Project (P2)

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

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
  - 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)

P2 objectives

- Integrate the three provided datasets
  - Handle duplicates, reconciliate data, clean, etc.
- Propose an extra dataset (qualifies for +1 point)
- Implement the calculation of three KPIs
  - Store them in views (DB, files, …)
    - "Average number of new listings per day"
    - "Correlation of rent price and family income per neighborhood"
    - *Propose a new one that also considers the new dataset*
- Graphically display the KPIs

Datasets

- Mandatory datasets
  - Barcelona rentals
    - idealista
  - Territorial distribution of income
    - Open Data Barcelona
  - Lookup tables
- Extra dataset
  - You can check out OpenData BCN portal or other Open Data portals
  - You might need to implement your own reconciliation process
    - See LearnSQL for a guideline on how to run a reconciliation process with OpenRefine

A (given) solution for P1

- Barcelona rentals
  - Each JSON file has been converted to a Parquet file
- Territorial distribution of income
  - CSV converted to one MongoDB collection
    - Use mongoimport to import the file into a collection
- Lookup tables
  - Four MongoDB collections
    - rent_lookup_district
    - rent_lookup_neighborhood
    - income_lookup_district
    - income_lookup_neighborhood
  - Prefixed attributes

    - di → district
    - n → name
    - ne → neighborhood
    - re → reconciled

Technologies

- Apache Spark
  - Integration and reconciliation using lookup tables
  - Calculate KPIs and store them in views
  - Your pipeline must be optimal from the perspective of…
    - Minimizes the number of wide dependencies
    - Caches results when required
    - Exploits parallelism
    - …
- Visualization tool
  - Choose the one you prefer
  - Provide online access or a video of the resulting solution

Delivery

- Document (max 5 pages)
  - Describe the pipelines to integrate and to calculate/store of KPIs
    - Sketch the pipelines at a higher abstraction level. Use the notation seen in the lectures to describe the Spark job
    - Elaborate on your assumptions. Refer to any specificity of your solution that should help the lecturer to understand the decisions you made in your code that, otherwise, might look like controversial
  - Describe the extra dataset and new KPIs (optional, +1 page)
- Code
  - Spark (Java, Python, Scala)
- Extra material
  - Online access to visualization tool, videos, etc.