

# Lab 7

In this lab, we analyze historical data about the stations of the bike sharing system of Barcelona. Your task consists in identifying the most “critical” timeslot (day of the week, hour) for each station and store the result in a KML file that can be visualized on a map.

The analysis is based on two files (available in the HFDS shared folder of the BigData@Polito cluster):

1. /data/students/bigdata-01QYD/Lab7/register.csv
2. /data/students/bigdata-01QYD/Lab7/stations.csv

1) register.csv contains the historical information about the number of used and free slots for ~3000 stations from May 2008 to September 2008. Each line of register.csv corresponds to one reading about the situation of one station at a specific timestamp. Each line has the following format:

- *stationId\timestamp\tusedslots\tfreeslots*
  - where *timestamp* has the format *date time*

For example, the line

```
23 2008-05-15 19:01:00 5 13
```

means that there were **5** used slots and **13** free slots at station **23** on **May 15, 2008** at **19:01:00**.

Pay attention that the first line of register.csv contains the header of the file. Moreover, some of the lines of register.csv contain wrong data due to temporary problems of the monitoring system. Specifically, some lines are characterized by used slots = 0 and free slots = 0. Those lines must be filtered before performing the analysis.

2) stations.csv contains the description of the stations. Each line of registers.csv has the following format:

- *stationId\tlongitude\tlatitude\tname*

For example, the line

```
1 2.180019 41.397978 Gran Via Corts Catalanes
```

contains the information about station **1**. The coordinates of station **1** are 2.180019,41.397978 and its name is **Gran Via Corts Catalanes**.

# Ex. 1

Write a single Spark application that identifies the most “critical” timeslot for each station. This analysis can support the planning of the rebalancing operations among stations. **Solve the problem by using RDDs. Do not use Datasets, DataFrames, and the other Spark SQL features in this practice** (you will use them to solve this problem during the next practice).

In this application, each pair “day of the week – hour” is a timeslot and is associated with all the readings associated with that pair, independently of the date. For instance, the timeslot “Wednesday - 15” corresponds to all the readings made on Wednesday from 15:00:00 to 15:59:59.

A station  $S_i$  is in the critical state if the number of free slots is equal to 0 (i.e., the station is full).

The “criticality” of a station  $S_i$  in the timeslot  $T_j$  is defined as

$$\frac{\text{number of readings with num. of free slot equal to 0 for the pair } (S_i, T_j)}{\text{total number of readings for the pair } (S_i, T_j)}$$

Write an application that:

- Computes the criticality value for each pair  $(S_i, T_j)$ .
- Selects only the pairs having a criticality value greater than a minimum criticality threshold. The minimum criticality threshold is an argument of the application.
- Selects the most critical timeslot for each station (consider only timeslots with a criticality greater than the minimum criticality threshold). If there are two or more timeslots characterized by the highest criticality value for a station, select only one of those timeslots. Specifically, select the one associated with the earliest hour. If also the hour is the same, consider the lexicographical order of the name of the week day.
- Stores in one single KML file the information about the most critical timeslot for each station. Specifically, the KML file must contain one marker of type Placemark for each pair  $(S_i, \text{most critical timeslot for } S_i)$  characterized by the following features:
  - StationId
  - Day of the week and hour of the critical timeslot
  - Criticality value
  - Coordinates of the station (longitude, latitude)

Do not include in the KML file the stations for which there are no timeslots satisfying the minimum criticality threshold.

The output KML file has the following format:

```
<kml xmlns="http://www.opengis.net/kml/2.2"><Document>
  <Placemark><name>44</name><ExtendedData><Data
    name="DayWeek"><value>Mon</value></Data><Data
    name="Hour"><value>3</value></Data><Data
    name="Criticality"><value>0.5440729483282675</value></Data></ExtendedData><
    Point><coordinates>2.189700,41.379047</coordinates></Point></Placemark>
  <Placemark><name>9</name><ExtendedData><Data
    name="DayWeek"><value>Sat</value></Data><Data
    name="Hour"><value>10</value></Data><Data
    name="Criticality"><value>0.5215827338129496</value></Data></ExtendedData><
    Point><coordinates>2.185294,41.385006</coordinates></Point></Placemark>
</Document></kml>
```

The KML file is composed of the three sections:

1. Header
2. One line for each marker of type Placemark
3. Footer

The template of the project available for this lab includes:

- The class `DateTool` characterized by the static `String DayOfTheWeek(String timestamp)` method. The provided method returns the day of the week of the timestamp provides as parameter. For instance the instruction `DateTool.DayOfTheWeek(" 2017-05-12")` returns the string "Fri".
- The code that is needed to write a HDFS file (at the end of the main method of `SparkDriver.java`). You cannot use the standard approach based on `saveAsTextFile()` because the content of the output file is obtained concatenating three different parts in the proposed order:
  - 1) The fix header of the output KML file.
  - 2) One line for each marker. The content of this central part is associated with the content of the `JavaRDD<String>` containing the result of your elaboration.
  - 3) The fix footer of the output KML file.

The content of the KML file can be used to show on a Google map the stations characterized by a high criticality value and the associated information. You can visualize the result of you analysis on a map by copy and paste the content of the generated KML file in the form of the following web page <http://display-kml.appspot.com/> or you can use one of the many other KML viewers available online.