Advanced Coding for Data Analytics (2025 / 2026)

Management and Computer Science – Luiss Guido Carli

Project — *Madrid Air Quality*

Teachers

- Andrea Coletta — acoletta@luiss.it
- Marco D'Elia — deliam@luiss.it

Abstract

Air quality is a major public-health concern. Exposure to pollutants is linked to several health risks, and cities require evidence-based tools to monitor and reduce harm.

In this project, students will build a complete analytics pipeline for Madrid air quality using the **METRAQ dataset** [1]. The project combines data engineering, data analysis, imputation, spatial and correlation networks, and visualization.

Goal

Use the dataset to understand:

- current pollution patterns in Madrid
- most affected areas
- relationships (correlation or potential causal links) with other variables (e.g., weather, traffic)

Suggested Workflow

Build a **reproducible pipeline** that:

1. Cleans and analyses large-scale time series
2. Compares imputation methods (your own vs METRAQ's `is_interpolated`)
3. Builds spatial and correlation networks
   3.1 Models pollutant propagation *(optional)*
4. Investigates which variables increase or reduce pollution
   4.1 Builds a forecasting model to estimate the impact of confounding variables *(optional)*
5. Produces clear visualizations of results

How to Work

You will work in groups of **3 students**.

The project covers most of the topics and tools discussed during lectures. Start early and ask questions. Please also report any issues or typos.

Project Data

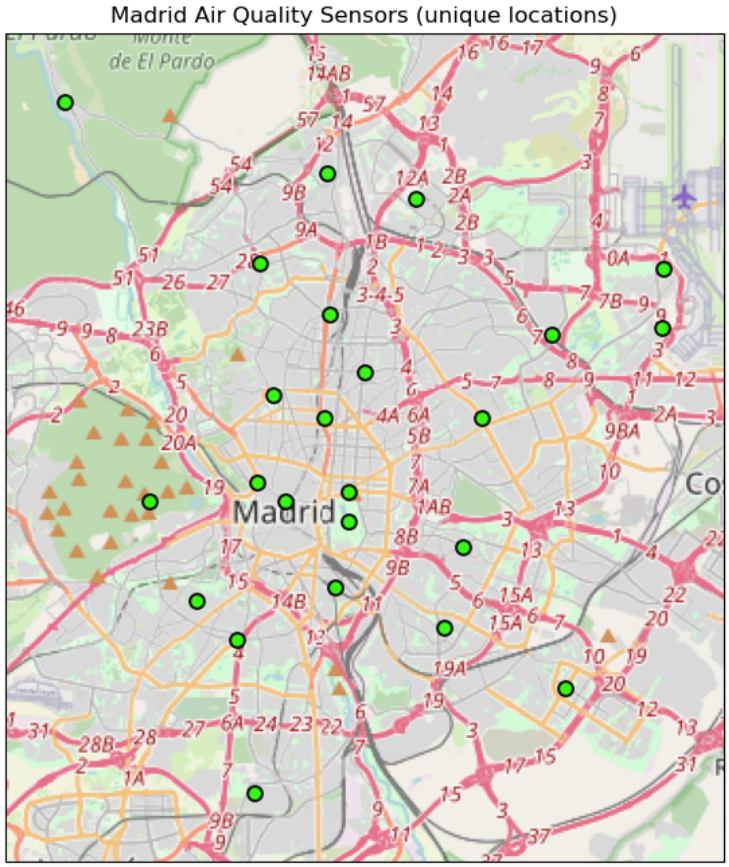
We use the **METRAQ Air Quality Dataset (Madrid)** [1], which contains hourly measurements from **January 2001 to December 2024**.

Available data

- Up to **14 pollutants**
- Up to **7 meteorological parameters**
- **3 traffic metrics** aggregated using **5 interpolation methods**

Note: Not all variables are available throughout the entire time span of the dataset; many were introduced in later years. Keep this in mind when designing your analysis. One possible approach is to restrict the dataset to a time period in which all selected variables are simultaneously available.

Sensor positions (Madrid):



Sensors are located across multiple areas of the city, for example: `Plaza de España`, `Escuelas Aguirre`, `Ramón y Cajal`, `Arturo Soria`, `Villaverde`, `Farolillo`, `Casa de Campo`, `Plaza del Carmen`, and others.

Each location may contain a different number of observations depending on sensor availability, operational periods, and measured variables.

