Universitat Politècnica de Catalunya

Big Data Management

Descriptive And Predictive Analysis

Eliya Tiram
eliya.tiram@estudiantat.upc.edu

Míriam Méndez
miriam.mendez.serrano@estudiantat.upc.edu

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Curs 2022/2023 Q1

Contents

1 Introduction

The second part of the course project continue last part project where we store parquet files in HDFS server at persistent landing zone. This project focus on implementing a formatted and exploitation zone. Formatted zone, contain reconciled sources, pre-processed, so it could be ready for the exploitation zone in order to perform descriptive and predictive analysis. To begin with, we performed few initialize steps (which described in the annex 5.1).

Our goal is to create an environment for analysts to be able to explore the data and create KPI's. for this purpose we create formatted and exploitation zones and we will create 2 descriptive KPI's and 1 predictive KPI using ML model.

2 Formatted and Exploitation zone

This project continue last project, thus we start by reading from the landing zone. In order to move to the formatted zone, we are going to use spark to load the data, transform it, store it in parquet files in HDFS, so it would be ready for exploitation zone.

We chose parquet files to take advantage on columnar partition for the KPI purposes that are going to be aggregated on some level. In addition, KPIs, in general, are likely to be aggregated data which also support of using parquet, in class we saw a rule-based choice of when using each file type in HDFS.

Parquet files use a columnar storage format that is designed to be highly efficient for analytical workloads. One of the key features of the Parquet format is its unique encoding schemes, which contribute to its performance and storage optimization[1].

By employing these encoding techniques, Parquet files can achieve high compression ratios, minimize disk I/O, and speed up query processing in data analytical systems. Additionally, the encoding schemes in Parquet are flexible, allowing different encoding strategies to be applied to different columns based on their characteristics, further enhancing the overall efficiency of data storage and retrieval[2].

load the data into spark RDD and for the formatted zone we are going to join the data RDD with the lookup table. As a result, our data would be prepared to join the different data RDD to fulfil the KPIs' requirements in the exploitation zone.

2.1 Process High Level Description

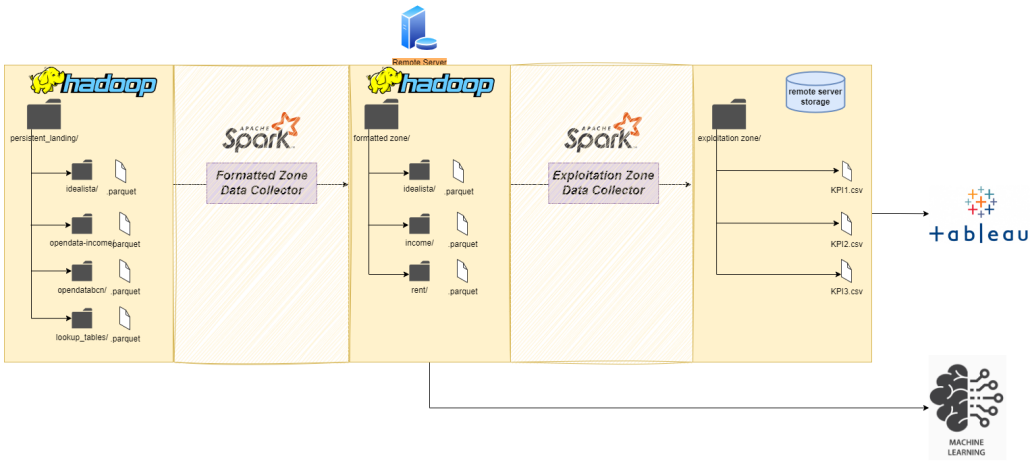

Figure 2.1: Project process flow description

Figure 2.1 describe the process in a high level. The whole project runs on the VM (remote server) and build with 2 data collectors using spark. In addition, shows formatted and exploitation zone which we work with.

Also, you can see the extra data source we are using in the project. From p1 of the project we took data (also from opebBCN), the data present the average price of average rent price per neighborhood in Barcelona.

2.2 Formatted Zone

Formatted zone will contain each data source by itself, since source schema might vary. In addition, when we load data into formatted zone we perform: data profiling, remove duplications and other data quality tasks.

First, we read all the parquet file into RDD for each source, then we join each source with the lookup data so each source will have the same neighborhood and district name. This step is important for future integration of the sources.

For each source we perform data cleaning that includes remove duplication, For example, idealista source has properties that repeats itself for different dates. We used `reduceByKey` and max on the dates to get the most updated property. As well we did reconciliation by join the sources with the lookup tables and add the reconciled values (e.g. district and neighborhood) to the logical key of each source. Finally, we prepared the data for the exploitation zone with the relevant columns for each data source.

