CARSHARING ANALYTICS LAB

Prof. Marco Mellia

AKA – a Big Data approach

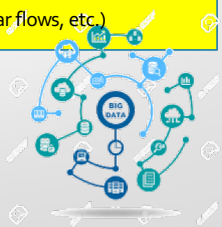
SCENARIO

- Car sharing platforms are becoming popular
  - Only in Torino: Car2Go, Enjoy, ~~CarCityClub~~, BlueTorino
- They offer a "click-and-rent" paradigm
  - Using an APP, or the web, users can check which cars are available, book one, then start the rental
- Platforms offer web API to interact with the system
  - Which cars are available, status of car, gas, etc.
  - Some API are public -- https://github.com/car2go/openAPI
- **Idea: use these platforms as source of information for mobility studies**
  - Lab focuses on this last aspect

CARSHARING ANALYTICS LAB

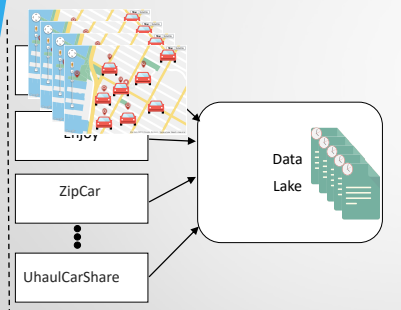
OVERALL PLATFORM

- Big Data approach
  - **Data Collection**: Collect data from Open Access Repositories
    - All car position in all Italian cities
    - Integration with maps information, public transportation, traffic status
  - **Data management and analysis**: Create a big data platform to harvest the raw data
    - Integrate data, and compute higher level metrics (car density, car flows, etc.)
  - **Design a graphical interface to visualize data**
    - Time series
    - Coloured maps
    - Dashboards



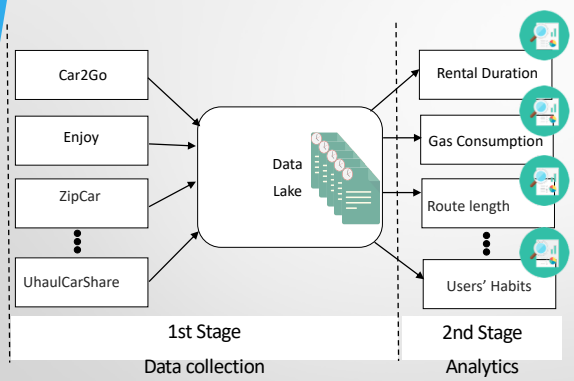
IDEA

- **Collect this data and create a platform to monitor car sharing**



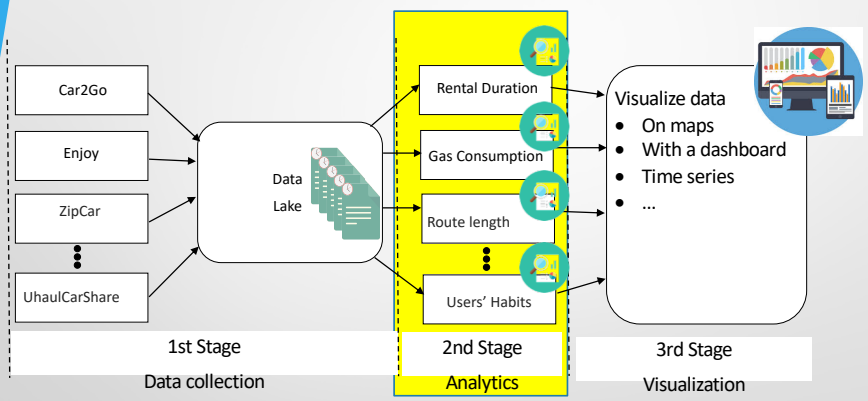
IDEA

- **Collect this data and create a platform to monitor car sharing**



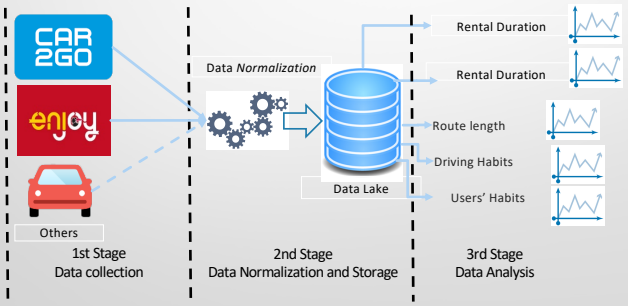
IDEA

- **Collect this data and create a platform to monitor car sharing**



The Big Data approach

- Multiple stage process
  1. **Data Collection**: **Collect** data from **different FFCS** providers
  2. **Data Normalization and Storage**: **Process** the collected data to **normalize** it in a **common format** and save it in a **Data Lake**
  3. **Data Analysis**: **Analyze** the data to extract **meaningful** information



