

CASE STUDY: Dự đoán chất lượng không khí theo giờ cho Beijing

ThS. Lê Thị Thùy Trang

2025-12-21

1 Dự đoán chất lượng không khí dựa trên dữ liệu

1.1 Giới thiệu dự án

Trong bối cảnh đô thị hoá nhanh và biến đổi khí hậu ngày càng rõ rệt, việc đánh giá và dự đoán chất lượng không khí (đặc biệt là nồng độ PM2.5) đã trở thành nhu cầu thiết yếu, không chỉ cho cơ quan quản lý mà còn cho y tế cộng đồng và người dân. Nếu chỉ “nhìn số liệu hiện tại” hoặc áp dụng một ngưỡng cảnh báo chung cho mọi thời điểm, ta rất dễ bỏ lỡ các đợt ô nhiễm tăng vọt theo mùa vụ, thời tiết, giờ cao điểm, hoặc các sự kiện bất thường. Đây là lúc bài toán dự đoán PM2.5 bằng hồi quy và chuỗi thời gian trở nên cấp thiết.

Dự đoán PM2.5 là quá trình mô hình hoá mối quan hệ giữa PM2.5 và các yếu tố liên quan (nhiệt độ, độ ẩm, gió, áp suất, giờ trong ngày, ngày trong tuần, mùa...), đồng thời khai thác tính phụ thuộc theo thời gian (xu hướng, chu kỳ, mùa vụ) của chính chuỗi PM2.5. Mục tiêu cốt lõi của việc này bao gồm:

- Hỗ trợ ra quyết định và cảnh báo sớm: dự báo trước PM2.5 trong vài giờ/ngày tới để phát cảnh báo sức khoẻ, khuyến nghị hạn chế hoạt động ngoài trời, hoặc điều chỉnh vận hành (trường học, bệnh viện, giao thông).
- Hiểu nguyên nhân và tác động của các yếu tố môi trường: với hồi quy, ta định lượng mức ảnh hưởng của biến thời tiết/khung giờ/mùa vụ lên PM2.5 (ví dụ: độ ẩm tăng có làm PM2.5 tăng/giảm, gió mạnh có giúp khuếch tán không).
- Lập kế hoạch và phân bổ nguồn lực: chủ động bố trí lực lượng quan trắc, kiểm soát phát thải, và tối ưu lịch kiểm tra/khuyến nghị dựa trên các giai đoạn có nguy cơ ô nhiễm cao (mùa đông, giờ cao điểm, ngày ít gió, v.v.).

Trong dự án **này**, chúng ta đóng vai trò Data Scientist xây dựng pipeline dự báo PM2.5 theo giờ từ dữ liệu chất lượng không khí Beijing (2013–2017). Mục tiêu không chỉ là “chạy đúng mô hình”, mà còn phải phân tích đặc điểm chuỗi thời gian để ra quyết định chọn ARIMA (p, d, q).

Dự án này được thiết kế dành cho các kỹ sư dữ liệu và nhà khoa học dữ liệu tương lai. Dự án tuân thủ triết lý lập trình hướng đối tượng thay vì tiếp cận vấn đề theo notebook rời rạc. Các bước thực hiện bao gồm:

1. Khám phá dữ liệu: Hiểu cấu trúc và phân bố của tập dữ liệu.
2. Làm sạch và chuẩn hoá thời gian.
3. Chuẩn bị dữ liệu cho dự báo
4. Đánh giá mô hình (Regression baseline)
5. Dự báo chuỗi thời gian bằng ARIMA.

