

# Bài toán học máy

☑ Trắc nghiệm 32    📧 Thực hành 1    ⬇ PDF bài

Học máy là để máy tự tìm quy tắc dự đoán từ dữ liệu, thay vì ta tự viết quy tắc. Ví dụ: câu lạc bộ muốn đoán số người đến buổi sinh hoạt thứ Bảy để đặt ghế, dựa vào số người đã đăng ký và dự báo mưa; dữ liệu là các buổi trước. Trước khi chọn model, cần trả lời ba câu: mỗi hàng dữ liệu là gì, lúc dự đoán được dùng thông tin nào, và kết quả được chấm thế nào. Sau đó ta viết một baseline: cách đoán đơn giản nhất chạy được, làm mốc để so với mọi model về sau.

MỤC TIÊU

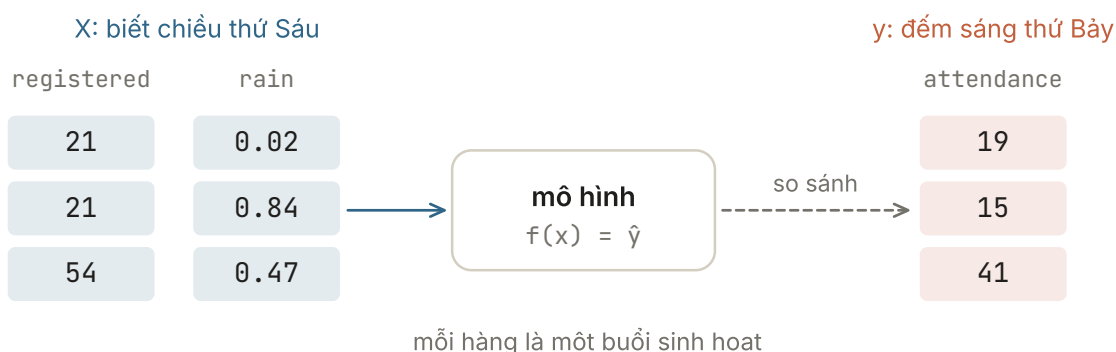
- Xác định input, output, thời điểm dự đoán và metric của một bài toán.
- Phân biệt classification, regression và học không giám sát.
- Dựng baseline bằng scikit-learn và so với cách đoán hằng số.
- Ghi file dự đoán đúng ID, đúng định dạng.

## Từ câu chuyện đến một hàng dữ liệu

Mỗi hàng của bảng là một mẫu (sample), mỗi cột là một thông tin về mẫu. Ta quy ước mỗi hàng là **một buổi sinh hoạt**, không phải một học sinh. Đặc trưng (feature) là thông tin dùng để dự đoán; nhãn hay mục tiêu (target) là giá trị cần đoán. Bảng có thể chứa cả hai lúc huấn luyện, nhưng lúc sử dụng mô hình thì chỉ có đặc trưng.

| Cột           | Ý nghĩa                                   | Biết chiều thứ Sáu? | Dùng làm gì?               |
|---------------|-------------------------------------------|---------------------|----------------------------|
| session_id    | Mã buổi sinh hoạt                         | Có                  | Ghép kết quả với đúng buổi |
| registered    | Số người đã đăng ký                       | Có                  | Feature                    |
| rain_forecast | Xác suất mưa được dự báo lúc chốt đăng ký | Có                  | Feature                    |
| attendance    | Số người thực sự đến                      | Không               | Target                     |

Không đưa attendance vào input. Cũng không thay rain\_forecast bằng lượng mưa đo được sau buổi sinh hoạt: tên cột khác nhau chỉ một chút, nhưng thời điểm biết thông tin đã đổi. Bài 1.4 gọi những đường tắt như vậy là data leakage.



Từ thông tin chiều thứ Sáu, mô hình đoán số người đến; nhân chỉ có sau buổi.

