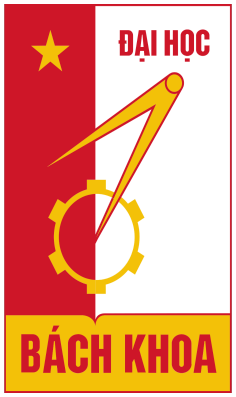
ĐẠI HỌC BÁCH KHOA HÀ NỘI
Khoa Toán Tin



BÁO CÁO CUỐI KÌ
Hệ hỗ trợ quyết định

Dự báo nhiệt độ trung bình London

SV thực hiện: Cao Quang Đăng
MSSV: 20227219
GVHD: TS. Trần Ngọc Thăng

Hà Nội, năm 2025

Mục lục

1 Lời nói đầu

Trong bối cảnh biến đổi khí hậu đang trở thành một thách thức toàn cầu, việc dự báo nhiệt độ chính xác không chỉ có ý nghĩa khoa học mà còn mang lại những ứng dụng thiết thực trong quy hoạch đô thị, nông nghiệp và quản lý năng lượng. London, một trong những trung tâm kinh tế - văn hóa lớn của thế giới, cũng chịu ảnh hưởng rõ rệt bởi sự thay đổi của khí hậu. Do đó, bài toán dự báo nhiệt độ trung bình tại đây không chỉ giúp hiểu rõ hơn về xu hướng thời tiết mà còn góp phần vào các chiến lược thích ứng và giảm thiểu rủi ro.

Với sự phát triển mạnh mẽ của học máy (Machine Learning), các phương pháp như hồi quy tuyến tính, mạng nơ-ron nhân tạo (ANN), hay mô hình chuỗi thời gian như ARIMA, LSTM đã mở ra nhiều hướng tiếp cận mới để giải quyết bài toán dự báo nhiệt độ. Bản báo cáo này sẽ khám phá hiệu quả của các mô hình học máy khác nhau trong việc dự đoán nhiệt độ trung bình tại London, từ đó đánh giá ưu nhược điểm của từng phương pháp và đề xuất hướng cải thiện.

Thông qua phân tích dữ liệu lịch sử và so sánh các kỹ thuật học máy, chúng tôi hy vọng sẽ cung cấp một cái nhìn tổng quan về khả năng ứng dụng của từng mô hình, đồng thời gợi mở những nghiên cứu sâu hơn trong lĩnh vực dự báo khí hậu. Báo cáo không chỉ hữu ích cho các nhà khoa học dữ liệu mà còn là tài liệu tham khảo cho những ai quan tâm đến giao thoa giữa công nghệ và môi trường.

Em xin chân thành gửi lời cảm ơn sâu sắc đến thầy Trần Ngọc Thăng, người đã tận tâm hướng dẫn, động viên và hỗ trợ em trong suốt quá trình thực hiện báo cáo này. Những lời góp ý quý báu, sự kiên nhẫn chỉ dẫn từng bước, cùng những trao đổi học thuật sâu sắc của thầy đã giúp em không chỉ hoàn thành công việcnày một cách tốt nhất mà còn tích lũy thêm nhiều kiến thức và kinh nghiệm quý giá. Em rất trân trọng sự nhiệt huyết và tâm huyết của thầy dành cho học trò, điều đó đã truyền cảm hứng để em không ngừng cố gắng và nỗ lực hơn nữa trong học tập và nghiên cứu.

2 Giới thiệu chung

2.1 Mô tả bài toán

Dự báo thời tiết, đặc biệt là nhiệt độ trung bình, đóng vai trò thiết yếu trong nhiều lĩnh vực của đời sống xã hội. Việc dự báo chính xác nhiệt độ giúp các cơ quan chức năng và người dân chủ động trong quản lý đô thị, điều phối giao thông, bảo vệ sức khỏe cộng đồng, tối ưu hóa tiêu thụ năng lượng, và hỗ trợ các hoạt động sản xuất nông nghiệp, du lịch, dịch vụ. Tại các đô thị lớn như London, nhiệt độ trung bình có ảnh hưởng đặc biệt đến chất lượng sống của người dân cũng như hiệu quả vận hành của thành phố.

