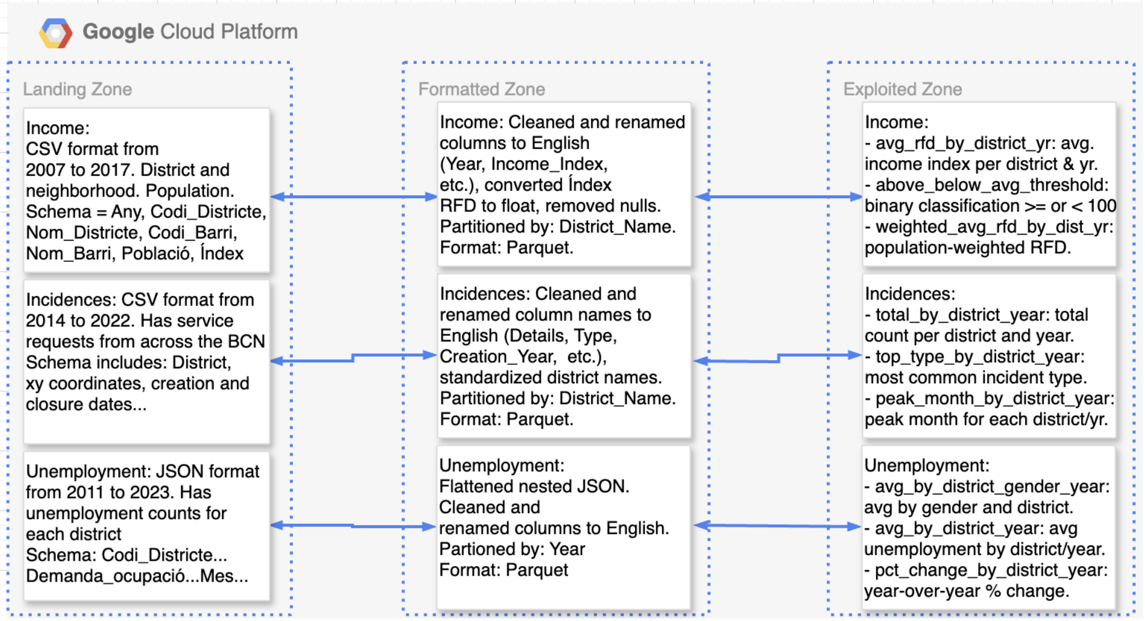
Big Data Management Lab 3: Spark (Data-driven architecture)

Victor Sobottka & Tirdod Behbehani

June 24, 2025

1 Sketch Data Pipelines



Income:
CSV format from 2007 to 2017. District and neighborhood. Population.
Schema = Any, Codi_Districte, Nom_Districte, Codi_Barri, Nom_Barri, Població, Índex

Incidences: CSV format from 2014 to 2022. Has service requests from across the BCN
Schema includes: District, xy coordinates, creation and closure dates...

Unemployment: JSON format from 2011 to 2023. Has unemployment counts for each district
Schema: Codi_Districte... Demanda_ocupació...Mes...

Income: Cleaned and renamed columns to English (Year, Income_Index, etc.), converted Índex RFD to float, removed nulls.
Partitioned by: District_Name.
Format: Parquet.

Incidences: Cleaned and renamed column names to English (Details, Type, Creation_Year, etc.), standardized district names.
Partitioned by: District_Name.
Format: Parquet.

Unemployment:
Flattened nested JSON.
Cleaned and renamed columns to English.
Partioned by: Year
Format: Parquet

Income:
- avg_rfd_by_district_yr: avg. income index per district & yr.
- above_below_avg_threshold: binary classification >= or < 100
- weighted_avg_rfd_by_dist_yr: population-weighted RFD.

Incidences:
- total_by_district_year: total count per district and year.
- top_type_by_district_year: most common incident type.
- peak_month_by_district_year: peak month for each district/yr.

Unemployment:
- avg_by_district_gender_year: avg by gender and district.
- avg_by_district_year: avg unemployment by district/year.
- pct_change_by_district_year: year-over-year % change.

Figure 1: Data Pipeline Architecture across Landing, Formatted, and Exploited Zones

2 Selected Datasets

The three datasets that we selected were:

- `income`: Contains yearly average family disposable income (RFD) for each neighborhood and district in Barcelona from 2007 to 2017. Also contains the population of that neighborhood. The district for each neighborhood is also included. Computed KPIs were average RFD per district, districts with RFD above/below 100 in a given year, population-weighted income index.