Dataset size

Total rows: **~64M**

Category	Rows	Percentage
Air quality	~25M	39.36%
Meteorology	~9M	13.62%
Traffic	~30M	47.02%

Example of Records

	sensor_id	sensor_name	utm_x	utm_y	magnitude_id	magnitude_name	entry_date	value	is_interpolated
29982492	28079055	Urb. Embajada	450779	4.47924e+06	1014	SP_KRIGING	2018-10-25 17:00:00	35.3513	False
28252841	28079036	Moratalaz	445246	4.47324e+06	1013	SP_IDW	2018-04-14 07:00:00	6.78828	False
13941943	28079017	Villaverde	439421	4.46653e+06	1024	OC_KRIGING	2015-12-26 03:00:00	2.21673	False
55451754	28079017	Villaverde	439421	4.46653e+06	1010	SP_RBF_MULTIQUADRIQ	2023-04-09 01:00:00	2.24084	False
33635836	28079038	Cuatro Caminos	440033	4.47745e+06	30	BENCENO	2019-08-06 03:00:00	0	True

Thus, the dataset contains the following columns:

`sensor_id`, `sensor_name`, `utm_x`, `utm_y`, `magnitude_id`, `magnitude_name`, `entry_date`, `value`, `is_interpolated`

Where:

- **sensor_id** — Unique numerical identifier of the monitoring station
- **sensor_name** — Human-readable name of the monitoring station
- **utm_x, utm_y** — The coordinates of the station
- **magnitude_id** — Numerical code identifying the measured variable
- **magnitude_name** — Name of the measured variable
- **entry_date** — Timestamp of the measurement
- **value** — Observed value of the variable
- **is_interpolated** — Boolean flag indicating whether the value was reconstructed through interpolation (`True` = originally missing, `False` = real measurement)

Download Dataset

The full dataset can be downloaded from Hugging Face (please let us know if you encounter any issues):

- https://huggingface.co/datasets/dmariaa70/METRAQ-Air-Quality

What is Hugging Face?
Hugging Face is an online platform used to host and share datasets and machine learning models. In this case, it simply serves as a repository from which you can download the data files.

Sample Dataset

We also provide a smaller sample dataset (around 100,000 rows) that you can use to start your project.
This is useful to quickly explore the structure and develop your pipeline before working with the full dataset.

File included in this project:

- `sample_madrid_air_quality.csv`

Beware of Missing Values

As in most real-world datasets, several datapoints are missing. The dataset authors already provide an imputation solution based on interpolation. You can use the `is_interpolated` column to identify which values were originally missing and later filled.

Missingness may occur in different forms:

- Entire days or hours may be missing
- Only specific variables may be missing at a given timestamp
- Missing values may affect some sensors but not others

For example, at a given timestamp only **SO2** may be missing while all other variables are present:

	sensor_id	sensor_name	utm_x	utm_y	magnitude_id	magnitude_name	entry_date	value	is_interpolated
10247	28079004	Plaza de España	439579	4.47505e+06	7	NO	2001-03-27 09:00:00	98	False
10248	28079004	Plaza de España	439579	4.47505e+06	8	NO2	2001-03-27 09:00:00	63	False
10249	28079004	Plaza de España	439579	4.47505e+06	12	NOX	2001-03-27 09:00:00	213	False
10250	28079004	Plaza de España	439579	4.47505e+06	1	SO2	2001-03-27 10:00:00	15	True
10251	28079004	Plaza de España	439579	4.47505e+06	6	CO	2001-03-27 10:00:00	0	False
10252	28079004	Plaza de España	439579	4.47505e+06	7	NO	2001-03-27 10:00:00	42	False

Air Quality Variables

Air Pollutants

We have 14 differen pollutants: **CO**, **NO**, **NO2**, **NOX**, **SO2**, **<PM10**, **<PM2.5**, **O3**, **TOLUENO**, **BENCENO**, **ETILBENCENO**, **HIDROCARBS_TOTALES**, **METANO**, **HIDROCARBS_NO_METANICOS**. Target Value

Meteorological Variables

Variable	Short explanation

Variable	Short explanation
VV	Wind speed
DV	Wind direction
TEMP	Air temperature
HR	Relative humidity
PRE	Atmospheric pressure
RS	Solar radiation
PRECIPITACION	Rainfall

Traffic Variables

Estimated traffic conditions reconstructed through spatial interpolation methods.
They represent **human mobility**, not pollution directly.

