Project (P1)

Big Data Management

Project's objective

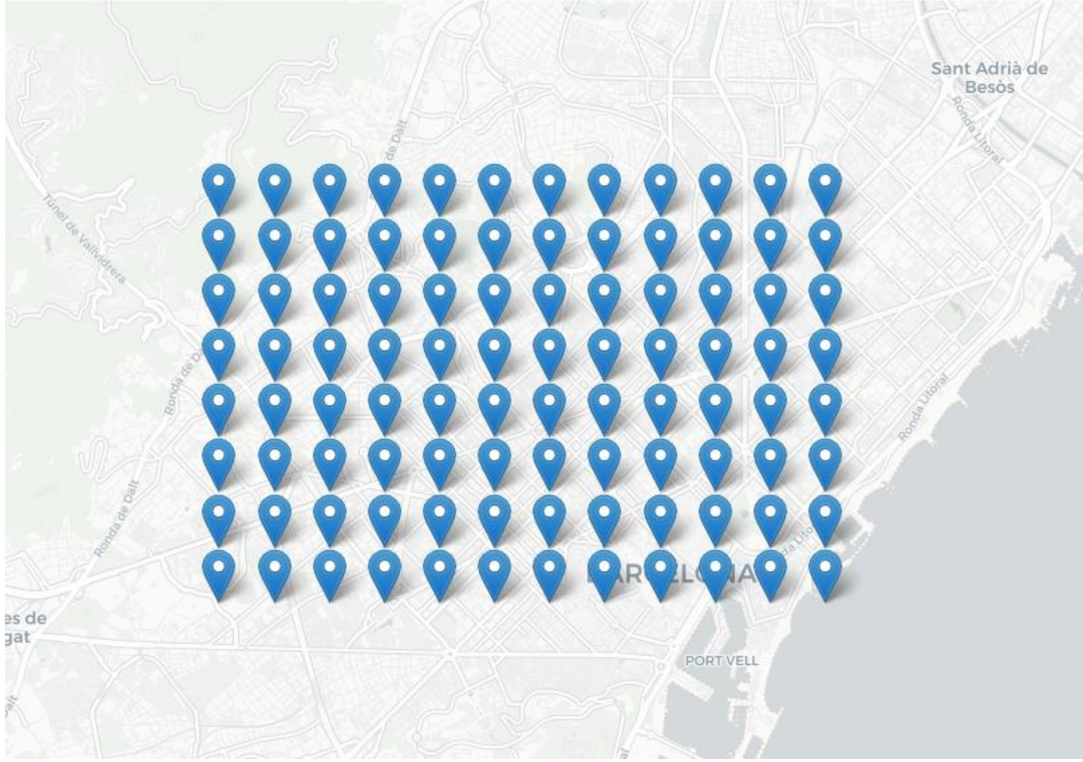
- Descriptive and predictive analysis of data related to Barcelona's housing and the relationship with its economy
- Examples of descriptive analysis KPIs
  - Average number of new listings per day
  - Correlation of rent price and family income per neighborhood
- Examples of predictive analysis KPIs
  - Estimate the rental price for a new apartment
  - Evaluate the deviation of a predicted price with respect to the real average price in a neighborhood

Mandatory datasets

- Barcelona rentals
  - idealista
- Territorial distribution of income
  - Open Data Barcelona
- Lookup tables

D1 - Barcelona rentals

- JSON documents
- Listings for apartments downloaded once per day on a random point in Barcelona’s grid (1.5 km radius)
- Ingestion date encoded in the filename

