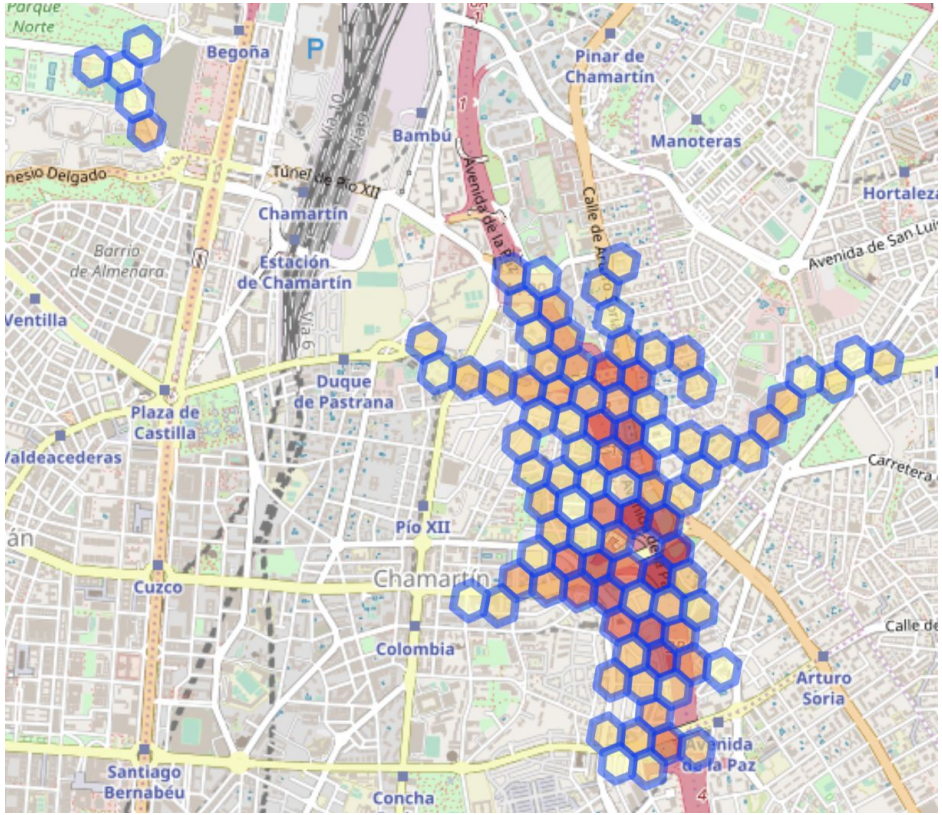
Encoded representation of h3 cells based on journeys. Dimensionality reduction from millions to a few hundred

cabify

Embeddings

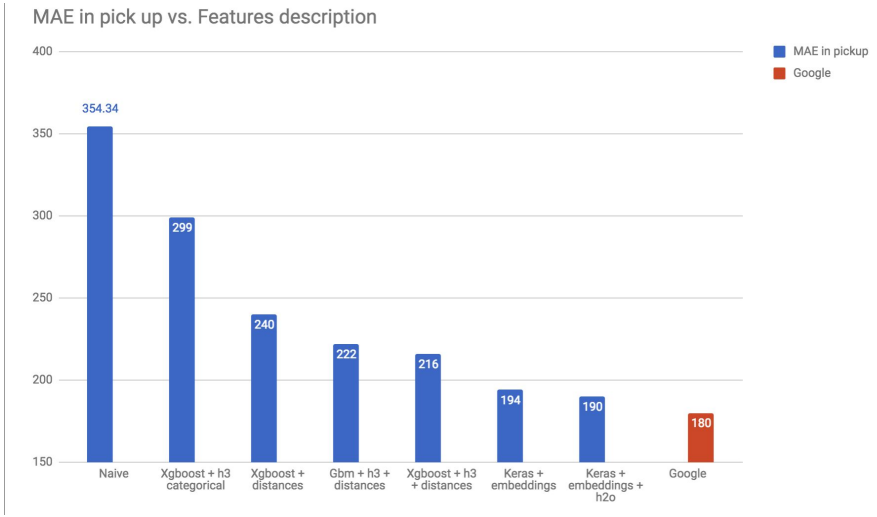
Are able to understand the geometry of the city:

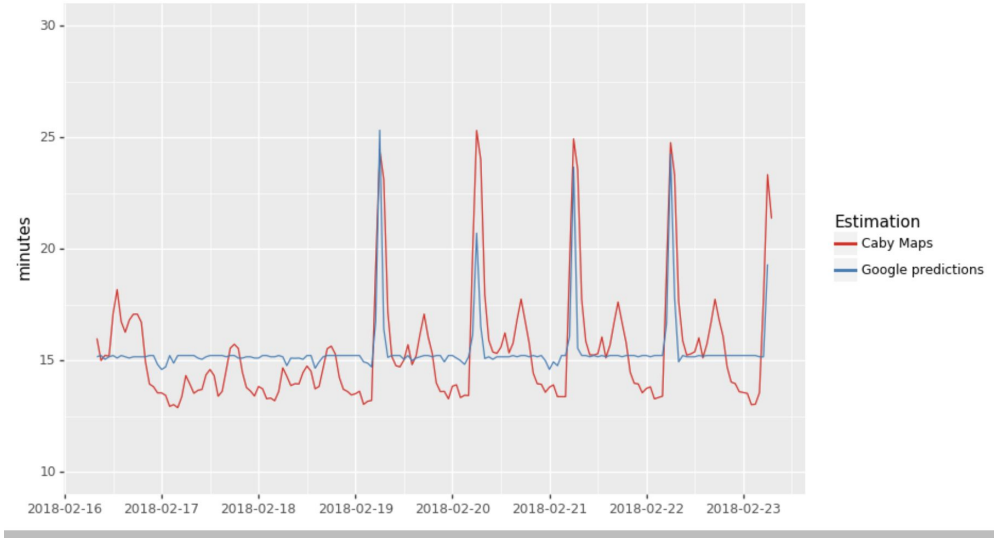
- Close areas of the city in terms of duration
- Principal roads used by the vehicles
- Directions



cabify

The results

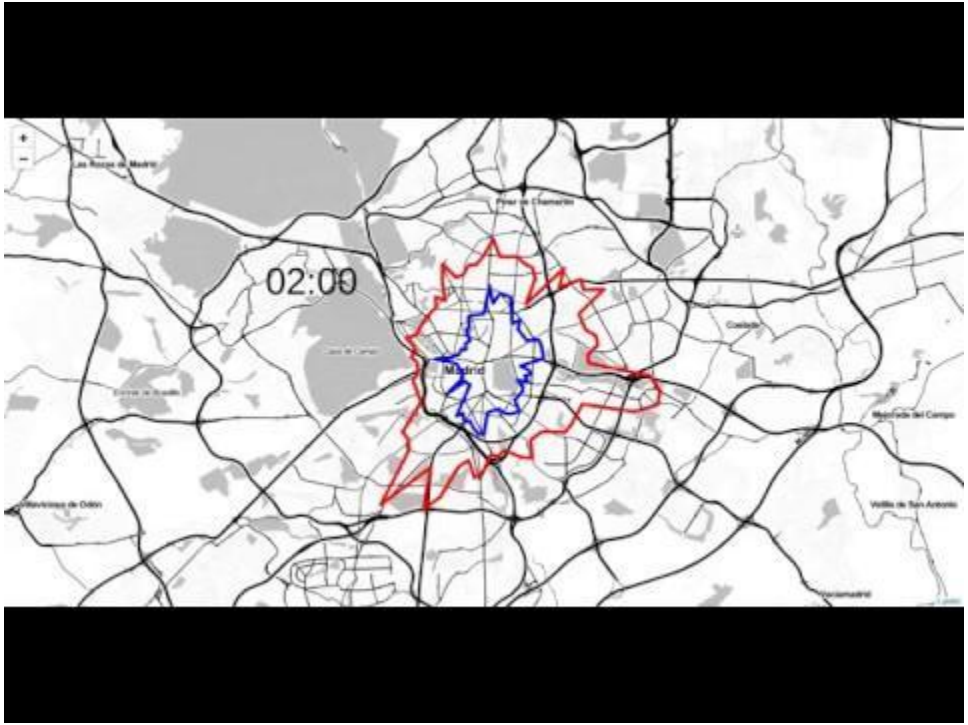




cabify

Isochrones

- All the points that can be reached from point A in *x* minutes
- Scales linearly with the number of drivers
- Commercial APIs does no provide this!



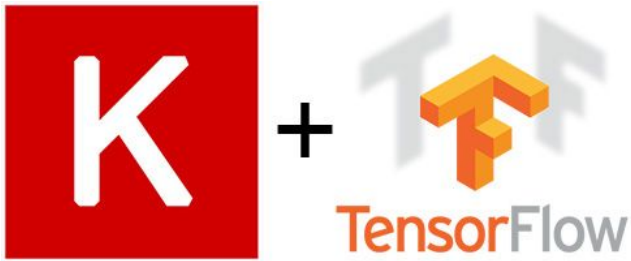
cabify

Soup of logos

- Spark + Scala for data manipulation (in big cluster!)
- H2o.ai and H2o Driverless
- Keras and tensorflow
- 2 x Nvidia P100
- Pub/sub for real time production