Bảng như trên, có cột cố định và mỗi dòng một mẫu, gọi là dữ liệu có cấu trúc. Ảnh, âm thanh và văn bản gọi là dữ liệu phi cấu trúc; muốn đưa vào model thì phải biến thành số trước. Ví dụ ảnh chữ số viết tay  $8 \times 8$  dưới đây được trải thành một hàng 64 số. Theo loại dữ liệu, ứng dụng AI thường chia ba mảng: dữ liệu bảng (khoa học dữ liệu, như gợi ý quảng cáo từ lịch sử mua hàng), ảnh và video (thị giác máy tính), chữ và lời nói (xử lý ngôn ngữ tự nhiên).



X: 64 cột, mỗi cột là một điểm ảnh y  
ảnh 1: lưới  $8 \times 8$  trải theo từng hàng thành 64 số, nhân 3

X: 64 cột, mỗi cột là một điểm ảnh y  
ảnh 1: lưới  $8 \times 8$  trải theo từng hàng thành 64 số, nhân 3



X: 64 cột, mỗi cột là một điểm ảnh y  
ảnh 3: lưới  $8 \times 8$  trải theo từng hàng thành 64 số, nhân 0



X: 64 cột, mỗi cột là một điểm ảnh y  
4 ảnh thành bảng X shape (4, 64) và vector nhãn y shape (4,)

Ảnh chữ số viết tay thật  $8 \times 8$  trải thành một hàng 64 số: mỗi ảnh là một hàng của X, chữ số đúng là giá trị y.

Trước khi viết code, nên tóm tắt bài toán thành một câu: “Với mỗi buổi, từ thông tin có lúc chiều thứ Sáu, đoán số người đến vào thứ Bảy; sai số tính bằng số người.” Câu này cho biết được dùng cột nào, chia dữ liệu thế nào và đọc kết quả ra sao.

#### HỎI NHANH

Câu lạc bộ muốn chạy mô hình vào chiều thứ Sáu để đoán số người đến buổi sáng thứ Bảy. Cột nào **không** được dùng làm feature?

- A Số người đăng ký tính đến chiều thứ Sáu
- B Số buổi câu lạc bộ đã tổ chức từ đầu học kỳ
- C Dự báo mưa lưu lúc chiều thứ Sáu
- D Số người quét mã vào cửa sáng thứ Bảy

---

## Đầu ra quyết định loại bài toán

Nếu hỏi “có ít nhất 30 người đến hay không?”, output là một nhãn trong hai lớp: đây là classification (phân loại). Nếu hỏi “có bao nhiêu người đến?”, output là một số: đây là regression (hồi quy). Số người là số nguyên, nhưng regression vẫn có thể dự đoán số thực như 28.4, hiểu là mức ước lượng. Có làm tròn hay không phải theo metric và quy định đầu ra.

Phân loại và hồi quy đều là **học có giám sát** (supervised learning): model học từ các cặp  $(\mathbf{x}_i, y_i)$ , tức đặc trưng và nhãn của mẫu thứ  $i$ . Vài điều cần nhớ:

- Số lớp có thể nhiều hơn hai: đọc chữ số viết tay có 10 lớp, từ 0 tới 9; nhận cảm xúc trên khuôn mặt có một lớp cho mỗi cảm xúc.
- Loại bài do đầu ra quyết định, không do đầu vào: đoán tuổi từ ảnh khuôn mặt vẫn là hồi quy.
- Khi dữ liệu không có nhãn, ta làm **học không giám sát** (unsupervised learning). Hai việc hay gặp là phân cụm (clustering), như gom khách hàng hay bài báo giống nhau thành nhóm, và phát hiện bất thường. Phân cụm không tự cho biết nhóm nào “tốt”, vì không có nhãn nào nói điều đó.
- Khi chỉ một phần nhỏ dữ liệu có nhãn, **học bán giám sát** (semi-supervised) dùng thêm cả phần chưa có nhãn.