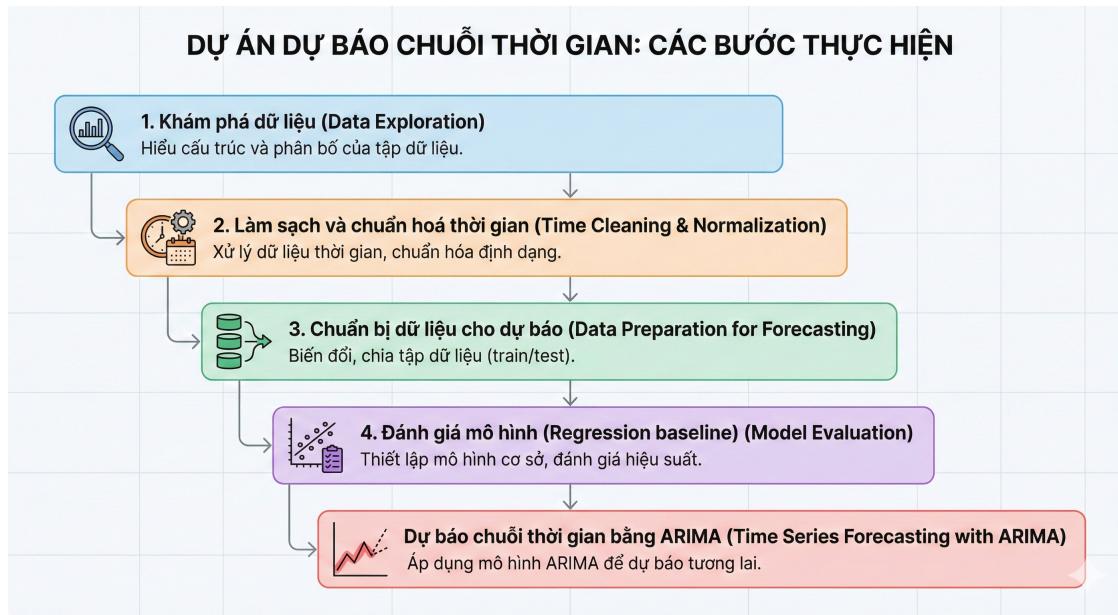


Figure 1: Pipeline project

1.2 Danh sách các module trong project

Để đảm bảo tính chuyên nghiệp, dự án này áp dụng tư duy lập trình hướng đối tượng (OOP). Thay vì viết hàng trăm dòng script xử lý rối rắm vào chính những file notebook mà chúng ta trình bày, chúng ta đóng gói logic vào các lớp (Class) trong một file riêng. Sau đó, khi làm việc với notebook ta có, ta sẽ sử dụng hàm import để sử dụng các lớp nói trên. Điều này giúp notebook sạch sẽ, tập trung vào kết quả và dễ dàng tái sử dụng.

Thành phần	Ý nghĩa và chức năng
src/classification_library.py	Chứa các hàm dùng chung: load dữ liệu từ UCI/ZIP, clean dữ liệu, tạo datetime, thêm time features, lag features... (cốt lõi cho pipeline dữ liệu).
src/regression_library.py	Chuẩn bị dataset cho hồi quy dự báo, train/test split theo thời gian, train regressor và tính metrics.
src/timeseries_library.py	Các hàm phân tích chuỗi thời gian (stationarity tests, ACF/PACF, grid search ARIMA, forecast + đánh giá)
notebooks/preprocessing_and_eda.ipynb	Notebook bước 1: Thực hiện tải dữ liệu thô, xử lý các giá trị thiếu, lọc dữ liệu và thực hiện phân tích khám phá dữ liệu (EDA) để hiểu tổng quan về tập dữ liệu.
notebooks/regression_modelling.ipynb	Notebook chuẩn bị dataset hồi quy dự báo PM2.5(t+h) và chạy baseline regression + metrics.
notebooks/arima_forecasting.ipynb	Notebook bước 3: Notebook này phân tích chuỗi thời gian + chọn (p,d,q) + huấn luyện ARIMA + forecast + đánh giá.
run_papermill.py	Script điều phối toàn bộ pipeline: sử dụng thư viện papermill để tự động chạy lần lượt các notebook với các tham số cấu hình; lưu lại các notebook sau khi thực thi, ghi nhận thời gian chạy (execution time) và giúp bạn có thể tái chạy toàn bộ quy trình chỉ bằng một lệnh từ dòng lệnh.
data/	Thư mục chứa dữ liệu đầu vào và lưu trữ các dữ liệu trung gian đã qua xử lý (processed data) để sử dụng giữa các bước phân tích.

Table 1: Bảng thành phần Project

1.3 Giới thiệu Papermill: Tự động hoá pipeline Notebook

Trong dự án này, chúng ta sử dụng thư viện papermill để điều phối việc thực thi tuần tự các bước: tiền xử lý (*preprocessing*), chuẩn bị giỏ hàng (*basket preparation*) và mô hình hoá luật kết hợp (*Apriori modelling*).

Thay vì mở từng notebook và chạy thủ công nhiều lần, papermill cho phép:

- Tự động chạy toàn bộ pipeline chỉ bằng một lệnh.
- Ghi lại kết quả vào các notebook đã thực thi.
- Thay đổi tham số đầu vào mà không chỉnh lại mã nguồn.
- Đảm bảo tính lặp lại (*reproducibility*) trong phân tích.