Prefixes:

- **TI** → Traffic intensity (vehicle flow)
- **SP** → Average speed
- **OC** → Road occupancy / congestion

Each suffix indicates the interpolation method used among the five: **RBF_MULTICUADRIC / GAUSSIAN / LINEAR**, **IDW**, **KRIGING**.

Examples: *TI_RBF_MULTICUADRIC, TI_IDW, TI_RBF_GAUSSIAN, TI_RBF_LINEAR, TI_KRIGING*

Note:

This classification is our interpretation of the variables, as the dataset does not explicitly provide semantic for each acronym. If you notice inconsistencies in the data, please let us know.

If you are curious, you may also try applying unsupervised clustering methods to verify whether the variables naturally group into the three macro categories proposed.

Tasks

Each task provides general guidelines. While we expect you to address the main goal of each task with a reasonable and well-motivated analysis, you are free to follow your own methodology and are not required to strictly adhere to the suggested steps.

Task 1 — Load Data and Inspect Structure

Goal: Understand schema, scale, and time coverage.

Suggested steps:

- Load dataset efficiently
- Inspect schema and datatypes
- Identify pollutants, weather, traffic variables.
- Compute descriptive statistics
- Produce initial time-series and distribution plots

Task 2 — Missingness and Data Quality

Goal: Quantify missingness and identify data quality issues.

In the METRAQ dataset, missing values have already been filled through interpolation. Use the `is_interpolated` column to reconstruct where missingness originally occurred.

Suggested steps:

- Measure missingness per column and per pollutant
- Distinguish between temporal gaps (missing periods) and sensor-specific gaps
- Detect invalid or inconsistent values

Important: Invalid or inconsistent measurements may still be present regardless of the `is_interpolated` flag.

Task 3 — Imputation

Goal: Compare imputation strategies and evaluate their impact.

As observed in the previous task, several datapoints were originally missing in the dataset. The dataset authors provide an imputation solution based on interpolation — however, we believe you can improve upon it.

Implement **at least two imputation methods**, for example:

- Mean / median
- Forward / backward fill
- KNN / regression
- Any other reasonable approaches

Required comparison

1. Identify interpolated values:
   - `is_interpolated == True`
2. Reconstruct the original missingness
3. Apply your own imputation methods
4. Compare your results against the provided interpolation

How to compare values?
Hint: analyze and compare the empirical distributions (before/after imputation, across methods).

Finally, discuss strengths and weaknesses of the different imputation strategies.

Task 4 — Temporal Analysis

Goal: Identify patterns and pollution cycles.

Suggested steps:

- Aggregate per sensor/station
- Choose time granularity and justify it
- Detect seasonality

Discussion examples:

- Do pollutants have different cycles?
- Are trends stable across stations?

Task 5 — Spatial Network

Goal: Build a network based on sensor locations.

Sensor coordinates are provided in the columns **utm_x** and **utm_y**.

Suggested steps:

- Build a NetworkX graph where sensors are nodes
- Study and analyse the graph

We have to decide what really means to be connected or not, the intuition could be that they might have the same qualities, but if they don't it might not be connected

Discussion:
How should edges be defined? Distance is a natural starting point, but since all sensors are located within the same city, a naive distance-based rule may result in a fully connected network.

Question: What are the implications if the network becomes fully connected?

Consider alternative strategies for deciding whether to connect two sensors, for example:

- k-nearest neighbors
- distance thresholds
- other reasonable spatial rules

After selecting a connection method, study how structural properties change under different thresholds or distance definitions:

- Compare network properties (e.g., degree distribution, connected components)
- Analyze communities (e.g., via modularity optimization)

Task 6 — Correlation Network

Goal: Capture relationships between sensors based on pollutant time series.

In the previous task, the graph was built using geographic distance: sensors were connected because they were physically close. But are two areas necessarily related just because they are nearby?

Now we want to connect sensors that behave similarly over time. Two locations may experience similar pollution patterns due to wind, traffic flows, or shared environmental conditions, even if they are not geographically adjacent.

In this task, edges represent **similarity in behavior**, not proximity in space.

you connect two sensors if they have similar air qualities, and the network is more based on the behavior than just on the distance of the sensors

Suggested steps:

- Compute pairwise similarity (e.g., correlation) between sensors (hint: to simplify the analysis, you may aggregate the data to a *monthly frequency*)
- Build a graph by connecting sensors whose similarity exceeds a chosen threshold
- Analyze how the network changes with different thresholds and time windows

If you design a different rule to define edges, discuss how the resulting network structure changes under different assumptions.