Data Collection

Lab focus – Car2Go

- **30 cities**: 15 in Europe, 15 In North America

Amsterdam
Berlin
Firenze
Frankfurt
Hamburg
Kobenhavn
Madrid
Milano
München
Rheinland
Roma
Stockholm
Stuttgart
Torino
Wien

Arlington County
Austin
Calgary
Columbus
Denver
Miami
Montreal
New York City
Portland
San Diego
Seattle
Toronto
Twin Cities
Vancouver
Washington DC

Data collection

- Harvest the data offered by car sharing platforms, and store them in a repository
- Consider the car2go platform
  - It offers public API to collect the data [need a key]

http://www.car2go.com/api/v2.1/vehicles?loc=turin&oauth_consumer_key=getacar&format=json

- It returns a json file

Data Lake

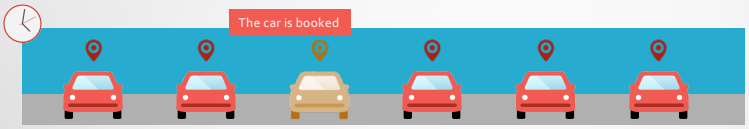

```
{
    "placemarks": [
        {
            "address": "Via Argelati Filippo, 24, 20143 Milano",
            "coordinates": [
                9.1727,
                45.44989,
                0
            ],
            "engineType": "CE",
            "exterior": "GOOD",
            "fuel": 96,
            "interior": "GOOD",
            "name": "581/FF119NT",
            "smartPhoneRequired": true,
            "vin": "WME4533421K145153"
        },
        {
            "address": "Via Santander, 9, 20143 Milano",
            "coordinates": [
                9.16443,
                45.43981,
                0
            ],
            "engineType": "CE",
            "exterior": "GOOD",
            "fuel": 96,
            "interior": "GOOD",
            "name": "285/FF457BT",
```


Data processing– high level intuition

- Every **minute**, for every city, we have information about parked (available) cars.

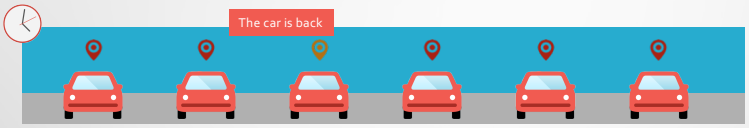
- Every minute, we get a snapshot of available cars

Data processing– high level intuition



- When a car **'disappears'** at time T1
  - It has been booked [!=rented]
  - We save the state "car booked at time T1"
    - We have temporary collection for activeBookings
  - [And we compute "**parking period = T1-To**"]

Data processing– high level intuition



- When the car '**reappears**' at T=T2
  - The booking has ended
  - We save the state "car available at time T2"
    - We have temporary collection for activeParkings
  - [And we compute "**rental period= T2-T1**"]

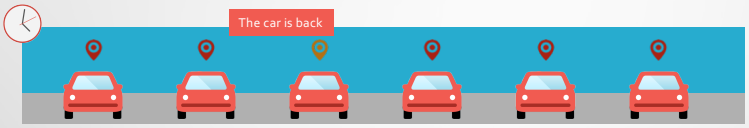
Data processing– high level intuition

- For every city, we have information about parked (available) cars

- Every minute, we get a snapshot of available cars
- We compare this with the latest status for each car
  - We have temporary collections
    - For **activeBookings**
    - For **activeParkings**

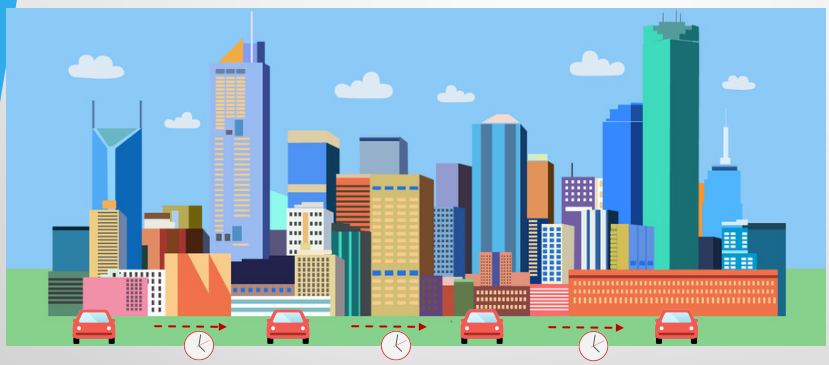
Dataset creation

- **Create a dataset**



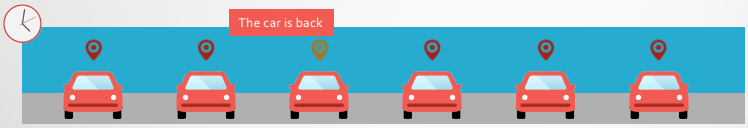
- By repeating this forever, we obtain a longitudinal dataset