Cách làm này phản ánh tư duy của các hệ thống phân tích dữ liệu chuyên nghiệp:

- Notebook không chỉ để trình bày mà còn là thành phần của pipeline.
- Tách biệt *logic phân tích* (nằm trong thư viện) khỏi *quy trình thực thi*.
- Giảm thiểu rủi ro thao tác thủ công và nhầm lẫn khi tái chạy.

Sinh viên không cần hiểu sâu về cơ chế nội bộ của papermill, nhưng cần nắm ý nghĩa của việc tự động hoá pipeline: đây là một bước tiến từ tư duy phân tích *mang tính trình diễn* sang tư duy *kỹ sư dữ liệu*, nơi các quy trình được chuẩn hoá, tái lập và dễ chuyển giao.

1.4 Chuẩn bị môi trường thực hành

Kích hoạt môi trường ảo

```
conda activate beijing_env
```

Toàn bộ các thư viện cần thiết cho dự án được liệt kê trong file `requirements.txt`. Vì vậy, sau khi kích hoạt môi trường, cài đặt các thư viện cần thiết bằng câu lệnh:

```
pip install -r requirements.txt
```

2 Nhắc lại kiến thức: Chuỗi thời gian & ARIMA

2.1 Chuỗi thời gian có gì khác so với dữ liệu thường?

Dữ liệu chuỗi thời gian khác dữ liệu “thường” ở chỗ thời gian là một phần của cấu trúc dữ liệu, không chỉ là một cột thông tin, mà có các tính chất sau đây:

- Có thứ tự và phụ thuộc theo thời gian: điểm dữ liệu hôm nay thường liên quan đến hôm qua/hôm trước, vì vậy không thể shuffle tự nhiên như dữ liệu i.i.d.
- Mang các “mẫu theo thời gian” đặc trưng: xu hướng (trend) tăng/giảm dài hạn, mùa vụ (seasonality) lặp lại theo chu kỳ (ngày/tuần/tháng/năm), và tự tương quan (autocorrelation) – giá trị hiện tại thường có liên hệ với các giá trị quá khứ.
- Cách chia train/test phải tôn trọng dòng thời gian: nếu chia sai (tron̄n̄ tương lai vào quá khứ), mô hình sẽ “nhìn trộm tương lai”, gây leakage theo thời gian; kết quả đánh giá có thể đẹp giả nhưng dự báo thực tế lại kém.

2.2 Mục tiêu dự báo của dự án

- Dự báo pm2.5 tại thời điểm $t + \text{horizon}$ (mặc định horizon = 1 giờ).
- Đánh giá trên tập test sau mốc cutoff (ví dụ: 2017-01-01).

2.3 Stationarity và quyết định tham số d

- Trong arima, giả định nền tảng là chuỗi (tương đối) dừng, tức là các đặc trưng thống kê như trung bình và phương sai không thay đổi quá mạnh theo thời gian.
- Nếu chuỗi không dừng, ta cần thực hiện sai phân (differencing) để “loại bỏ” xu hướng hoặc biến động dài hạn; mỗi lần sai phân tương ứng với việc tăng tham số d .
- Nếu chuỗi đã dừng ngay từ đầu (hoặc trở nên dừng sau khi biến đổi phù hợp), ta có thể chọn $d = 0$ để tránh sai phân quá mức.

Quy trình ra quyết định:

- Quan sát trực quan: vẽ chuỗi gốc và chèn thêm rolling mean, rolling std để xem trung bình và độ biến động có ổn định theo thời gian hay không.
- Kiểm định thống kê để củng cố kết luận:
 - adf: nếu p nhỏ, ta bác bỏ giả thuyết “có unit root”, từ đó ủng hộ nhận định chuỗi có xu hướng dừng.
 - kpss: nếu p lớn, ta không bác bỏ giả thuyết “đừng”, tức là dữ liệu phù hợp với giả định dừng.

2.4 acf/pacf và quyết định p, q

- Sau khi đã chọn được d (đảm bảo chuỗi đủ gần dừng), bước tiếp theo là quyết định hai thành phần còn lại của arima: p cho phần tự hồi quy (ar) và q cho phần trung bình trượt (ma).
- pacf thường được dùng để gợi ý p : nếu pacf “cắt” rõ sau một vài độ trễ (lag), đó là tín hiệu số bậc ar có thể nằm quanh vị trí cắt này.
- acf thường được dùng để gợi ý q : tương tự, nếu acf giảm mạnh và gần như triệt tiêu sau một số lag nhất định, ta có thể cân nhắc q quanh vùng đó.
- Tuy nhiên, trong dữ liệu thực tế, tín hiệu từ acf/pacf có thể không hoàn toàn sắc nét (do nhiễu, mùa vụ, hoặc cấu trúc phức tạp). Vì vậy, cách chắc ăn là thử một lối nhỏ các giá trị ứng viên và chọn mô hình tốt nhất theo tiêu chí aic/bic để chốt bộ tham số (p, d, q).