Bài đặt ghế là regression. Model là một hàm  $f$  nhận đặc trưng và trả về dự đoán  $\hat{y} = f(\mathbf{x})$ . **Huấn luyện** là dùng các cặp cũ để tìm hàm đó; **dự đoán** (inference) là đưa đặc trưng mới vào hàm đã tìm, lúc này không có nhãn. Trong scikit-learn, huấn luyện là `model.fit(X_train, y_train)`, dự đoán là `model.predict(X_new)`; gọi `fit` lần nữa là học lại từ đầu.

### Classification



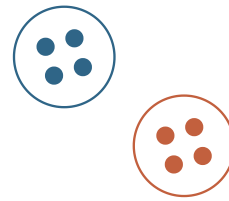
≥ 30 người?

### Regression

28.4  
một số thực

bao nhiêu người?

### Clustering



không có nhãn

Loại bài toán do đầu ra quyết định: một nhãn, một số, hay chưa có nhãn.

#### HỎI NHANH

Input là ảnh chụp một bài giải. Output là một trong ba nhãn “đúng”, “sai”, “chưa đủ bước”. Đây là loại bài toán nào?

- A Regression, vì nhãn mã hóa được thành 0, 1, 2
- B Classification có giám sát, với ba lớp
- C Clustering, vì cần chia bài giải thành ba nhóm
- D Classification nhị phân, đúng hoặc không đúng

## Tạo một bộ dữ liệu có thể kiểm chứng

Ta tự sinh 120 buổi. Công thức giả định rằng số đến tăng theo đăng ký, giảm theo dự báo mưa, cộng thêm nhiễu do những yếu tố không ghi lại. Vì tự sinh nên ta biết quy luật thật, và có thể đối chiếu xem mô hình học được bao nhiêu phần của nó.

Python

```
import numpy as np
import pandas as pd

rng = np.random.default_rng(11)
n = 120
registered = rng.integers(15, 65, size=n)
```

```

rain_forecast = rng.uniform(0, 1, size=n)
attendance = np rint(
    4 + 0.75 * registered - 7 * rain_forecast
    + rng.normal(0, 2, size=n)
).clip(0)
data = pd.DataFrame({
    "session_id": np.arange(n),
    "registered": registered,
    "rain_forecast": rain_forecast,
    "attendance": attendance,
})
X = data[["registered", "rain_forecast"]]
y = data["attendance"]
print(X.shape, y.shape)
print(data.head(3).round(2).to_string(index=False))

```

Nhìn shape trước: X có 120 hàng, hai cột; y có 120 phần tử. `data[...]` chọn một bảng hai chiều; `data["attendance"]` chọn một vector nhãn. Mã buổi vẫn nằm trong data, nhưng không xuất hiện trong X. Không có lý do chung nào để số thứ tự tự nó dự đoán số người đến.

#### HỎI NHANH

`session_id` chạy từ 0 đến 119 theo thứ tự thời gian. Vì sao không đưa cột này vào X?

- A ID chỉ là mã để ghép kết quả, buổi mới sẽ có ID chưa từng thấy
- B Đưa vào cũng không sao, mô hình tự bỏ qua cột vô dụng
- C Nên đưa vào, miễn là chuẩn hóa trước cho cùng thang
- D Phải đưa vào để dự đoán giữ đúng thứ tự các buổi

| Đại lượng | Shape    | Một phần tử nói gì?      |
|-----------|----------|--------------------------|
| <i>X</i>  | (120, 2) | Hai feature của một buổi |
| <i>y</i>  | (120,)   | Số người đến             |
|           |          |                          |

| Đại lượng | Shape               | Một phần tử nói gì?      |
|-----------|---------------------|--------------------------|
| $\hat{y}$ | (số_buổi_cần_đoán,) | Một dự đoán cho mỗi buổi |

## Ba phần dữ liệu, ba công việc

Các hàng đang xếp theo thứ tự thời gian. Ta chia thành ba phần, mỗi phần một việc:

- **Train**, 80 buổi đầu: để model học.
- **Validation**, 20 buổi tiếp: để so các model và chọn cái tốt nhất.
- **Test**, 20 buổi cuối: giữ riêng đến cuối, chấm đúng một lần cho model đã chọn.