To finish the process of the formatted zone, we write the the 3 sources into HDFS as parquet flies (i.e. for each data source we divide the file to be up to 128MB) in the proper structure to be able to build ML model and different KPI's.

2.3 Exploitation Zone

The exploitation zone generating the data view required for the analysis. Therefore, it contains a set of tables that are useful for the data analysis backbone. Thus, we perform data integration and data quality processes for integration.

One of the abilities in spark is to run a SQL query on the data which create a relation-like structure[3], thus even though that we work in NO SQL environment we can still supply a framework that the data analyst could work with.

Once we the environment is ready we can start developing KPI's. In the next part we are going to show in a visualization tool the KPI's we create. In reality we believe that a visualization tool would be directly connected to exploitation zone so the data flow would be complete with prepared dashboard in advance.

3 Data Analysis

For the analysis part we are going to save the data in a CSV. We chose CSV because our data volumes are not too big and can fit in a CSV. Also, our visualization tool is not connected to any live database. All descriptive KPI's are on the same dashboard.

3.1 Average Listing Per Neighborhood

Our first KPI is to show the average price of a neighborhood from our data. It is giving a quick review on the which neighborhood is more expensive or cheaper.

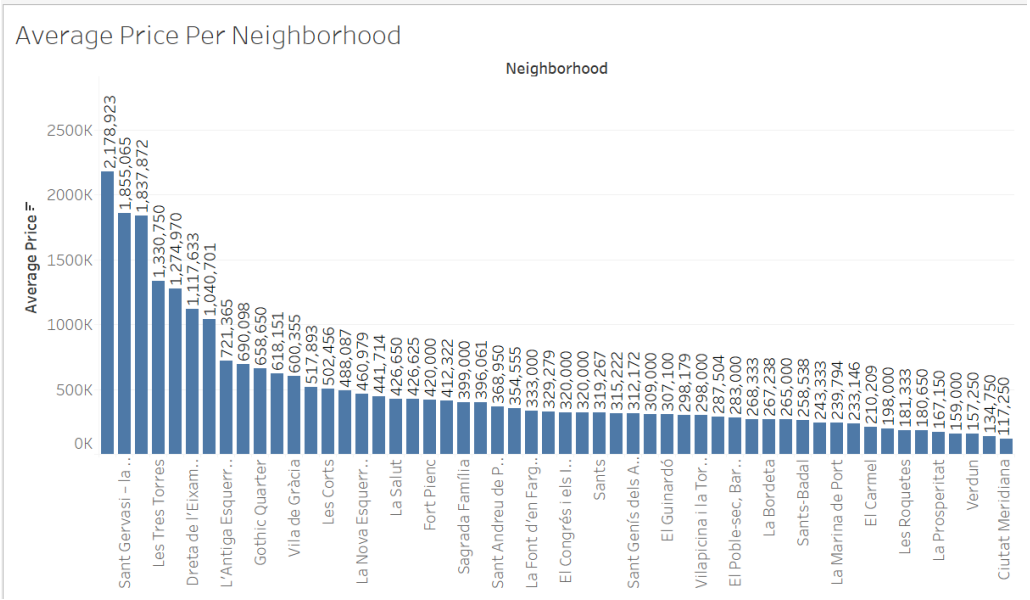

Figure 3.1: Average price per each neighborhood

Figure 3.1 shows a descending bar chart of the average price per neighborhood.

3.2 ROI Per Year and Neighborhood

Return of investment (ROI) index represent the time, according to the rent amount, to return the investment of a property.