Trong những năm gần đây, London đã và đang chịu nhiều tác động từ biến đổi khí hậu toàn cầu. Nhiệt độ trung bình có xu hướng tăng lên, mùa hè trở nên nắng nóng và kéo dài, trong khi mùa đông lại có xu hướng ẩm ướt và lạnh sâu hơn. Bên cạnh đó, sự gia tăng bất thường của biến động nhiệt độ trong ngày cũng gây khó khăn trong việc thích nghi và lập kế hoạch ứng phó của các cơ quan chức năng. Việc hiểu rõ xu hướng nhiệt độ và xây dựng các mô hình dự báo chính xác không chỉ phục vụ nhu cầu dân sinh mà còn hỗ trợ chiến lược thích ứng và giảm thiểu tác động tiêu cực của biến đổi khí hậu đối với thành phố.

Nhờ vào sự phát triển của công nghệ cảm biến, hệ thống thu thập dữ liệu thời tiết tự động và kho dữ liệu mở, kết hợp với các phương pháp học máy và phân tích chuỗi thời gian, bài toán dự báo thời tiết đã có nhiều bước tiến vượt bậc. Các mô hình thống kê và học sâu hiện nay cho phép khai thác mối quan hệ phức tạp trong dữ liệu khí tượng, từ đó xây dựng các hệ thống dự báo có độ chính xác cao. Trong nghiên cứu này, em sử dụng dữ liệu thực tế , bao gồm các thông tin khí tượng lịch sử tại London, để xây dựng mô hình dự báo nhiệt độ trung bình trong tương lai, phục vụ cho các mục tiêu ứng dụng thực tiễn và nghiên cứu học thuật.

2.2 Các đặc trưng đầu vào (Input)

Bộ dữ liệu: **London Weather Data**
Nguồn: **Kaggle**
`https://www.kaggle.com/datasets/emmanuelfwerr/london-weather-data/data`
Dữ liệu bao gồm những đặc trưng thời tiết như số giờ nắng, bức xạ, áp suất, độ ẩm, nhiệt độ, ... được đo hằng ngày tại London giai đoạn 1979 - 2020.
Dữ liệu dạng bảng như sau:

	date	cloud_cover	sunshine	global_radiation	max_temp	mean_temp	min_temp	precipitation	pressure	snow_depth
0	1970-01-01 00:00:00.019790101	2.0	7.0	52.0	2.3	-4.1	-7.5	0.4	101900.0	9.0
1	1970-01-01 00:00:00.019790102	6.0	1.7	27.0	1.6	-2.6	-7.5	0.0	102530.0	8.0
2	1970-01-01 00:00:00.019790103	5.0	0.0	13.0	1.3	-2.8	-7.2	0.0	102050.0	4.0
3	1970-01-01 00:00:00.019790104	8.0	0.0	13.0	-0.3	-2.6	-6.5	0.0	100840.0	2.0
4	1970-01-01 00:00:00.019790105	6.0	2.0	29.0	5.6	-0.8	-1.4	0.0	102250.0	1.0

Hình 1: Dữ liệu thời tiết London

Các đặc trưng đầu vào:

- `date`: Thời điểm lấy dữ liệu
- `cloud_cover`: Độ che phủ mây
- `sunshine`: Số giờ nắng
- `global_radiation`: Bức xạ toàn phần
- `max_temp`: Nhiệt độ cao nhất trong ngày
- `min_temp`: Nhiệt độ thấp nhất trong ngày
- `precipitation`: Lượng mưa
- `pressure`: Áp suất
- `snow_depth`: Độ dày tuyết

2.3 Mục tiêu đầu ra (Output)

Output đầu ra của bài toán (Mục tiêu cần dự báo): `mean_temp` – Nhiệt độ trung bình Trong khoảng thời gian (n) nào đó (Phụ thuộc vào việc chia các tập huấn luyện - kiểm tra)
Kết quả đánh giá cho mô hình thông qua các chỉ số đánh giá
Trực quan hóa kết quả dự báo với thực tế để so sánh độ chính xác.