Finally:

- Compare and discuss the *correlation-based network* against the *distance-based network*

Task 7 — Propagation Modeling (Optional)

This task is optional. Network propagation models are not covered in the lectures and may be challenging, but it is intended for students interested in this particular topic / problem. **We encourage you to first complete the main project tasks and then come back to this one.**

Goal: Study how pollutants may propagate and diffuse across the network. How bad air qualities propagate in the network

Suggested steps:

- Design and implement a propagation model
- Compare and validate your model against observed data

Task 8 — Parallelization

Now it's time to compute, for **each year** and **each sensor**, the hourly **correlation matrix** representing the relationships between variables.

Then summarize key observations, for example:

- correlations that are strong and stable across years.
- percentage of variable pairs above a chosen threshold (e.g., 0.6)
- investigate which variables are associated with increases or reductions in pollution

Understanding how much the CO2 could be correlated to the speed of cars, it takes half an hour and parallelize to speed up the computational time

Compare the runtime of:

- **Sequential execution**
- **Parallel execution**
  (If the sequential version is too slow to run end-to-end, provide a reasoned estimate of the expected runtime based on partial runs.)

Suggested:

- Parallelize across **years** and/or **sensors**
- Each parallel worker should independently read the dataset --- load only the relevant subset of data needed --- and save the correlation matrix.

Finally, discuss scalability and the practical limits of your solution (e.g., CPU cores, memory, and I/O constraints).

Task 9 — Forecasting Model *(Optional)*

This task is optional. Forecasting and causal interpretation go beyond the core requirements of the project and may require additional modeling choices. We encourage you to first complete the main project tasks and then come back to this one.

Goal: Estimate the impact of confounding variables on pollution levels.

Suggested steps:

- Select one or more pollutants as prediction targets
- Build a forecasting or regression model using meteorological and/or traffic variables
- Evaluate predictive performance
- Interpret how different variables influence pollution levels

Discuss assumptions and limitations of your approach.

Task 10 — Final Visualization

Communicate your results clearly.

Suggested visualization deliverables:

- Structured plots
- Graph visualizations or map of sensors
- Time-series views
- Correlation views

Note: *Dynamic plots or a Streamlit app are optional.*

Final Deliverables

- Python code (reproducibility is mandatory!)
- `README.md` (describe how to run your code)
- `requirements.txt`
- Presentation deck (~10 minutes, 8 to 15 slides)
- Optional appendix (≤1000 words, not graded)

Optional: invite us to a github (private) project to share your coding experience!

Note: Keep in mind that the goal of this project is not only to write clean and efficient code, but also to extract and communicate meaningful insights from a large air-quality dataset using the tools and methods covered in the course.

Evaluation Criteria

Criterion
Code structure & design
Implemented Tasks
Presentation

Notes:

- You do **not** need to complete all tasks to succeed
- All the mandatory tasks already give the maximum score
- Optional tasks just help recover mistakes or obtain honors

AI Policy

This is a learning exercise, so we discourage reliance on AI coding tools.

In general, you remain responsible for:

- understanding the code
- validating outputs
- explaining decisions

And thus we assume you fully understand the implementation you submit.

Final notes.

The project is **mandatory** and accounts for **70% of the final grade**. You are expected to work in groups of **three students**.

We will share a form where you must register your group and indicate your teammates.

Submit your project by email to both instructors (Coletta and D'Elia) as a `.zip` file containing the presentation/slides and the code.

Please use the subject line:
"Adv.Coding2026: group X", where *X* is the group ID assigned by the instructors.

Submission Deadline:

You may submit your project at any time, but no later than **one day before** the scheduled presentation or written test — whichever comes first — of the exam session in which you intend to finish the course.

Some examples

- You want to complete the course in the first May session and the written exam is on May 18th.
  If you want to present the project during the last lecture (May 5th), you must submit it by **May 4th**.
  If instead you want to present the project on the afternoon of May 20, the project deadline is **May 17**.
- You are not ready to submit the project for the first May session, but still want to take the written exam in May. You may take the written exam and present the project in a later session (e.g., June). In that case, the project must be submitted the day before the presentation in that session.

For any doubts, write to us.

Dataset License and Citation

[1] - David María-Arribas et al. "METRAQ Air Quality dataset." https://huggingface.co/datasets/dmariaa70/METRAQ-Air-Quality