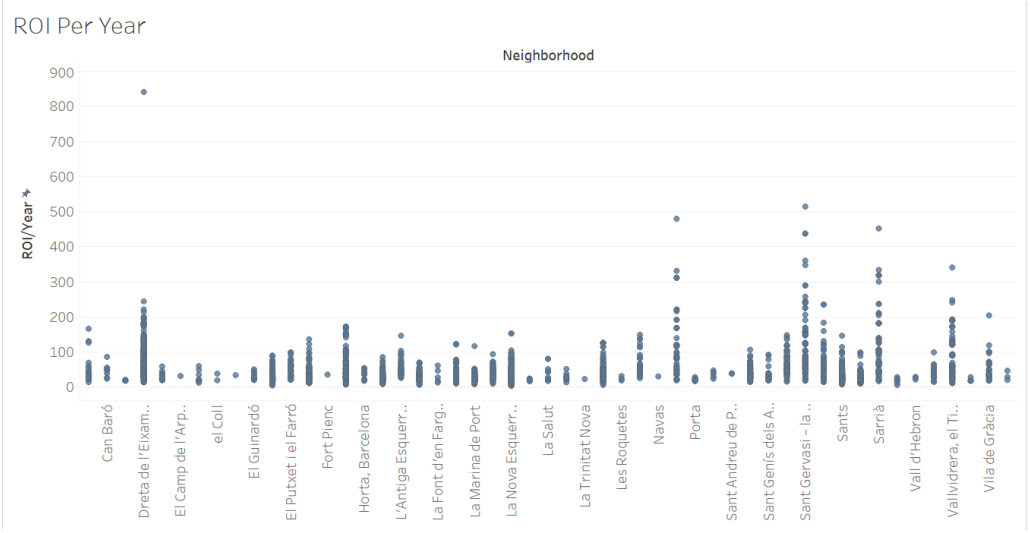

Figure 3.2: ROI in years of a listing

Figure 3.2 shows how many years will it take to return the price that was paid for a given listing. In the dashboard, which attached to this project, when hover over the dots the property code and the year can be seen.

A dashboard was created for the KPI's and attached to this project files.

3.3 Predictive Analysis Using ML

For predictive analysis we implemented a ML model Random Forest (RF). We want to predict what is the price of a listing given the following column: neighborhood, district, date, size, rooms, bathrooms, propertyType, floor, status, hasLift.

Metric	RMSE	R2 Score	Mean Absolute Error
Random Forest	358,768.32	0.74	169,798.68

Table 1: Based on validation data, matrix for model measures

Table 1 present the measures on the RF model we created.

```
+------------------------------+--------+----------+----+-----+---------+------------+-----+------+-------+------+------------------+
|neighborhood                  |district|date      |size|rooms|bathrooms|propertyType|floor|status|hasLift|price |prediction        |
+------------------------------+--------+----------+----+-----+---------+------------+-----+------+-------+------+------------------+
|Camp d'en Grassot i Gràcia Nova|Gràcia  |2020-10-14|90.0|4    |2        |flat        |1    |good  |true   |349000|395507.7736209846|
+------------------------------+--------+----------+----+-----+---------+------------+-----+------+-------+------+------------------+
```

Figure 3.3: Prediction example from the ML model

Figure 3.3 shows one example of a prediction example. given the column mentioned above, the model predict what the price would be. In this example, in Camp d'en Grassot i Gràcia Nova a flat listing on the 1st floor with a lift in a good status and size 90 m2, we predict it will cost 395,507.77

4 Conclusions

In this project, our objective to create descriptive and predictive analysis. In order to achieve this goal we created a formatted and exploitation zone. In the exploitation zone, we have an environment to analyze data and create KPI's based on the processing and integration of the data that we did in the formatted zone.

We have two main scripts we implemented in python, one for getting the data from previous project, clean and reconcile the data, write it into HDFS. The second script, reads the files the first script wrote, join the data in order to create KPI. We chose two descriptive KPI's but there could be many more according to analyst decision.

In addition, we created a ML model to be able to perform predictive analysis. We chose one model based on random forest and we predict the price for a given neighborhood based on variety of variables. like before, any other model can be chosen and any other prediction based on analyst decision.

5 Annex

5.1 Steps To Initialize The Project

As a working environment and since we want to work with spark on the VM (where the parquet files are already found in), we connected to PyCharm professional and an interpreter to the VM. We performed all the vital installations of python on the VM.

For this project we got an new VM so first we ran the P1 project so the parquet files would be in the HDFS server and the P2 of the project is ready to begin. Action that I did on the new server:

- sudo apt update
- sudo apt install python3-pip
- Run P1 to have the parquet files on the new server after adjust the parameters to fit to the new server and to the dataset where it sits on your PC locally. This is .env file with all the configuration for p1.
- P2 of the project should also be adjust to the VM ip and the relevant HDFS properties. After, all the py files can be run on the VM. The order for P2 should be: uploadToFormattedZone.py, uploadToExploitationZone.py and MLmodel.py. An alternative is to run the run.sh file on the server even though we had permission problems, anyway all the files should be executed on the server and not locally, it is possible only if everything set locally.
- Please note that the whole code of the project runs on Pycharm professional when the interpreter is connected to the VM and it has eployment of sync and upload project files to the VM in order for them to run there.
- Project files could be found on the VM in path: /tmp/pycharm_project_743/

References

[1] URL: https://parquet.apache.org/docs/file-format/data-pages/encodings/.

[2] URL: https://parquet.apache.org/docs/overview/motivation/.

[3] URL: https://spark.apache.org/docs/2.2.0/sql-programming-guide.html#running-sql-queries-programmatically.