- `incidences`: Contains citizen service requests from 2014 to 2022. Includes geographic coordinates of the incident, neighborhood and district of the incident location, the date that the service was was created and closed, the manner in which the request was created, type of incident, etc. Computed KPIs were incident totals for each district per year, top incidents for each district per year, and peak incident months by district per year.

- `unemployment`: Reports unemployment data for each neighborhood/district in Barcelona from 2011 to 2023. Dataset includes unemployment counts and is separated by gender. Computed KPIs include average unemployment rate by gender for each district in a given year, average unemployment rate (not separated by gender) for each district in a given year, and percent change in unemployment for each district in a given year.

3 Data Management Backbone

Explore the data and choose the KPI's

For the data exploration, we created a host of plots across all three datasets. While going through these plots, our goal was to understand the dataset structure, distribution, and missing values. This analysis drove our KPI selection and helped inform the preprocessing that was required for model development.

The computed KPIs were:

- `income`: Average RFD per district, districts with RFD above/below 100 in a given year, population-weighted income index.
- `incidences`: Contains citizen service requests from 2014 to 2022. Incident totals for each district per year, top incidents for each district per year, and peak incident months by district per year.
- `unemployment`: Average unemployment rate by gender for each district in a given year, average unemployment rate (not separated by gender) for each district in a given year, and percent change in unemployment for each district in a given year.

Data Formatting Process

In the formatting stage, we standardize and clean the datasets to prepare them for downstream analysis. Some of the key steps included:

- Translating all column names from Catalan to English for interpretability.
- Fix data types errors (ex. converting year string to integer).
- Filter out "non-incidents" in the incidents dataset
- Un-nest JSON structure in the unemployment dataset.
- Standardized district and neighborhood names/codes, ensuring that the same districts/neighborhoods did not have mismatching names/codes.

Following the data formatting, datasets were exported to the formatting zone in parquet format and were partioned by district and year. This was done to improve query performance and support time-based analysis.

Move Data to the Exploitation Zone

Once data was present in parquet format in the exploitation zone, we were ready to compute the relevant KPIs for each dataset. The resulting KPIs were exported to the exploitation zone. We unified key identifiers, district code and year, to ensure easy joins across the income, unemployment and incidences datasets. A mapping table for district codes and district names was also created.

Validate the Data

Before rushing onto the predictive analysis, we validated the data to ensure that there weren't any erroneous errors that would have harmed our analysis. This included checking row counts, null values, checking the schema, and thoroughly reviewing the underlying logic for the calculated KPIs. This ensured that the data that we trained our models on was sound and was free of schema mismatches that would have hindered model performanc.e

4 Predictive Analysis

For our models, we seek to clasify Barcelona districts as "low income" or "high income". To do so, we create a binary threshold of 100 on the RFD index. RFD values above 100 are labeled as high income (1), and those RFD values below 100 are labeled as low income (0). We utilize selected KPIs from the exploitation zone as the features in our model.

Model 1: Logistic Regression

For our initial model, we used a simple logistic regression. Our goal was to use selected KPIs from the exploitation zone to attempt to predict whether a district is considered to be high income or low income. We used 5 features, aggregating and merging from different datasets in the exploitation zone:

- `total_incidences`: Total number of citizen service requests per district-year.
- `avg_unemployment_rate_per_district_year`: Average unemployment rate for each district-year.
- `incident_type_index`: String-indexed value of the most frequent incident type.
- `gender_unemp_gap`: Difference between male and female unemployment rates.
- `avg_pct_change`: Average yearly percentage change in unemployment rate.

Model Results

The logistic regression achieved an ROC AUC of 0.5556. In the end, this is only marginally better than a random classifier.

Model 2: Random Forest

We decided to use a Random Forest algorithm for our second model since they often outperforming single models like Decision Trees or Logistic Regression and also combine multiples trees reducing overfitting.

The hyperparameters and metrics are defined below:

- Used a hyperparameter grid for `numTrees`, `maxDepth`, and `maxBins`.
- Cross-validated (5 folds) for robust selection.
- Outputs probabilities, predictions, and feature importances.
- Metrics evaluated: **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **ROC AUC**.

Model Results

The Random Forest Classifier achieved a **ROC AUC of 0.6667** and an accuracy of 66.67% on the test set.

- Precision = 0.80 suggests that when the model predicts a district as high - income, it is correct 80 % of the time, indicating relatively few false positives.
- F1 Score = 0.625 balances the trade-off between precision and recall, showing reasonably consistent performance but with room for improvement.

In conclusion the Random Forest classifier outperformed the simple logistic regression as it was expected, since we are working with a more complex model.