Air Quality 🌨️

1. Introduction
2. Collect Data
3. Clean Data
4. Processing Data
5. Analysis
6. Conclusions

Introduction

Introduction

- Big Data : Powerful method to analyze for insights from the sporadic data.
- We aim to increase the knowledge on data analysis from this project

Project Subject

- Air quality of the lleida city between 2019 and 2020
- How the pandemic situation affect to air quality of Lleida city?
- Based on abundance in Microgram per Cubic Meter of Air (ug/m3) of **O_3, CO, NO_2, SO_2**

Introduction

Environment

- Python3
- Anaconda : python virtual environment
- Docker : spark context
- Jupyter Notebook , Python pyspark module
- Cookiecutter : project structure

Data Collecting

Source

- Open AQ API : non-profit organization empowering communities around the globe

Data collecting

- Building a python script “get_data.py” via HTTP request **OpenAQ API, JSON format**
- **Parameters.json :** units & types - **O_3, CO, NO_2, SO_2**
- **Locations.json :** sensor information in Lleida city
- **Measurement.json :** the obtained data from each sensor

Issues

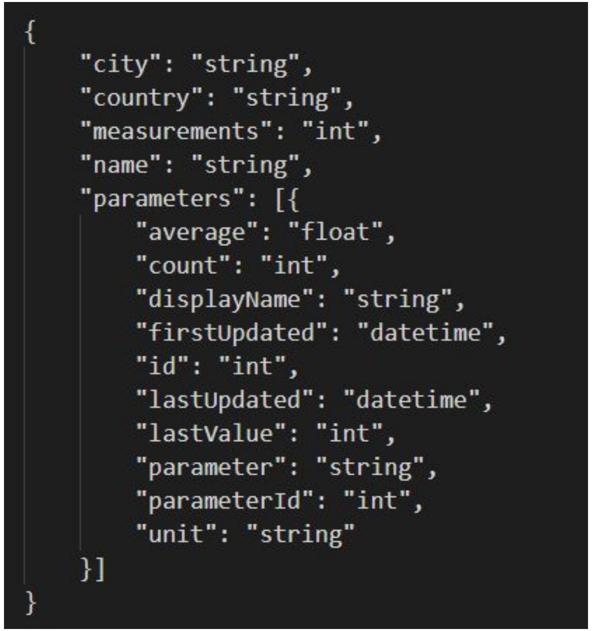
- Choosing Source : different models, categorization and formats by each data source
- Comparing the data of the city : limited computational power

Data Cleaning

- Jupyter notebook
- Pandas Profiling => Reports
- Remove useless data
- Remove negative values from the sensors => lack of information
- Re-design initial model

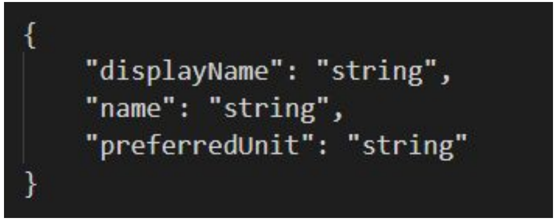
Data Cleaning

locations schema



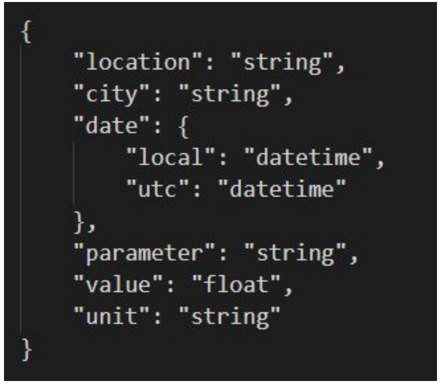
```
{
    "city": "string",
    "country": "string",
    "measurements": "int",
    "name": "string",
    "parameters": [{
        "average": "float",
        "count": "int",
        "displayName": "string",
        "firstUpdated": "datetime",
        "id": "int",
        "lastUpdated": "datetime",
        "lastValue": "int",
        "parameter": "string",
        "parameterId": "int",
        "unit": "string"
    }]
}
```

