Spark Mini Project

Working with Spark

Objectives

Deliverable: a R or Python script / notebook implementing required features as specified here after

Overview: Load & process gas price data with Spark:

- ❑ Collect data
- ❑ Prepare data
- ❑ Visualize gas prices
- ❑ Model gas prices evolution

OR

- ❑ Model gas price competition

Data Collection

Download 2014 to 2017 gas price data (3 years minimum) from Github (https://github.com/rvm-courses/GasPrices).

Feel free to download additional previous years if you wish to test performances on higher volumes.

Also download gas Stations file & Services file (2017 versions).

Bonus question (not mandatory for the project evaluation):

- write a R function / Python function / Shell Script that download files from Github to a directory provided to the function (independent)

Bonus question (not mandatory for the project evaluation):

- use a yaml file as a parameter file to provide global values to the script (location on GitHub, first year to process, last year to process)

Data Preparation – step 1

With Spark, either in R or Python (your choice):

- Read and merge all gas files
- Split date in year, month, week of the year
- Prepare latitude & longitude for mapping (divide by the right power of 10)
- Make data available as a table in order to be able to use Spark SQL

Data Preparation – step 2

Compute price index for each station per week:

- Compute a new variable called "Price Index" for each gas type sold in a station such as:

$$\boldsymbol{Price\ Index} = \mathbf{100} \times \left(\frac{\boldsymbol{Day\ Price\ in\ station} - \boldsymbol{Average\ Day\ Price\ in\ France}}{\boldsymbol{Average\ Day\ Price\ in\ France}} + \mathbf{1} \right)$$

Compute week index:

- Compute a new variable called "Week Index" for each record counting the number of week since the first week in the file.
- Example: if you loaded years 2015 to 2017, first week of 2015 should be numbered 1, last week of 2017 should be numbered 156 (3 x 52)

Data Visualization

Using ggplot or matplotlib (depending on the API language you selected, R or Python)

- Represent the weekly evolution of average gas price over France such as:
  - Each gas type is a line
  - X coordinates is the week index
  - Y coordinates is the average price for gas type in France over the week index

Hint for Python: consider using Seaborn FacetGrid for multiple lines graph.

Data Visualization – Bonus question

This question provides bonus points (+10%, up to 100%)

- Represent a grid for each gas type of France geo heat maps of price indexes, for example at department level

Modeling – Option 1 – Forecast next day price

!!! Please select either option 1 or 2 for the modeling step of the project !!!

- Build a model based on Spark ML to forecast the next day price for a gas type in a station
- Do not consider using time series models (such as AR/ARMA/ARIMA) but rely on existing techniques from Spark ML / Mllib such as LinearRegression, RandomForestRegressor
- Provide relevant accuracy measures and a relevant dispersion plot between actual & forecast
- **Hints**: consider using lag features (such as price of the day before, and the day before, etc.) to build the model
- **Important Notice**: we do not expect model fine tuning, but more a working pipeline

Modeling – Option 2 – Model Gas Price Competition

!!! Please select either option 1 or 2 for the modeling step of the project !!!

- We would like to test if local competition is important for pricing in the gas market and what is the range of this competition
- Use a model to estimate if competition within 1 km, 3 km, 5 km or 10 km has an impact on prices in a station
- You do not have to use geospatial libraries with Spark (such as in http://www.kartikramalingam.com/scaling-geo-spatial-analysis-with-apache-spark/) but you have to explain how you efficiently find stations in the vicinity of an other
- **Hints**: you may refer to the following paper for inspiration https://drive.google.com/file/d/1XnNtEV7xZIBZIiTEyMcLUdgcvlhwFZ4P/view
- Additional References: https://www.afpacp.fr/voir_departements_limitrophes.php

Project Delivery

One or several script(s) / notebook(s) are expected to be delivered.

Code will be reviewed and evaluated with the comments in the script(s)/notebook(s). Make sure to document your code, it is very important for the evaluation! Adding relevant tests would be a plus.

No additional deliverable is expected.

Consider deliverying your script(s)/notebook(s) as a **Github project**.

More info are available on Github in the #sparkproject on Slack.

Appendix: computing distance between two points

```
def haversine(lon1, lat1, lon2, lat2, unit=1):
    """
    Calculate the great circle distance between two points
    on the earth (specified in decimal degrees)
    """
    # convert decimal degrees to radians
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

    # haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 * unit # Radius of earth in kilometers. Use 3956 for miles
    return c * r
```

Questions?

#sparkproject on Slack