Test chỉ chấm một lần nên điểm test gần với điểm khi gặp dữ liệu mới. Nếu dùng test để chọn giữa nhiều model, model thắng có phần may trên chính tập đó, và điểm báo cáo sẽ đẹp hơn thực tế. Khi so hai model, phải chấm chúng trên cùng một tập.



Chia theo thứ tự buổi: học trên quá khứ, chọn và chấm trên các buổi sau.

Lưu ý về tên gọi: trong bài này test có nhãn, để ta tự chấm được. Trong cuộc thi, file `test.csv` của đề thường **không** có nhãn; đó là dữ liệu cần dự đoán để nộp, còn việc chấm do ban tổ chức làm.

Python

```
train_rows = np.arange(0, 80)
valid_rows = np.arange(80, 100)
test_rows = np.arange(100, 120)
X_train, y_train = X.iloc[train_rows], y.iloc[train_rows]
X_valid, y_valid = X.iloc[valid_rows], y.iloc[valid_rows]
X_test, y_test = X.iloc[test_rows], y.iloc[test_rows]
```

```
print(len(train_rows), len(valid_rows), len(test_rows))
print(len(set(train_rows) & set(test_rows)))
```

Dòng cuối phải bằng 0: không có hàng nào vừa học vừa chấm test. Đó là điều kiện cần nhưng chưa đủ: nếu một buổi bị ghi thành hai hàng, hay nhiều hàng cùng thuộc một nhóm gắn bó (cùng trường, cùng người), còn phải chia theo nhóm, bài 1.3 sẽ làm.

#### HỎI NHANH

Có hai mô hình, ta cần chọn một. Ta so điểm của chúng trên tập nào?

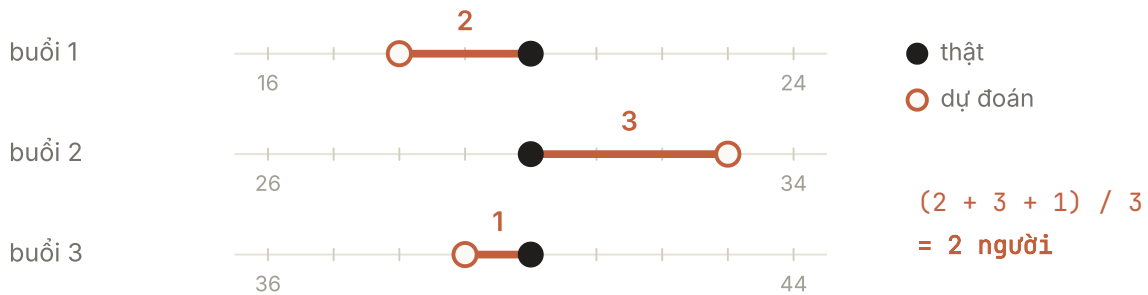
- A Validation
- B Train, vì mô hình đã học trên đó
- C Test, tập đã giữ riêng từ đầu
- D Gộp train và test lại

## Baseline và một thước đo đọc được

Ta dùng mean absolute error (MAE): trung bình trị tuyệt đối của các sai số. Với  $m$  mẫu đang chấm,

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |\hat{y}_i - y_i|.$$

Ví dụ ba buổi có số người thật (20, 30, 40), dự đoán (18, 33, 39). Sai số là  $(-2, 3, -1)$ , nên  $\text{MAE} = (2 + 3 + 1)/3 = 2$  người. Trung bình của sai số có dấu bằng 0, nhưng dự đoán vẫn sai; lấy trị tuyệt đối để sai cao và sai thấp không bù trừ nhau.



MAE là trung bình độ dài các đoạn sai số, tính theo đơn vị người.

MAE bằng 2 nghĩa là trung bình độ lệch có độ lớn hai người; không có nghĩa mọi buổi đều sai đúng hai người. Số càng nhỏ càng tốt. Bài 1.6 sẽ so với những metric khác.