parameters schema



```
{
    "displayName": "string",
    "name": "string",
    "preferredUnit": "string"
}
```

measurements schema

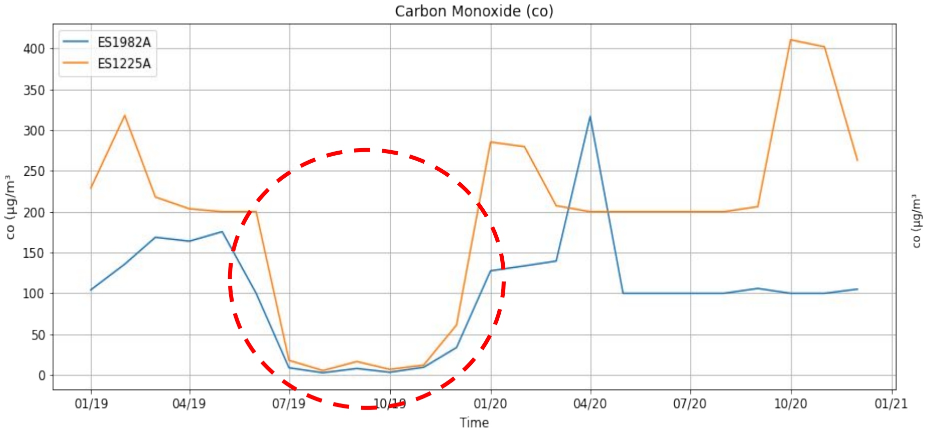


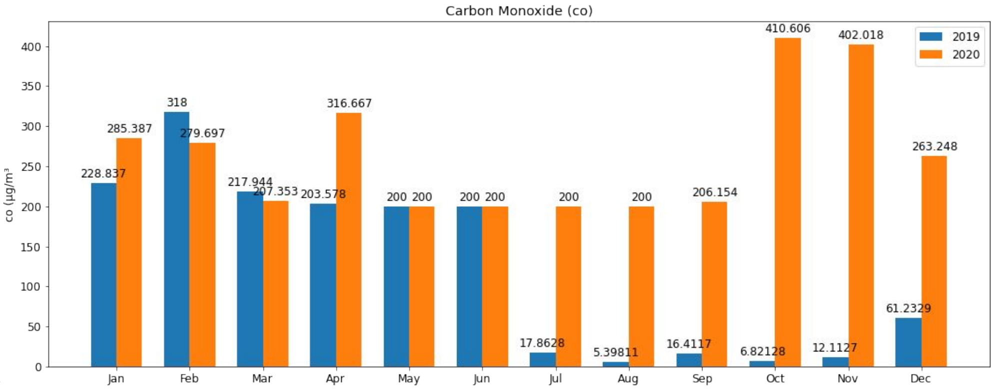
```
{
    "location": "string",
    "city": "string",
    "date": {
        "local": "datetime",
        "utc": "datetime"
    },
    "parameter": "string",
    "value": "float",
    "unit": "string"
}
```

Data Processing

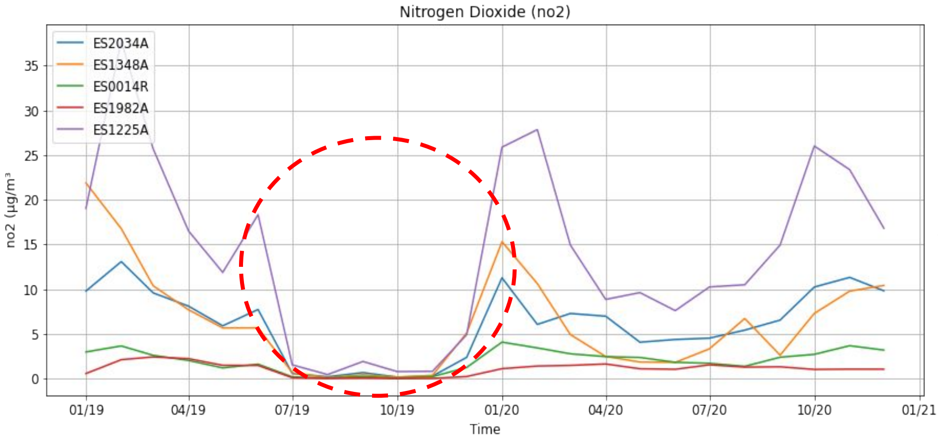
- Docker Image: **jupyter/all-spark-notebook -> pyspark + notebook + matplotlib**
- Group measurements by: **Sensor**
- Aggregate: Date by **month**
- **JOIN ALL**
- Parameters: **NO2, CO, SO2, O3**
- **! Issue ! - No available data** during the last six months of **2019**
- Air Sensors: **High volume of data + different precision + location**