2.5 Đánh giá dự báo

Vì sao phải đánh giá dự báo?

- Trong bài toán dự báo, “mục tiêu cuối cùng” không phải là vẽ được đường dự báo đẹp, mà là dự báo có hữu ích khi đưa vào sử dụng thực tế. Vì vậy, đánh giá dự báo là bước neo (anchor) để mọi quyết định về xử lý dữ liệu, chọn mô hình, và chỉnh tham số đều có căn cứ.
- Nếu chưa có một quy trình đánh giá đúng (tách train/test theo thời gian, chọn thước đo phù hợp), ta rất dễ rơi vào hai sai lầm phổ biến: (i) kết quả đẹp giả do leakage theo thời gian, hoặc (ii) tối ưu sai mục tiêu, ví dụ mô hình “bám đường” tốt khi nhìn hình nhưng sai số thực tế lại lớn.
- Đánh giá trước giúp trả lời các câu hỏi quan trọng: mô hình có dự báo tốt hơn baseline không, sai số có chấp nhận được không, mô hình có ổn định qua các giai đoạn khác nhau không, và mức rủi ro khi gặp các đỉnh ô nhiễm cao

là bao nhiêu. Chỉ khi những câu hỏi này được trả lời rõ ràng, ta mới có lý do hợp lý để đi tiếp vào phần tối ưu hoá và ra quyết định mô hình.

Các thước đo đánh giá dự báo hay dùng:

- mae (sai số tuyệt đối trung bình): dễ hiểu, ít nhạy với ngoại lai hơn rmse; phản ánh “trung bình mỗi lần dự báo lệch bao nhiêu đơn vị”.
- rmse (căn bậc hai sai số bình phương trung bình): phạt mạnh các sai số lớn, phù hợp khi ta muốn mô hình tránh những lần dự báo lệch quá nhiều (đặc biệt quan trọng với các định pm2.5).
- (tùy chọn) smape/mape: hữu ích khi cần nhìn sai số theo tỷ lệ phần trăm, nhưng phải cẩn thận khi giá trị thực gần 0 vì tỷ lệ có thể bùng nổ và gây hiểu sai chất lượng mô hình.

3 Phân tích và làm sạch dữ liệu

3.1 Tổng quan về tập dữ liệu

1) Tập dữ liệu sử dụng

Tập dữ liệu dùng trong case study này là *beijing multi-site air quality data (prsa)*. Dữ liệu gồm các phép đo theo giờ về chất lượng không khí và khí tượng tại 12 trạm quan trắc ở bắc kinh trong giai đoạn 2013–2017. Mỗi trạm được lưu thành một file .csv riêng, và toàn bộ được đóng gói trong một file zip:

`data/raw/PRSA2017_Data_20130301-20170228.zip`

Trong dự án, ta sẽ đọc trực tiếp từ file zip, sau đó hợp nhất (merge(concat) các file theo trạm thành một bảng lớn để làm việc theo đúng cấu trúc dữ liệu chuỗi thời gian.

2) Đơn vị quan sát, tần suất và phạm vi thời gian

- Đơn vị quan sát (1 dòng): một phép đo theo giờ tại một trạm.
- Tần suất: hourly (mỗi giờ có một bản ghi cho mỗi trạm).
- Phạm vi thời gian: từ 2013-03-01 00:00:00 đến 2017-02-28 23:00:00.
- Số giờ mỗi trạm: 35,064 dòng (tương ứng đúng số giờ trong khoảng thời gian trên).
- Tổng số dòng toàn bộ dữ liệu: 420,768 dòng ($= 12 \times 35,064$).

Đây là một bộ dữ liệu chuỗi thời gian theo giờ khá “chuẩn” vì khung thời gian đầy đủ và đều đặn. Tuy vậy, một số biến đo vẫn bị thiếu (missing) ở các thời điểm nhất định, rất phù hợp để sinh viên thực hành eda chuỗi thời gian và xử lý dữ liệu thiếu trước khi dự báo.