DATASET



Dataset creation

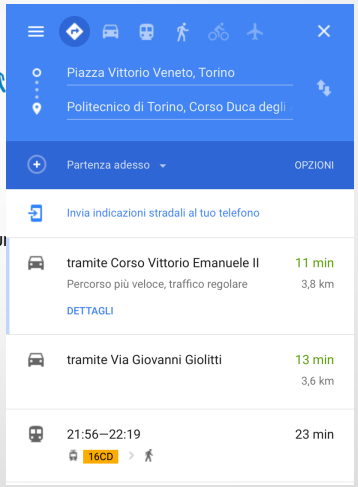
- **Create a dataset**



- By repeating this forever, we obtain a longitudinal dataset
  - A collection for **PermanentBookings**
    - With durations of car booking periods
  - A collection for **PermenentParkings**
    - With duration of car parking periods

Data Lake

Dataset integration



- Integrate the data with other sources
  - Given a rental from [x0,y0] to [x1,y1]
  - Which was the possible path the driver followed?
  - Which should be the minimum time to complete jou
  - How long would it take to walk instead of driving?
  - How long would it take to use a bus/tram?
- Use google map to integrate the data
  - When a booking ends, query gmap for data
    - Limitations due to the number of (free) queries

Caveats

- In reality, things are more complicated
- A car may disappear, and reappear in the same place
  - GPS fix position error
  - Booking that has been cancelled
  - System issues?
- Car may disappear for (very) long time
  - Car brought to maintenance
- A lot of car may disappear at the same time
  - Maintenance?
  - System issues?
- Gmaps data may be missing
- …
- Always double check what you get!

Repository for dataset

Data collection

- Which technology for our project?
- We have
  - json files
  - Heterogeneous platforms (car2go, enjoy, bluetorino, …)
- NoSQL technology seems more adequate
  - Data can be not structured
  - Data may be missing
  - They are simpler to be used
  - They scale horizontally
- MongoDB is a free and open source document oriented DB
  - Well suited for our goals

Data Lake

MongoDB

- It is a scalable NoSQL DB
- Uses JSON-like documents with schemas
- It has drivers for a variety of popular programming languages
  - Python, C, C++, Java, …
- Simple query mechanisms

```
db.getCollection('ActiveBookings').find({})
db.getCollection('ActiveBookings').find({city: "torino"})
db.getCollection('ActiveBookings').find({city: "torino"}).count()
db.getCollection('ActiveBookings').find({city:
"torino"}).sort({plate: 1})
db.getCollection('ActiveBookings').distinct("city")
```

- We will learn how to use mongoDB for queries
  - The goal is not to become MongoDB masters, but rather to be practitioners

MongoDB - concepts

- Think of **documents** as database **records**
  - Documents are just JSON objects that MongoDB stores in binary
- Think of **collections** as database tables

RDBMS (mysql, postgres)	MongoDB
Tables	Collections
Records/rows	Documents/objects
Queries return a record	Queries return a **cursor**

MongoDB - concepts

- Queries return "cursors" instead of a collections
  - A cursor allows you to iterate through the result set
  - A big reason for this is performance
  - Much more efficient to load results into memory
    - Especially if results are big as in big data
- The find() function returns a cursor object

```
var c = db.ActiveBookings.find( {city: "Torino"} )
var i = 0
while (c.hasNext() && i<10)
{
  var o = c.next() // this is the object
  print(o.init_time + " " + o.city)
  i++
}
```

DEMO

ROBOMONGO

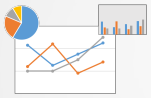
Analytics

CARSHARING DASHBOARD

- The goal of the labs is to derive higher level information from a dataset
- Develop analytics which, starting from data and external information, show results and statistics in plots
  - Given a dataset
    - Car2Go dataset in different cities
  - Derive typical statistics related to transports
  - Get insight about the usage of free floating car sharing

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.