2.4 Yêu cầu xử lý

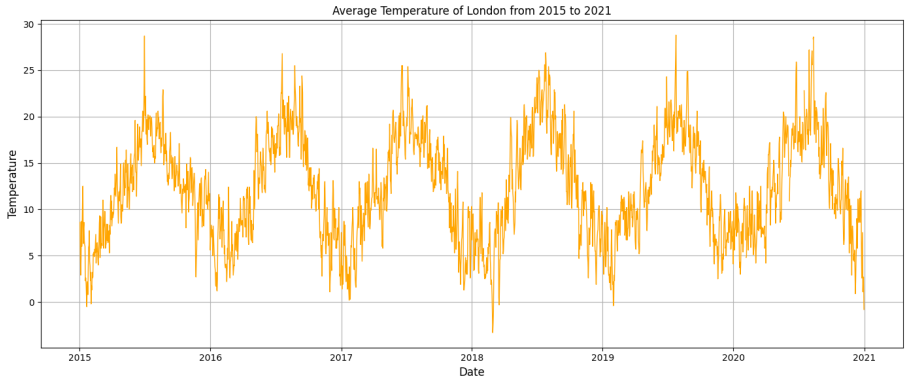
1. Thu thập và tìm hiểu bộ dữ liệu
   - Thu thập dữ liệu trên nguồn tin cậy (Kaggle)
   - Tìm hiểu các đặc trưng trong bộ dữ liệu, đầu vào, đầu ra, mục tiêu, ...
2. Tiền xử lý dữ liệu
   - Thống kê dữ liệu mẫu: Thông tin tổng thể về bộ dữ liệu, độ dài dữ liệu, các chỉ số trung bình, trung vị, ...

- Thống kê dữ liệu thiếu: Thống kê xem mỗi đặc trưng có bao nhiêu dữ liệu thiếu
- Xử lý dữ liệu thiếu bằng các phương pháp như nội suy, điền trung bình, 0, ... tùy theo các đặc trưng mô hình để đảm báo mô hình tối ưu nhất có thể
- Chuyển đổi dữ liệu (nếu cần) theo đặc thù các mô hình, như chuẩn hóa trong LSTM, làm dữ liệu dùng theo Arima, Sarima, ...

3. Trực quan hóa nhiệt độ trung bình London để dễ so sánh về sau
4. Sắp xếp dữ liệu tăng dần theo thời gian

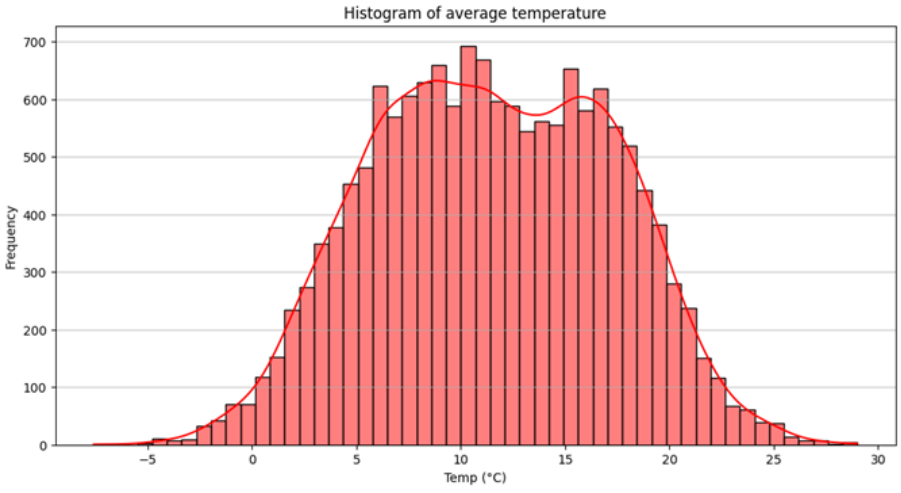
2.5 Tổng quan dữ liệu gốc

Biểu đồ nhiệt độ trung bình gốc



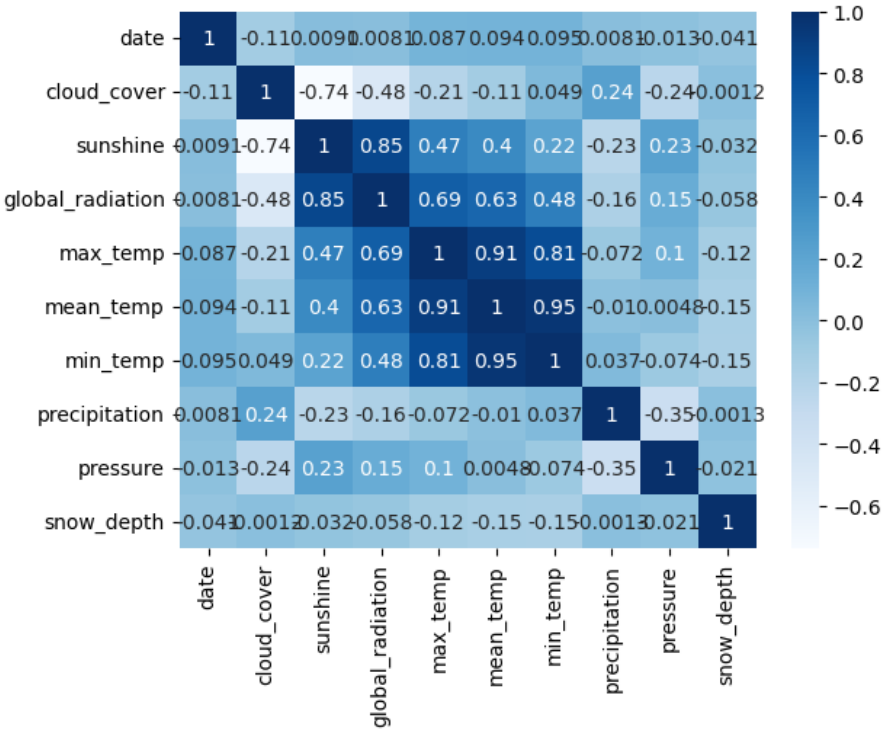
Hình 2: Dữ liệu TB theo ngày 2015 - 2021