3) Danh sách cột và ý nghĩa

Mỗi file csv của từng trạm có cùng cấu trúc với 18 cột. có thể hiểu nhanh theo các nhóm sau:

- chỉ mục:
 - no: số thứ tự dòng trong file.
- thời gian:
 - year, month, day, hour: thông tin thời gian theo giờ (sẽ được ghép thành một cột datetime).
- ô nhiễm:

- pm2.5, pm10, so2, no2, co, o3: nồng độ các chất ô nhiễm (mục tiêu chính của lab là pm2.5).
- khí tượng:
 - temp, pres, dewp, rain, wspm: nhiệt độ, áp suất, điểm sương, lượng mưa, tốc độ gió.
- gió và trạm:
 - wd: hướng gió (biến phân loại).
 - station: tên trạm (12 giá trị khác nhau).

4) Độ thiếu dữ liệu (missingness) – vì sao cần làm sạch

Khi gộp dữ liệu của 12 trạm, một số biến ô nhiễm có tỷ lệ thiếu đáng kể hơn nhóm khí tượng. Tỷ lệ missing (xấp xỉ) thường gặp:

- co: ~ 4.92%
 - o3: ~ 3.16%
 - no2: ~ 2.88%
 - so2: ~ 2.14%
 - pm2.5: ~ 2.08%
 - pm10: ~ 1.53%
 - wd: ~ 0.43%
 - Nhóm khí tượng (temp, pres, dewp, rain, wspm): dưới ~ 0.10%
- Riêng pm2.5, tỷ lệ thiếu có thể khác nhau theo trạm (dao động khoảng ~ 1.09% đến ~ 2.72%). Ví dụ:
- cao hơn: huairou (~ 2.72%), aotizhongxin (~ 2.64%)
 - thấp hơn: wanliu (~ 1.09%)

5) Phân phối pm2.5

Khi nhìn vào phân phối pm2.5 (gộp 12 trạm và bỏ các giá trị na), ta thường thấy chuỗi có đuôi phải dài và đôi lúc xuất hiện các đỉnh rất cao (spike). Thông kê mô tả thường gặp:

- số mẫu hợp lệ: 412,029
- mean: ~ 79.79
- median (50%): 55
- q1–q3: 20 → 111
- 95th percentile: 242
- max: 999 (rất hiếm; có thể là spike/outlier hoặc giới hạn cảm biến)

Ý nghĩa: vì có spike và đuôi phải dài, lựa chọn thước đo đánh giá cũng cần cân nhắc, rmse sẽ nhạy với các sai số lớn (đặc biệt khi dự báo lệch ở những đỉnh ô nhiễm cao), trong khi mae phản ánh mức sai lệch “trung bình” ổn định hơn. Đây là điểm rất phù hợp để sinh viên thảo luận và ra quyết định theo đúng

3.2 Hiểu dữ liệu chuỗi thời gian - preprocessing_and_eda.ipynb

Mục tiêu của notebook:

- Đọc dữ liệu từ file zip (hoặc từ uci nếu được cấu hình).
- Làm sạch dữ liệu và tạo cột thời gian dạng datetime để sắp xếp, lọc, và vẽ theo trục thời gian.
- Phân tích dữ liệu để trả lời các câu hỏi cốt lõi:
 - (1) chuỗi có thiếu dữ liệu không?
 - (2) có xu hướng hoặc mùa vụ không?
 - (3) có tự tương quan không?
 - (4) mức độ dừng (stationarity) như thế nào?

Tham số quan trọng: sinh viên cần xem cell parameters và hiểu ý nghĩa các biến cấu hình:

- `use_ucimlrepo = false`
- `raw_zip_path = `data/raw/PRSA2017_Data_20130301-20170228.zip``
- `lag_hours = [1, 3, 24]` (dùng để tạo đặc trưng lag ở các notebook sau)
- output cleaned: `data/processed/cleaned.parquet`

Lưu ý lỗi hay gặp: nếu không đặt đúng file zip trong `data/raw/` thì notebook sẽ báo `zip not found`. vì vậy, trước khi chạy cần kiểm tra đúng tên file và đúng đường dẫn.

3.3 Baseline hồi quy cho dự báo - regression_modelling.ipynb

Mục tiêu của notebook:

- tạo bộ dữ liệu dự báo bằng cách dịch mục tiêu theo horizon: `pm2.5_target = pm2.5(t + horizon)`.
- sinh đặc trưng lag (1, 3, 24 giờ) và các đặc trưng thời gian (giờ, ngày, tháng, ...).
- chia train/test theo mốc cutoff để bảo toàn thứ tự thời gian và tránh leakage.
- huấn luyện một mô hình hồi quy baseline, sau đó lưu lại kết quả đánh giá (metrics) để làm mốc so sánh.