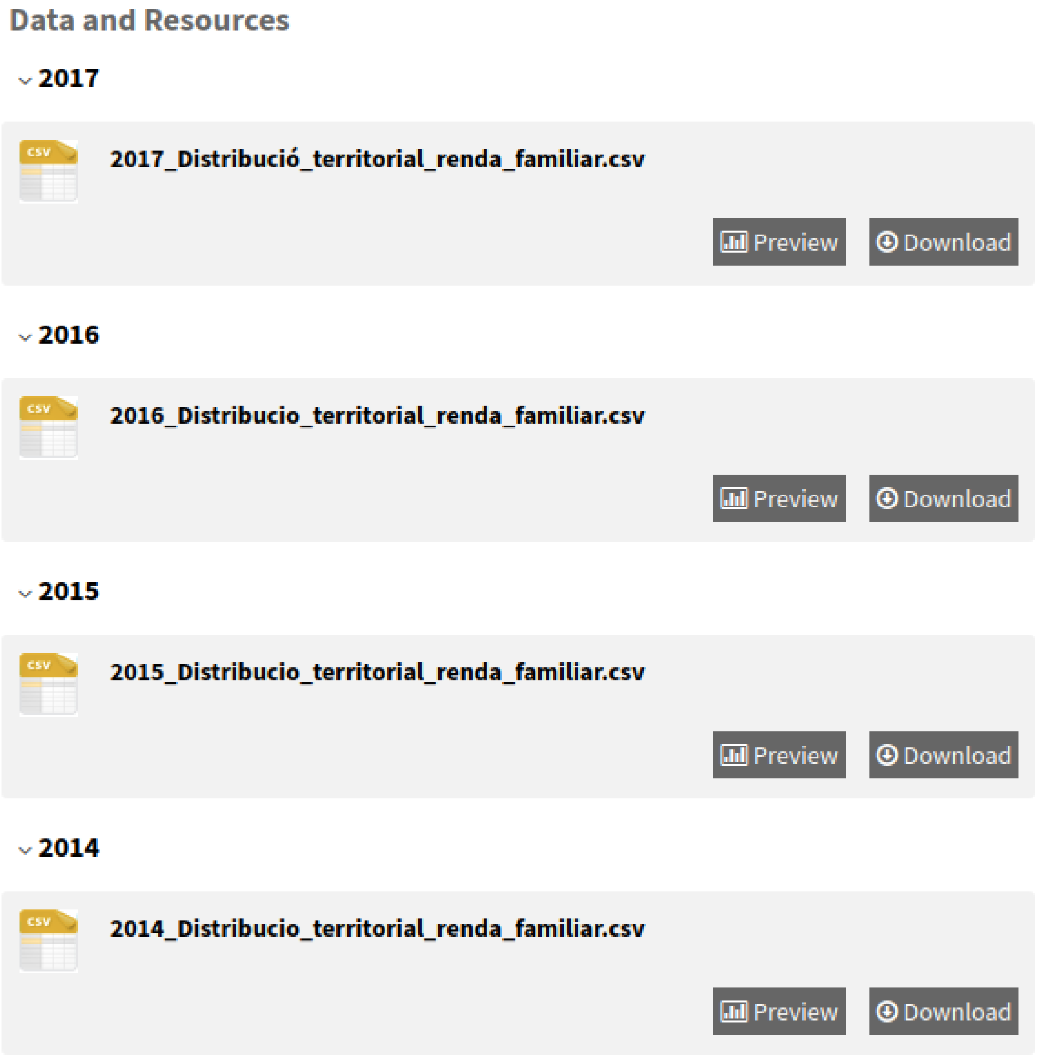


D2 - Territorial income distribution in the city of Barcelona

- CSV files
- Population and RFD (family income index
  - Per year (encoded in the filename)
  - Per neighborhood

Open Data **BCN**

Ajuntament de Barcelona's open data service



Data and Resources

2017

2017_Distribució_territorial_renda_familiar.csv

2016

2016_Distribucio_territorial_renda_familiar.csv

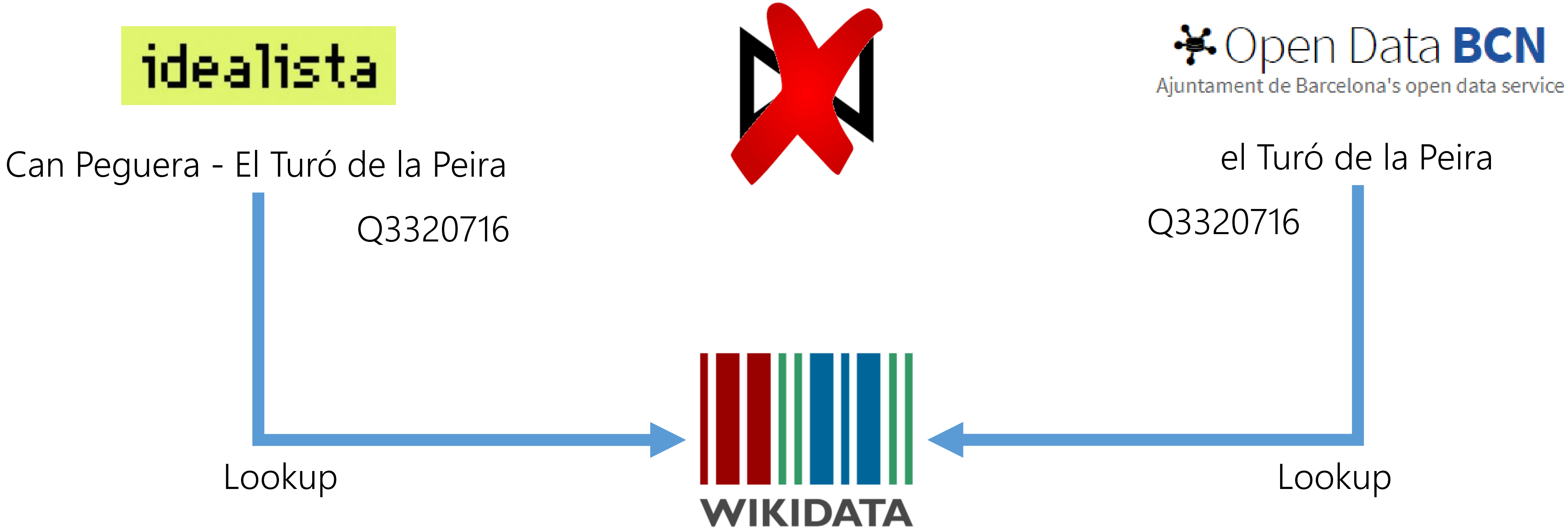
2015

2015_Distribucio_territorial_renda_familiar.csv

2014

2014_Distribucio_territorial_renda_familiar.csv

Data reconciliation



Can Peguera - El Turó de la Peira

Q3320716

el Turó de la Peira

Q3320716

Lookup

Lookup

D3 - Lookup tables

- Two CSV files (D1 and D2)
- For each distinct district and neighborhood their Wikidata ID