CarSharing Statistics

Number of booked cars

Booking duration

(Guessed) time of booking

(Guessed) time of driving

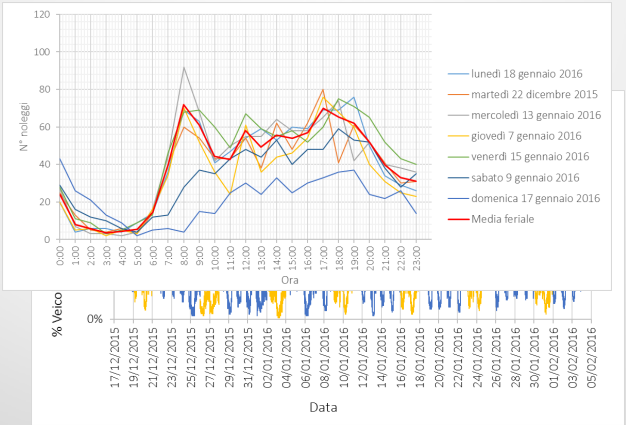
Parking Duration

Locations

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.

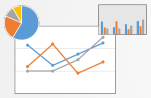
Number of booked cars

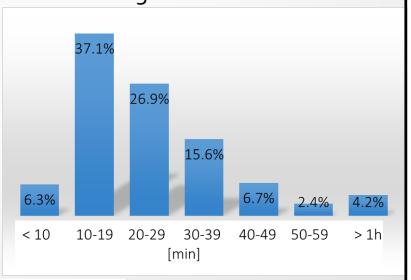


CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.

CarSharing Statistics





Booking duration(*)

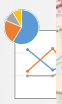
(*) for a proper definition of duration…

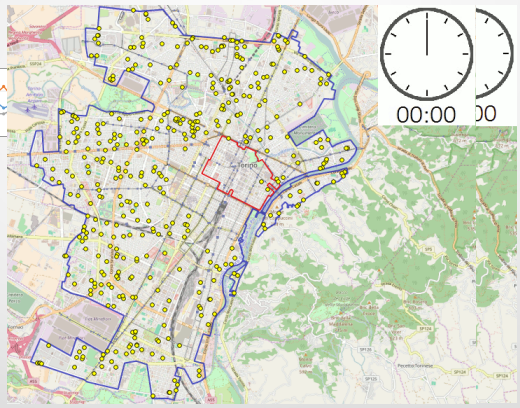
Rental Type(**)	Total	Fraction [%]
Chain	3297	8.8%
Single	31940	84.9%
Single*	1517	4.0%
tour	849	2.3%

(**) for a proper definition of type

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.

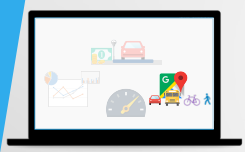




Locations

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.



Google APIs
[Google Maps and Google Direction API]

- Develop a system which, starting from data, shows a dashboard offering statistics
  - Correlate rentals
  - with Public Transport availability
  - Walking distance
  - Time of day
  - …

Lab organization

Lab organization

- Students work in groups of 3 people
  - 2 is too small, 4 is too big
- Better having a mixed background
  - As when you get your job
- We give you access to a MongoDB database
- You extract the data, process it, obtain statistics, and plot them
- Useful tools
  - Data extraction: A bit of MongoDB
  - Data postprocessing: A bit of python | Java | awk | bash | yourchoice
  - Data plotting: Matlab | Matplotlib | Gnuplot | excel | yourchoice
  - Report editing: Word | Openoffice | LaTeX | yourchoice

Lab Evaluation

- Each group writes a 5pp report + appendix
  - Describing what you have done
  - The results you got
  - Source code goes in appendix
- The group report must be uploaded on the didattica website
  - Filename: **groupXX.pdf**, one file per group, latest copy is evaluated
  - Deadline: 23:59 of the same day of the exam reservation deadline

Lab Evaluation

- The report is evaluated
  - On a range between [ins:30]
    - It's about 25% of final vote
  - All students get the same vote for the report
    - No report => no oral exam
    - Insufficient report => no oral exam
    - Copied report => no oral exam + disciplinary committee
- During the oral exam
  - One question on your report to discuss what you did
    - And check that you did it
    - Copied report => no oral exam + disciplinary committee
  - No answer => no pass

Lab Organization

- Formation of lab groups
  - Go to the gDrive document and enter your group

    https://goo.gl/SqdTDY

  - No preference => I choose
- One or more PC per group

QUESTIONS

- Ask whether there are doubts or something not really clear.