3.4 Dự báo ARIMA - arima_forecasting.ipynb

Mục tiêu của notebook:

- Chuẩn bị chuỗi pm2.5 theo thời gian và đảm bảo tần suất theo giờ (resample nếu cần).
- Phân tích tính dừng để quyết định tham số sai phân d .
- Dùng acf/pacf để gợi ý vùng giá trị p, q , sau đó thử một lối nhỏ để chốt mô hình.
- `fit arima` và dự báo 1-step ahead (`horizon = 1`) trên tập test.
- Đánh giá mae/rmse và vẽ so sánh forecast với actual để nhìn trực quan chất lượng dự báo.

4 Thực hành

Sinh viên thực hiện các question sau và viết trong blog, sau đó chuẩn bị slide trình bày cho hoạt động FIT-DNU CONQUER:

- Q1:** Sinh viên kiểm tra lại xem code đã đầy đủ các thành phần để hiểu dữ liệu này chưa? Nếu chưa thì bổ sung:
- Kiểm tra khoảng thời gian dữ liệu phủ (start/end) và xác nhận tần suất theo giờ là liên tục.
 - Tính tỷ lệ thiếu theo từng biến, đồng thời quan sát thiếu theo thời gian (thiếu tập trung vào giai đoạn nào hay rải đều).
 - Dùng boxplot hoặc quantile để nhìn nhanh ngoại lai (outliers) và phân phối lệch.
 - Vẽ chuỗi pm2.5 theo thời gian: một đồ thị toàn giai đoạn và một đồ thị phóng to 1–2 tháng để nhìn rõ dao động.
 - Kiểm tra tự tương quan đơn giản: so sánh tương quan của pm2.5 với các độ trễ như 24h và 168h để gợi ý chu kỳ ngày/tuần.
 - Kiểm tra tính dừng: chạy adf/kpss và viết nhận xét ngắn gọn về kết luận.

Đồng thời, giải thích “*thiếu theo biến nào là đáng lo nhất cho dự báo pm2.5 và vì sao?*”

Q2: Đối với baseline hồi quy cho dự báo, sinh viên cần giải thích được:

- Vì sao lag 24h thường có ý nghĩa: do nhịp sinh hoạt theo ngày và các điều kiện khí tượng lặp lại theo chu kỳ ngày.
- Vì sao phải chia theo thời gian bằng cutoff: để mô hình không “nhìn thấy tương lai”, tránh đánh giá đẹp giả do leakage.
- Phân biệt rmse và mae: khi nào rmse cao hơn nhiều? thường xảy ra khi dữ liệu có spike hoặc dự báo sai mạnh ở một số thời điểm, vì rmse phạt nặng sai số lớn.

Q3: Đối với model dự báo bằng ARIMA, sinh viên cần giải thích được: quy trình ra quyết định arima mà sinh viên phải viết trong báo cáo

- Quan sát chuỗi gốc để nhận diện xu hướng và mùa vụ (nếu có).
- Kiểm định dừng bằng adf/kpss để chọn d .
- Xem acf/pacf để đoán các giá trị ứng viên cho p và q .
- grid search với p, q nhỏ (giữ d cố định) và chọn mô hình theo aic/bic.
- Chẩn đoán phần dư: kiểm tra residual có gần “white noise” hay không, nhằm đánh giá mô hình đã bắt được cấu trúc chính của chuỗi chưa.

5 Hướng dẫn thực hiện hoạt động FIT-DNU CONQUER

5.1 Mục tiêu của hoạt động

Hoạt động FIT-DNU CONQUER được thiết kế nhằm giúp sinh viên:

- Hiểu sâu bản chất của hồi quy và chuỗi thời gian thông qua việc ứng dụng trên dữ liệu thực.
- Tập diễn giải kết quả bằng ngôn ngữ đơn giản (Feynman style), tránh chỉ mô tả code hoặc trích xuất số liệu.
- Rèn luyện khả năng trình bày, giải thích và thuyết phục thông qua hoạt động chia sẻ kết quả.
- Phát triển tư duy phân tích theo góc nhìn kinh doanh và đưa ra đề xuất hành động.

Sinh viên cần xem xét dự án như một nhiệm vụ Data Scientist thực thụ: không chỉ “chạy đúng thuật toán”, mà phải chuyển dữ liệu thành tri thức hữu ích.