Cloud Pub/Sub

Real-time *and* reliable messaging with Pub/Sub



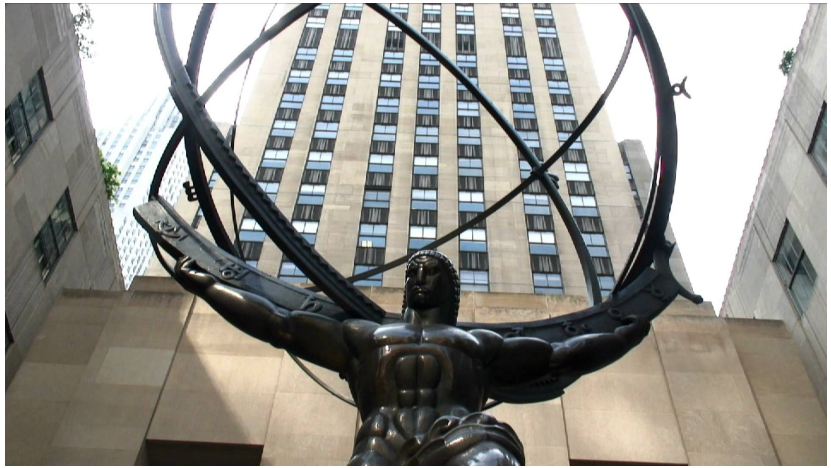
cabify

Production specs

- SLA 99.9%
- Up to 100K req/sec
- Accuracy over < 90%
- Auto-retraining from up to 1K journeys per minute

Why are we here?

- 4 Data Engineers + 7 DBAs
- 8 Data Scientists
- 15 Data Analysts + 10s of biz analysts
- 700+ dashboard daily active users
- Yes, we have a tech center in 🇧🇷



Where are we headed for Q2-2019

- x3 Data Engineers
- x2 Data Scientists

cabify

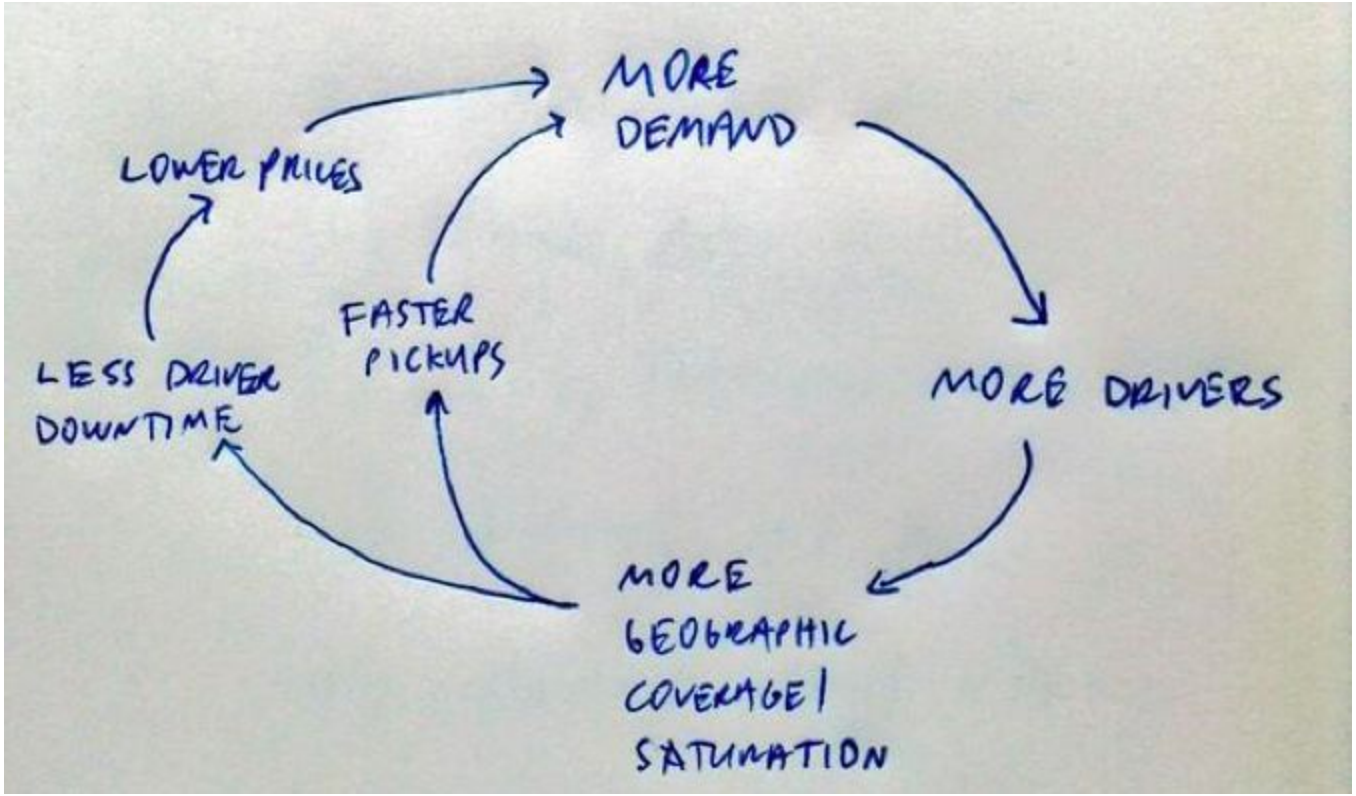
Thanks! BTW, we are hiring!

