Project (P3)

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 sale price and family income per neighborhood
  - Top-seller neighborhoods in a time window
- Examples of predictive analysis KPIs
  - Evaluate the deviation of a predicted price with respect to the real average price in a neighborhood
  - Predict the family income index of a neighborhood based on its sale price

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), a visualization tool (e.g., Tableau)
- P3 – Predictive analysis
  - Distributed machine learning (ML) and real-time data prediction
  - Technologies: Apache Spark (MLlib, Streaming), Apache Kafka, a visualization tool for streams (e.g., Kibana)

P3 objectives

- Prepare the input data and train an ML model
  - Store it in disk
- Ingest a data stream
  - Perform predictions applying the model on the data stream elements
  - Describe the data stream using approximate stream analysis algorithms
- Graphically display the output (qualifies for +1 point)

A (given) solution for P2

- Result of the reconciliation in P2
- CSV input file
  - Reconciliated neighborhood ID
  - Date
  - Unique property code
  - Reconciliated neighborhood name
  - Sale price
  - Family income index (RFD)

```
neigh_id,date,propertyCode,neigh_name,price,RFD
Q980253,2020_06_22,92608642,El Poble Sec - Parc de Montjuïc,185000,82.2
Q1758503,2021_02_12,88001229,El Raval,255000,71.2
Q1627690,2020_08_06,91955139,Les Roquetes,170000,49.7
Q3321657,2020_08_12,86104553,Sant Gervasi - La Bonanova,2500000,184.6
Q3321805,2020_07_20,92247543,El Putxet i el Farró,410000,144.6
```

Distributed machine learning

- Create two datasets – perform the necessary transformations, cleaning
  - Training
  - Validation
- Use the training dataset to create a classifier using Spark MLlib (RDD-based)
  - https://spark.apache.org/docs/latest/mllib-guide.html
- You are free to choose the kind of model
  - The objective of the course is not to optimize this part
- Validate the model
  - Compute recall and accuracy
- Store the model

Near real-time analysis

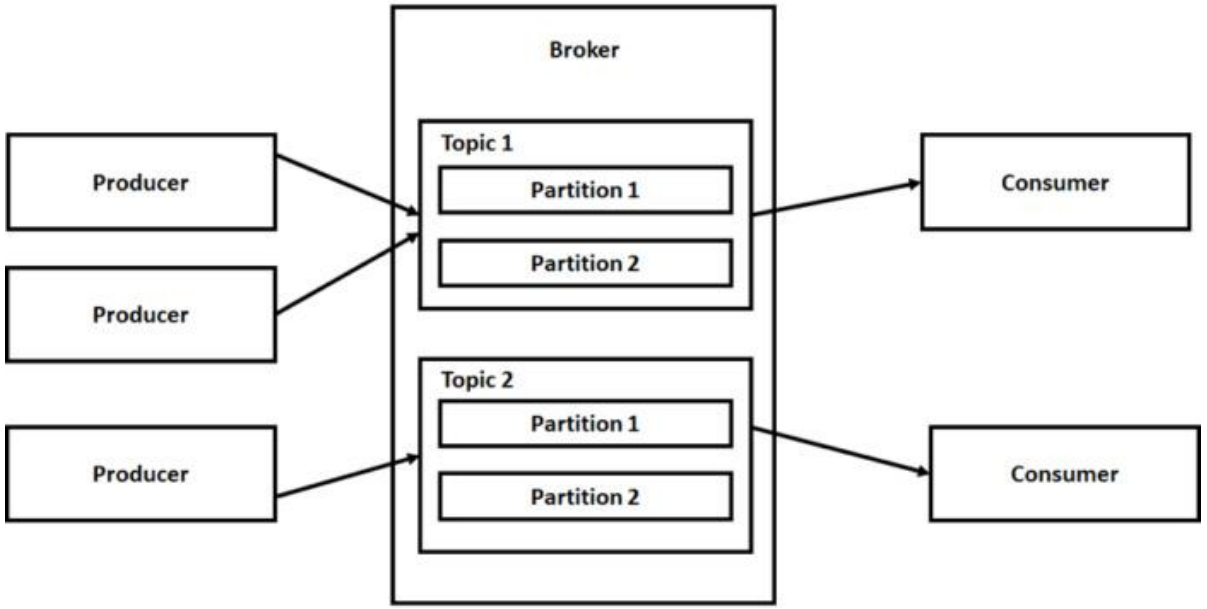
- Ingest and process a data stream
- Predictive analysis (use the classifier you created before)
  - a) predict the rental price for a new apartment; or
  - b) predict the family income index of a neighborhood based on its sale price
- Stream analysis (approximate algorithms)
  - Load shedding
    - E.g., maintain a set of representatives that guarantee an unknown query can be answered
  - Bloom filters
    - E.g., filter a black/white list of properties
  - Exponentially-decaying window
    - E.g., maintain the historical average price per neighborhood
  - Heavy hitters
    - E.g., top-seller neighborhoods in a time window

Technologies

- Apache Kafka
  - Endpoint for stream ingestion
- Apache Spark
  - Transformations and matrix preparation
- Apache Spark MLlib
  - Classifier and evaluation
- Apache Spark Streaming
  - Stream operators
    - window
    - reduceByKeyAndWindow
    - updateStateByKey
    - mapWithState
- Real-time visualization tool (optional, qualifies for +1 point)
  - Choose the one you prefer (Kibana, Streamlit, …)
  - Provide online access or a video of the resulting solution

Kafka endpoint

- A message queue for raw data streams that are pushed from the data sources



- Available at *sandshrew.fib.upc.edu:9092*, topic *bdm_p3*
  - Check out the example code for integrating Spark Streaming + Kafka

Delivery

- Document (max 5 pages)
  - Describe the pipelines to train/test the classifier
  - Describe the streaming algorithms chosen and their implementation in Spark
    - Justify the choice of streaming operator
    - 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 visualization component (optional, +1 page)
- Code (Java, Python, Scala)
  - Pipeline to train classifier
  - Pipeline to test classifier and describe the stream using **two** approximate algorithms
- Extra material
  - Online access to visualization tool, videos, etc.