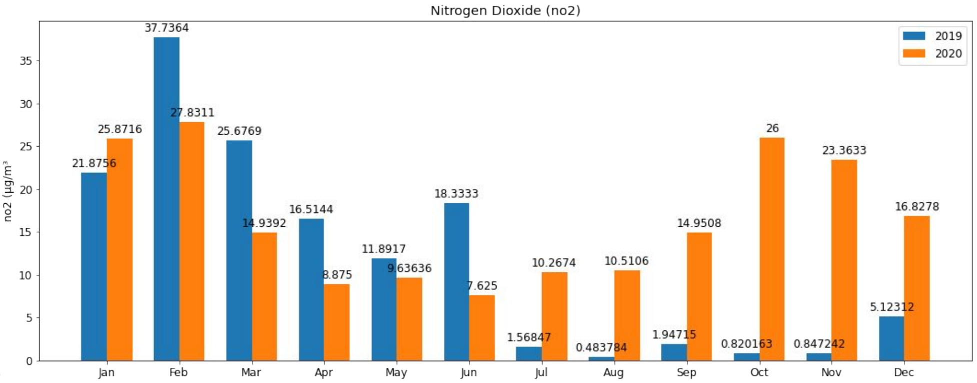
Analysis (CO)


Carbon Monoxide (co)
ES1982A
ES1225A
co (µg/m³)
0
50
100
150
200
250
300
350
400
01/19
04/19
07/19
10/19
01/20
04/20
07/20
10/20
01/21
Time

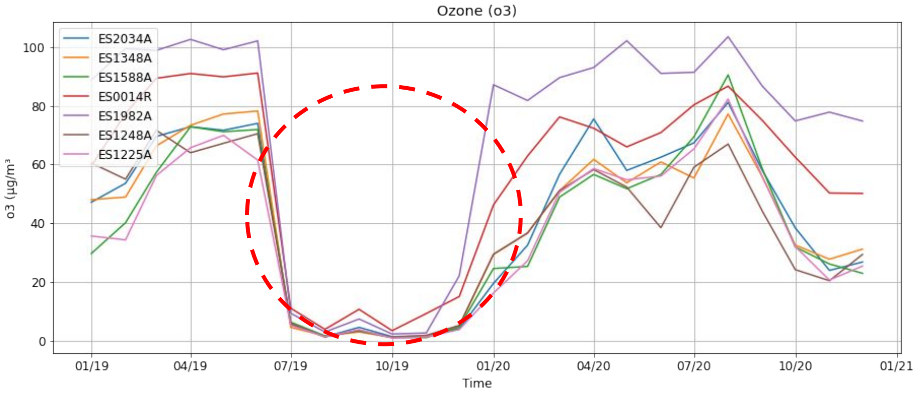

Carbon Monoxide (co)
2019
2020
co (µg/m³)
0
50
100
150
200
250
300
350
400
228.837
285.387
318
279.697
217.944
207.353
203.578
316.667
200
200
200
200
17.8628
200
5.39811
200
16.4117
206.154
6.82128
410.606
12.1127
402.018
61.2329
263.248
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