5.2 Yêu cầu

Đối với mỗi nhóm (3-4 sinh viên), thực hiện thiết lập và chạy pipeline (EDA → Regression → ARIMA, như sau:

- Cài đặt môi trường và chạy pipeline đầy đủ bằng `run_papermill.py` (hoặc chạy tuần tự các notebook).
- Đảm bảo tạo ra đầy đủ artifacts trong `data/processed/` và notebook output trong `notebooks/runs/`.
- Ghi rõ cấu hình tham số đã dùng: `CUTOFF`, `LAG_HOURS`, `HORIZON`, trạm dùng cho ARIMA (STATION).

Thực hiện đầy đủ 3 câu hỏi trong phần thực hành.

1. Trực quan hóa và diễn giải (bắt buộc):

- Tối thiểu **04 hình** gồm: (1) PM2.5 toàn giai đoạn, (2) PM2.5 zoom 1–2 tháng, (3) ACF/PACF (hoặc 2 hình tách), (4) Forecast vs Actual (ARIMA).
- Mỗi hình phải có 2–4 câu diễn giải: “Nhìn hình này kết luận gì?” (không chỉ mô tả hình).

2. Insight và khuyến nghị (bắt buộc):

- Tối thiểu **05 insight chất lượng** dựa trên EDA + kết quả dự báo.
- Mỗi insight phải trả lời câu hỏi: “Nếu là người quản lý môi trường/đô thị, tôi có thể làm gì?” hoặc “Nếu triển khai hệ thống cảnh báo sớm, tôi cần chú ý điều gì?”

3. Báo cáo dạng blog (bắt buộc):

- Trình bày logic rõ ràng, súc tích; có tiêu đề, mục lục ngắn, kết luận.
- Không dump code, không copy toàn bộ bảng số liệu; chỉ trích đoạn cần thiết kèm diễn giải.
- Ngôn ngữ đơn giản, hướng tới người đọc không chuyên; giải thích thuật ngữ khi xuất hiện lần đầu.

4. Trình bày và chia sẻ kết quả (bắt buộc):

- Thời lượng trình bày: 5–7 phút/nhóm.
- Không đọc code; dùng Feynman style: giải thích như đang dạy một người khác.
- Phải trả lời 3 câu trong phần Q1–Q3 bằng lời (không chỉ để trong báo cáo).

5.3 Các chủ đề gợi ý phát triển từ bài mẫu

Mỗi nhóm sẽ chọn một trong các chủ đề dưới đây để triển khai và trình bày.

5.3.1 Chủ đề 1: Regression vs ARIMA – khi nào chọn cái nào?

Trong chủ đề này, mỗi nhóm giữ nguyên pipeline hiện tại và chỉ so sánh hai hướng dự báo đã có: baseline hồi quy (dùng time features và lag features) và ARIMA (mô hình chuỗi thời gian đơn biến). Yêu cầu quan trọng nhất là so sánh phải công bằng:

- Cùng một trạm (ví dụ `Aotizhongxin`).
- Cùng mốc chia train/test theo thời gian bằng `CUTOFF`.
- Cùng horizon (đặc biệt là `horizon=1`).

Sau khi chạy xong hai mô hình, nhóm phải trả lời ba câu.

1. Mô hình nào tốt hơn cho `horizon=1`? Ở đây sinh viên cần dựa vào số liệu MAE/RMSE và giải thích được vì sao dự báo rất ngắn hạn thường bị chi phối mạnh bởi độ trễ gần như `PM2.5_lag1`, khiến regression baseline thường bám sát tốt nếu feature engineering đúng, trong khi ARIMA cũng có thể tốt nhưng phụ thuộc vào cấu trúc tự tương quan và quyết định sai phân.

2. Mô hình nào ổn hơn khi có spike?: nhóm phải chọn một đoạn thời gian có đỉnh PM2.5 rõ (1–3 ngày), vẽ *forecast vs actual* của cả hai mô hình trên cùng đoạn đó, rồi phân tích mô hình nào phản ứng nhanh hơn hoặc bị muộn hóa quá mức; đồng thời liên hệ với sự khác nhau giữa RMSE và MAE, vì RMSE sẽ tăng mạnh nếu mô hình sai nặng ở một vài thời điểm spike.
3. Nếu triển khai thật, bạn chọn gì và vì sao?: câu này không chỉ dựa trên điểm số mà còn dựa trên bối cảnh vận hành: regression baseline thường dễ mở rộng khi muốn thêm đặc trưng, dễ cập nhật và chạy nhanh, còn ARIMA có ưu thế về giải thích theo (p, d, q) và có thể kèm khoảng tin cậy; nếu mục tiêu là cảnh báo sớm trong điều kiện thời tiết biến động mạnh.