Baseline hằng số dùng `DummyRegressor(strategy="mean")`: dự đoán mọi buổi bằng trung bình **train**. Phương án thứ hai là `LinearRegression`, cộng các đóng góp của hai feature. Ta chưa cần tự cài thuật toán tìm hệ số; bài 2.1 sẽ giải thích cách học nó. Với bài phân loại, baseline tương ứng là luôn đoán lớp đông nhất: dữ liệu có 70% khách ở lại thì luôn đoán “ở lại” đã đạt accuracy 70%.

Python

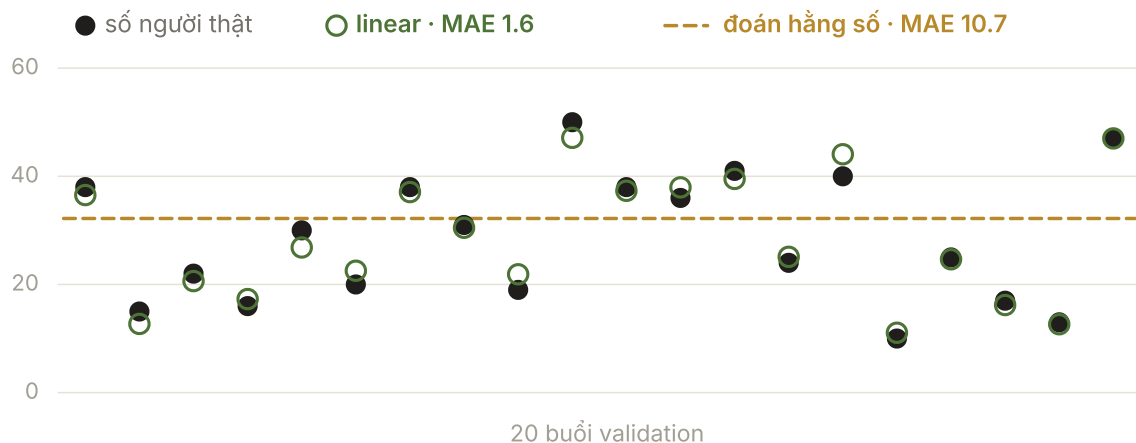
```
from sklearn.dummy import DummyRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error

models = {
    "mean": DummyRegressor(strategy="mean"),
    "linear": LinearRegression(),
}

valid_mae = {}
for name, model in models.items():
    model.fit(X_train, y_train)  # chỉ học từ train
    prediction = model.predict(X_valid)
    assert prediction.shape == y_valid.shape
    valid_mae[name] = mean_absolute_error(y_valid, prediction)
    print(name, f"valid MAE = {valid_mae[name]:.3f}")
chosen_name = min(valid_mae, key=valid_mae.get)
print("Chọn bằng validation:", chosen_name)
```

So hai dòng `valid MAE` trên cùng 20 buổi: đoán hằng số sai trung bình gần 11 người, `linear` chỉ sai khoảng 1.6 người. Số đăng ký và dự báo mưa mang gần hết thông tin cần

thiết, đúng như công thức sinh dữ liệu. Baseline hằng số còn một vai trò nữa: nó là mốc tối thiểu để phát hiện lỗi.



Trên 20 buổi validation: đường hằng số lệch xa, linear bám sát số người thật.

#### HỎI NHANH

Trên cùng 20 buổi validation, một mô hình mới có MAE 11.5 người, còn baseline đoán hằng số có MAE 10.74. Ta nên làm gì tiếp?

- A Giữ mô hình mới, vì nó dùng nhiều thông tin hơn
- B Bỏ baseline đi, vì đoán hằng số không phải mô hình
- C Tìm lỗi trong dữ liệu hoặc trong cách tính điểm
- D Tính thêm điểm trên test để có thêm ý kiến

Đã chọn xong, ta học lại model trên cả train lẫn validation (nhiều dữ liệu hơn thì học tốt hơn), rồi chấm test đúng một lần. Nếu xem điểm test xong lại sửa model, test đã thành một tập validation nữa, không còn độc lập.