- carlos.herrera@cabify.com => this is me!!!
- alberto.gonzalezcalero@cabify.com => Head of Data Engineering

If you have not used Cabify, today we are having a FREE DAY in Barcelona xD

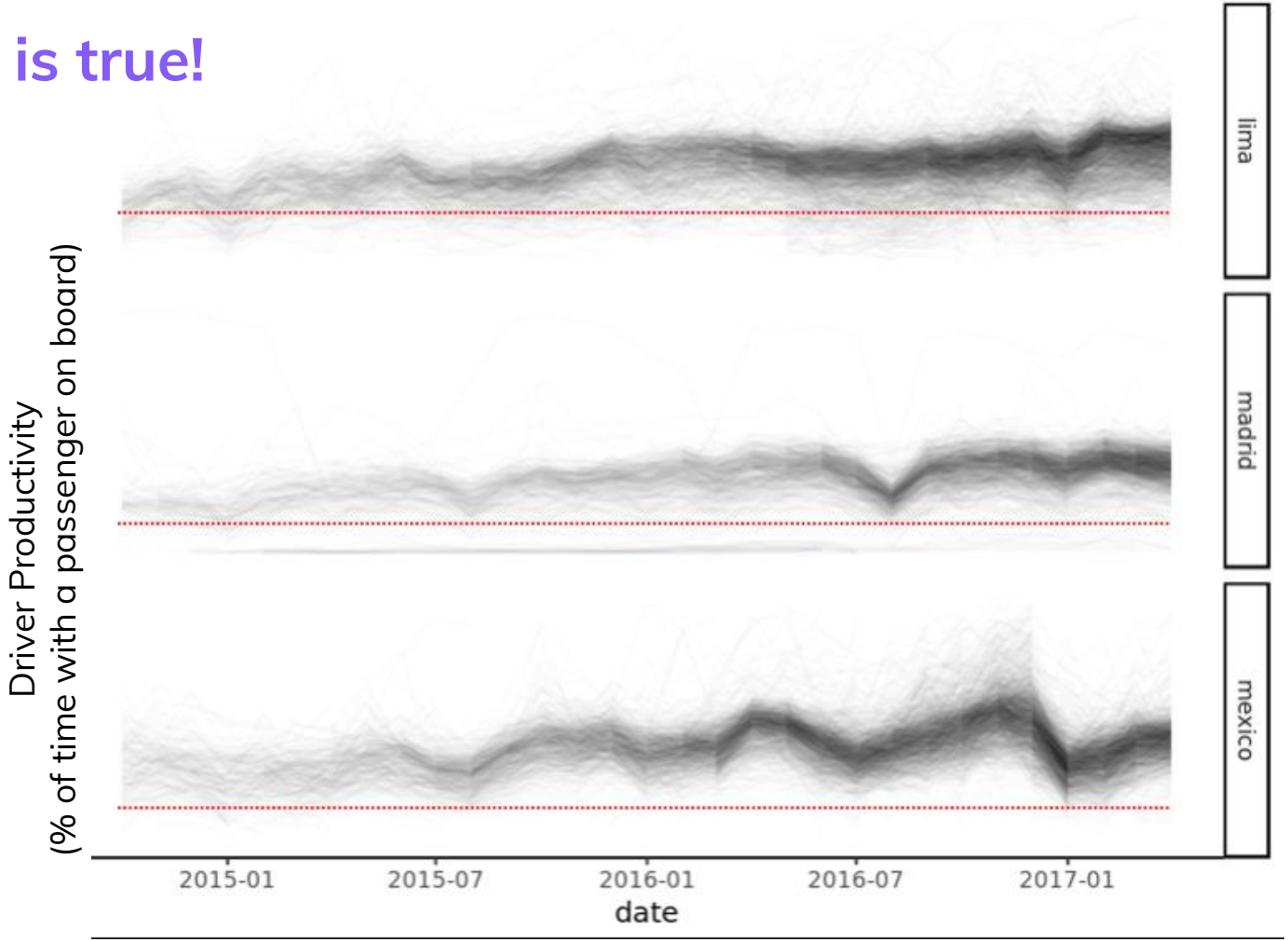
cabify

Geographic density is the new network effect



cabify

And it is true!



cabify