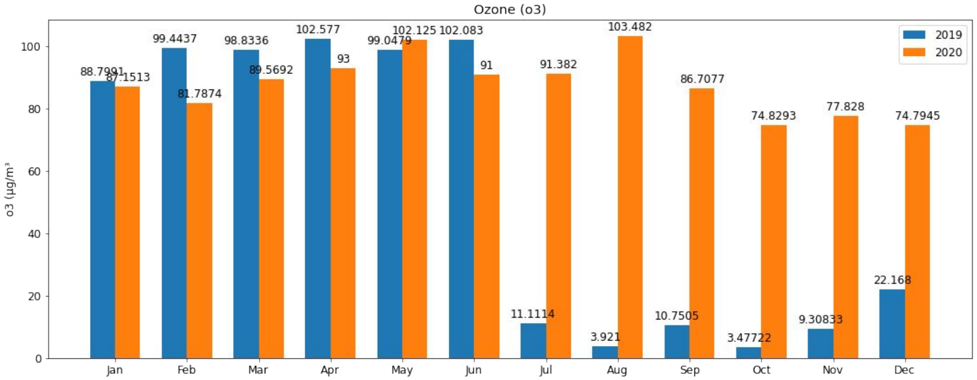
Analysis (NO2)



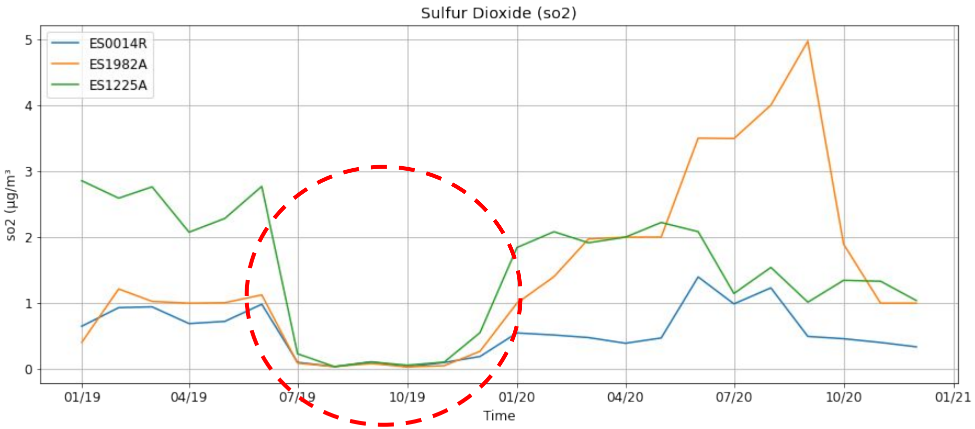


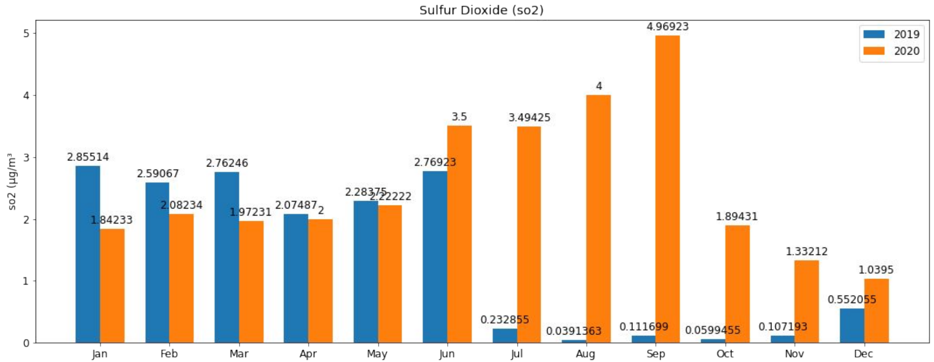
Analysis (O3)


Ozone (o3)
ES2034A
ES1348A
ES1588A
ES0014R
ES1982A
ES1248A
ES1225A
o3 (µg/m³)
0
20
40
60
80
100
01/19
04/19
07/19
10/19
01/20
04/20
07/20
10/20
01/21
Time


Ozone (o3)
2019
2020
o3 (µg/m³)
0
20
40
60
80
100
88.7991
87.1513
99.4437
81.7874
98.8336
89.5692
102.577
93
99.0479
102.125
102.083
91
11.1114
91.382
3.921
103.482
10.7505
86.7077
3.47722
74.8293
9.30833
77.828
22.168
74.7945
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Analysis (SO2)





Conclusions