Python

```
final_model = models[chosen_name]
final_model.fit(X.iloc[:100], y.iloc[:100])
```

```
test_prediction = final_model.predict(X_test)
print(f"Test MAE = {mean_absolute_error(y_test, test_prediction):.3f}")
```

Đọc theo đơn vị người: dự đoán lệch trung bình chừng một, hai người, nên đặt thêm vài ghế dự phòng là đủ. MAE là trung bình, nên vẫn có buổi lệch nhiều hơn. Con số này chỉ đo trên 20 buổi nên còn dao động (bài 0.7), và chỉ đúng khi các buổi sau còn giống các buổi đã thấy.

## Ghi file dự đoán

Giả sử đề yêu cầu CSV gồm `session_id`, `attendance`. Ta tạo bảng trong bộ nhớ, giữ ID theo thứ tự input, và đọc lại bằng pandas như một phép kiểm vòng kín.

Python

```
from io import StringIO

submission = pd.DataFrame({
    "session_id": data.iloc[test_rows]["session_id"].to_numpy(),
    "attendance": test_prediction,
})
csv_text = submission.to_csv(index=False)
read_back = pd.read_csv(StringIO(csv_text))
assert list(read_back.columns) == ["session_id", "attendance"]
assert read_back["session_id"].is_unique
assert np.isfinite(read_back["attendance"]).all()
assert read_back["session_id"].tolist() == list(test_rows)
print(read_back.head(3).round(2).to_string(index=False))
print("Số hàng:", len(read_back))
```

`index=False` để không ghi thêm cột index. Với đề thật, tên cột và định dạng lấy đúng theo đề, và các dòng `assert` sửa theo cho khớp.

### HỎI NHANH

Một bạn sắp xếp cột dự đoán từ nhỏ đến lớn nhưng để nguyên cột ID. File nộp còn đúng không?

- A Đúng, vì số hàng và số cột vẫn khớp với đề
- B Đúng, vì MAE không phụ thuộc thứ tự hàng
- C Sai, vì phải sắp theo ID tăng dần
- D Sai, vì dự đoán đã bị ghép nhầm sang mẫu khác

## Lần nộp đầu tiên

Đề Thực hành đầu tiên, [Hình chữ nhật ẩn](#) (ROAI 2025), cho các điểm trên hình vuông  $[0, 100]^2$ : điểm trong một hình chữ nhật ẩn có nhãn 1, ngoài là 0. Input là tọa độ X, Y; output là nhãn của từng datapointID; metric là accuracy.

Cách đơn giản: lấy hình chữ nhật nhỏ nhất chứa mọi điểm train nhãn 1, điểm test nào nằm trong thì đoán 1. Thử trên dữ liệu tự sinh với 30 điểm train, như kịch bản khó nhất của đề:

Python

```
rng = np.random.default_rng(5)
x_lo, x_hi, y_lo, y_hi = 20, 65, 30, 80 # hình chữ nhật ẩn, chỉ dùng để sinh dữ liệu

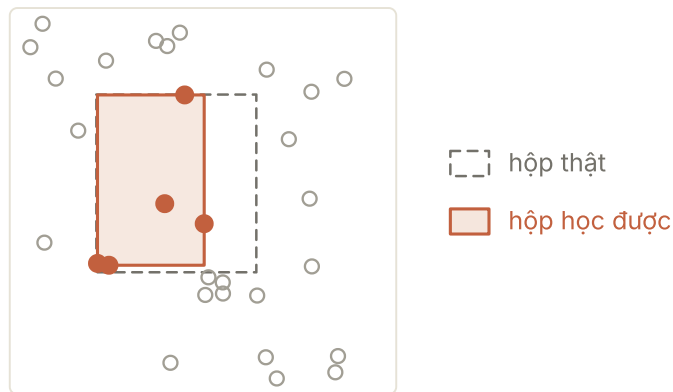
def make_points(n):
    points = rng.uniform(0, 100, size=(n, 2))
    inside = (points[:, 0] >= x_lo) & (points[:, 0] <= x_hi) & (points[:, 1] >= y_lo) & (points[:, 1] <= y_hi)
    return pd.DataFrame({"X": points[:, 0], "Y": points[:, 1], "label": inside.astype(int)})

train_pts, test_pts = make_points(30), make_points(3000)
pos = train_pts[train_pts["label"] == 1]
box = (pos["X"].min(), pos["X"].max(), pos["Y"].min(), pos["Y"].max())
```

```

answer = (test_pts["X"].between(box[0], box[1]) & test_pts["Y"].between(box[2],
box[3])).astype(int)
print("số điểm nhãn 1 trong train:", len(pos))
print("hộp học được:", [round(float(v), 1) for v in box])
print("accuracy đoán toàn 0:", round(float((test_pts["label"] == 0).mean()), 3))
print("accuracy hộp nhỏ nhất:", round(float((answer == test_pts["label"]).mean()), 3))
sub_pts = pd.DataFrame({"subtaskID": 3, "datapointID": np.arange(len(test_pts)), "answer":
answer})
print(sub_pts.head(3).to_string(index=False))

```



Hộp học được luôn nằm trong hộp thật.

Chỉ 5 điểm nhãn 1 đã cho accuracy 0.917, so với 0.773 khi đoán toàn 0. Cạnh phải dừng ở 50.3 trong khi hộp thật kéo tới 65: hộp nhỏ nhất luôn nằm bên trong hộp thật, nên mọi lỗi là đoán 0 cho điểm sát biên. Điểm nhãn 0 trong train cho biết biên có thể nói tới đâu.

Để nộp: tải dữ liệu ở trang đề, chạy `baseline.ipynb`, nộp `submission.zip` trên LQDOJ với kiểu **Output**. Ở đề thật, `datapointID` lấy từ `test.csv`, mỗi kịch bản một hình chữ nhật riêng.

## Khi đi thi

- Viết bài toán thành một câu: mỗi mẫu là gì, lúc dự đoán biết gì, chấm bằng metric nào.
- Nộp baseline sớm: vừa có mốc để so, vừa kiểm được đường đi tới file nộp.
- Giữ đúng ID cho từng dự đoán, đọc lại file trước khi nộp.

## Tóm tắt

- Bài học máy bắt đầu bằng: mỗi mẫu là gì, lúc dự đoán biết gì, đầu vào và đầu ra là gì.
- Train để học, validation để chọn, test để chấm một lần sau khi đã chọn.
- Baseline là mốc so sánh; model thua baseline thường là có lỗi.
- File nộp phải đúng cột, và mỗi dự đoán đi với đúng ID của nó.

## Kiểm tra cuối bài

### CÂU 1

Một bảng có 12 trường, mỗi trường ghi lại 8 buổi sinh hoạt. Ta cần đoán số người đến từng buổi. Mỗi mẫu là gì, và target của nó là gì?

- A Mỗi mẫu là một cột, target là tổng của cột ấy
- B Mỗi mẫu là một buổi, target là số người thực sự đến buổi ấy
- C Mỗi mẫu là một trường, target là số người đến trung bình qua 8 buổi
- D Mỗi mẫu là một người, target là số buổi người ấy tham gia

### CÂU 2

Đề yêu cầu đoán số người đến từng buổi và chấm bằng MAE. Một bạn trong nhóm đổi sang bài phân loại “đông” (từ 30 người trở lên) hay “vắng” cho dễ huấn luyện. Làm vậy sai ở đâu?