5.3.2 Chủ đề 2: SARIMA – thêm mùa vụ (seasonality)

Chủ đề SARIMA yêu cầu nhóm nâng cấp trực tiếp từ ARIMA lên SARIMA $(p, d, q)(P, D, Q, s)$ để mô hình hóa mùa vụ theo ngày hoặc theo tuần.

Nhóm phải chứng minh rằng chuỗi PM2.5 có mùa vụ bằng cách dùng ACF: nếu thấy các đỉnh lặp lại mạnh ở lag 24, 48, ... thì đó là tín hiệu mùa vụ theo ngày, còn nếu xuất hiện tín hiệu ở lag 168, 336, ... thì gợi ý mùa vụ theo tuần.

Sau bước chứng minh, nhóm thử ít nhất một cấu hình SARIMA với $s=24$ (bắt buộc), và có thể thử thêm $s=168$ như một phiên bản nâng cấp.

Điểm quan trọng là nhóm không được grid search bừa, mà phải có chiến lược: giữ d theo kết luận stationarity của ARIMA baseline, sau đó chọn một vùng tìm kiếm nhỏ cho p, q và thêm P, Q ở mức thấp (0–2), cân nhắc D (0 hoặc 1) nếu mùa vụ thể hiện rõ và chuỗi có dấu hiệu không dừng theo chu kỳ.

Kết quả bắt buộc phải được trình bày dưới dạng so sánh giữa ARIMA và SARIMA bằng cả AIC/BIC lẫn RMSE/MAE trên tập test, sau đó viết kết luận ngắn gọn: SARIMA cải thiện/không cải thiện vì sao, và mùa vụ 24h hay 168h phù hợp hơn với trạm đó.

5.3.3 Chủ đề 3: SARIMAX – ARIMA + biến ngoại sinh (weather as X)

Chủ đề SARIMAX giúp sinh viên vượt qua giới hạn lớn nhất của ARIMA đơn giản: ARIMA chỉ nhìn vào quá khứ của PM2.5, trong khi PM2.5 chịu tác động mạnh bởi các yếu tố khí tượng.

Vì vậy, nhóm sẽ xây dựng mô hình **SARIMAX** bằng cách đưa một tập biến ngoại sinh như TEMP, WSPM, RAIN, PRES, DEWP làm *exog*, tối thiểu chọn 3 biến và phải giải thích lý do lựa chọn.

Trước khi fit mô hình, nhóm cần làm một minh họa nhỏ để chứng minh thời tiết có liên quan: có thể là tương quan giữa PM2.5 với WSPM/RAIN (có hoặc không độ trễ), hoặc một biểu đồ scatter đơn giản cho thấy khi gió mạnh hoặc mưa tăng thì PM2.5 có xu hướng giảm.

Yêu cầu quan trọng nhất của chủ đề này là **chống leakage**: nhóm phải mô tả rõ ràng cách ghép dữ liệu sao cho chỉ dùng $X(t)$ để dự báo $y(t+1)$; tuyệt đối không dùng X tương lai trong lúc dự báo, vì trong triển khai thật ta không biết thời tiết tương lai (trừ khi có nguồn dự báo thời tiết riêng, nhưng lab này chưa dùng).

Sau khi chạy, nhóm phải so sánh SARIMAX với baseline ARIMA/SARIMA bằng RMSE/MAE và bổ sung một phần đánh giá chất lượng mô hình bằng chẩn đoán phần dư: tối thiểu xem residual còn tự tương quan không (ví dụ ACF của residual) hoặc diễn giải trực quan residual trước/sau.

Kết quả bắt buộc: trả lời rõ SARIMAX cải thiện mạnh trong giai đoạn nào (thường là lúc thời tiết biến động, đặc biệt gió/mưa), cải thiện ít trong giai đoạn nào (khi chuỗi bị chi phối bởi chu kỳ nội tại), và nếu triển khai hệ thống cảnh báo sớm, SARIMAX sẽ được vận hành ra sao để cập nhật mô hình và theo dõi sai số theo thời gian.

5.4 Kết quả kỳ vọng

Mỗi nhóm cần hoàn thành:

- Blog/Report (link Notion/GitHub).
- Slide trình bày.

5.5 Gợi ý cho phần trình bày tại lớp

1. Giới thiệu bài toán và mục tiêu nhóm.
2. Trình bày kết quả trọng tâm (không kê lê, không giải thích code).
3. Diễn giải các biểu đồ.
4. Kết luận và đề xuất hành động.