Organization

Three parts

- P1 – Data design
  - Conceptualization and Data Lake design
  - Technologies: Apache Hadoop (+ file formats), Apache HBase, MongoDB
- P2 – Descriptive analysis
  - Data integration and reconciliation (+ 3rd dataset)
  - Technologies: Apache Spark (core and/or SQL), a visualization tool (e.g., Tableau)
- P3 – Predictive analysis
  - Distributed machine learning and real-time data prediction
  - Technologies: Apache Spark (MLlib, Streaming), Apache Kafka, a visualization tool for streams (e.g., Kibana)

- A possible solution will be provided after P1 and P2

Teams

- Work in pairs
  - You have to define the teams
  - All pairs must be different in P1, P2 and P3
- How to deal with the incremental nature of the project?
  - You are free to extend the solution of the other team member
    - As long as both members of the team agree
  - Otherwise, use the provided solution

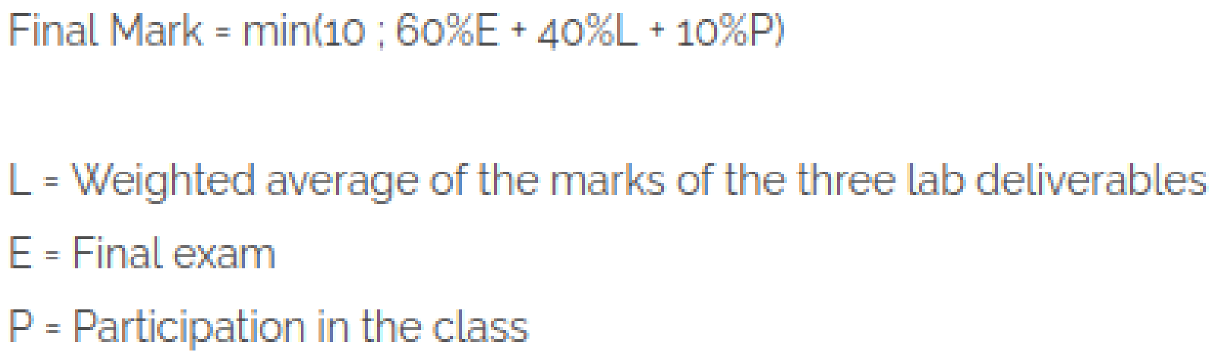
Development environment

- Virtual machines hosted at FIB (Virtech)
  - Ubuntu Desktop with HDFS, HBase and MongoDB installed in standalone mode
  - See the manual in LearnSQL
  - Credentials will be provided in the team's description
    - Create them ASAP!
- Your own development environment
  - Java (intellij IDEA)
  - Python (PyCharm)

Validation tests

- Each part (P1, P2, P3) will have associated a validation test
  - See specific dates in LearnSQL
- Individual test
- Questions related to the project development and its relationship with the concepts studied in class

Evaluation



Final Mark = min(10 ; 60%E + 40%L + 10%P)

L = Weighted average of the marks of the three lab deliverables

E = Final exam

P = Participation in the class

- L = 0,2 * P1 + 0,4 * P2 + 0,4 * P3
- Where, each Pi is computed as
  - Pi = 0,4 * Ti + 0,6 * Di
  - where Ti is the mark on the validation test, and Di is the deliverable’s mark

Data Design

P1

Objectives

- Familiarize with the datasets
- Propose and deploy your data model
  - Implement the required transformations
- Conceptualize the data processing pipelines required for the integration
  - These will be implemented in P2

Data design

- Propose the right kind of storage, data model and structure for each dataset
- Some possibilities are:
  - A distributed file system with Big Data formats
    - Using some of the studied Big Data formats: SequenceFile, Avro, Parquet
  - A column-family key-value store
    - Apache HBase
  - A document store
    - MongoDB

- There is not a single correct solution
  - The most important part is how you **justify your choices**, and **discuss pros/cons**

Delivery

- Document (max 5 pages)
  - For each dataset, proposing its design and data storage
    - Every choice must be justified!
  - Present in a high-level manner (BPMN, sequence diagrams, boxes and arrows, …) the data transformations implemented
- Code to deploy the proposed design
  - Java/Python
  - Only the required transformations to deploy the design are expected
  - We do not expect data cleaning, integration, etc.
    - This will be implemented in P2

Questions?