- A Không sai, vì phân loại thường chính xác hơn hồi quy
- B Đầu ra không còn là thứ để chấm bằng MAE
- C Phân loại cần nhiều dữ liệu hơn nên không chạy được
- D Nhãn “đông” hay “vắng” để lộ thông tin của buổi

### CÂU 3

Bảng chỉ có feature của từng buổi, không có cột nhãn. Ta muốn gom các buổi giống nhau thành từng nhóm. Đây là loại bài toán nào?

- A Học không giám sát, chẳng hạn clustering
- B Học có giám sát, vì feature đã là số
- C Classification có giám sát
- D Regression có giám sát

PHÒNG THEO [USA-NA-AIO 2026 · Vòng 1 · Part 1.1](#)

#### CÂU 4

Một mô hình được huấn luyện và kiểm trên dữ liệu năm 2023: đúng 99.8% trên train, 92% trên validation. Dem dùng cho dữ liệu năm 2025 thì chỉ còn đúng 64%. Vấn đề chính nằm ở đâu?

- A Dữ liệu năm 2025 đã khác dữ liệu năm 2023
- B Validation cùng năm với train nên con số 92% là ảo
- C Dữ liệu năm 2025 bị gán nhãn sai nên điểm thấp giả tạo
- D Mô hình quá yếu, dù đúng tới 99.8% trên train

**ĐỀ THI** NOAI 2026 Singapore · Vòng sơ loại · Câu 286

#### CÂU 5

Hai mẫu có nhãn thật là 10 và 20. Mô hình đoán 13 và 19. MAE bằng bao nhiêu?

- A 1
- B 4
- C 5
- D 2

**CÂU 6**

Baseline DummyRegressor(strategy="mean") đoán mọi buổi bằng một số trung bình. Số trung bình ấy phải tính trên tập nào?

A Validation

B Test

C Train

D Cả ba tập

**CÂU 7**

Đề chấm bằng MAE. Trên cùng tập validation, mô hình A có MAE 3 và sai số lớn nhất là 7. Mô hình B có MAE 4 và sai số lớn nhất là 5. Ta chọn mô hình nào?

A A, vì tổng hai con số của A nhỏ hơn

B A, vì MAE thấp hơn

C B, vì sai số lớn nhất nhỏ hơn

D B, vì hai con số của B gần nhau hơn

### CÂU 8

Dữ liệu gồm các buổi sinh hoạt của nhiều trường. Ta chia ngẫu nhiên các buổi, nên validation chỉ gồm những trường đã có trong train. Trên validation, baseline có MAE 8, linear có MAE 2. Nhưng ta muốn dùng model cho một trường mới chưa có trong dữ liệu. Nên làm thí nghiệm nào tiếp theo?

- A Chia ngẫu nhiên thêm nhiều lần, lấy lần có điểm tốt nhất
- B Học lại linear trên cả validation rồi tính MAE ngay trên đó
- C Giữ cách chia cũ, chỉ làm tròn MAE cho gọn
- D Để riêng toàn bộ vài trường ra, rồi so lại hai mô hình

### CÂU 9

Một đội thử vài mô hình, chọn mô hình có điểm test cao nhất, rồi báo điểm test ấy là đánh giá độc lập. Làm vậy có đúng không?

- A Đúng, vì test không dùng trong fit
- B Đúng, nếu chỉ thử hai mô hình
- C Sai, vì test đã được dùng để chọn
- D Đúng, nếu điểm test cao hơn validation

#### CÂU 10

Ghi file nộ bằng pandas. Viết thế nào để file không có thêm cột index?

- A `to_csv(index=True)`
- B `sort_values("attendance")`
- C `reset_index()` rồi xuất mọi cột
- D `to_csv(index=False)`

### Đọc thêm

- James, Witten, Hastie, Tibshirani và Taylor — [An Introduction to Statistical Learning](#), chương 2. Đọc để phân biệt dự đoán, suy luận và các loại bài toán học thống kê.
- [scikit-learn — DummyRegressor](#). Xem các baseline hằng số và ý nghĩa của `strategy`.
- [scikit-learn — Metrics and scoring](#). Tra đúng hàm metric thay vì mặc định dùng `model.score`.