Biểu đồ Histogram



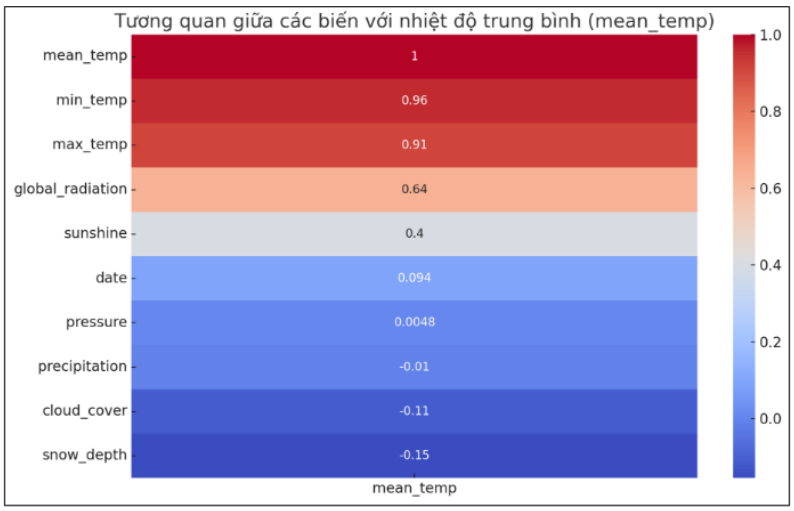
Hình 3: Tần suất phân bố nhiệt độ trung bình

Ma trận tương quan



Hình 4: Ma trận tương quan các đặc trưng

Mức độ tương quan các đặc trưng đối với biến mục tiêu



Hình 5: Mức độ tương quan tới `mean_temp`

Sau khi đã trình bày qua một số thông tin cơ bản về dữ liệu, ta sẽ đi vào xử lý dữ liệu.

3 Xử lý dữ liệu

3.1 Đánh nhãn & Tiền xử lý dữ liệu

1. Đánh nhãn dữ liệu Do dữ liệu thu thập được đã có tên đặc trưng rõ ràng, ta không cần đánh nhãn dữ liệu trong trường hợp này.

2. Tiền xử lý dữ liệu
   Kiểm tra thông tin tổng quan về dữ liệu

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15341 entries, 0 to 15340
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   date              15341 non-null  int64
 1   cloud_cover       15322 non-null  float64
 2   sunshine          15341 non-null  float64
 3   global_radiation  15322 non-null  float64
 4   max_temp          15335 non-null  float64
 5   mean_temp         15305 non-null  float64
 6   min_temp          15339 non-null  float64
 7   precipitation     15335 non-null  float64
 8   pressure          15337 non-null  float64
 9   snow_depth        13900 non-null  float64
dtypes: float64(9), int64(1)
memory usage: 1.2 MB
The strength of data:  15341
```

Hình 6: Thông tin dữ liệu tổng quát

Mô tả dữ liệu

- Bộ dữ liệu bao gồm 10 cột tương đương 10 đặc trưng khác nhau
- Các cột đang có hiện tượng thiếu dữ liệu (Tổng số dữ liệu mỗi cột không đồng đều)
- Độ dài dữ liệu là 15341 (dòng)
- Cột `date` chưa có định dạng đúng (Là *int* thay vì *datetime*)

Ta cần xử lý các giá trị thiếu và chuyển đổi dữ liệu. Trước hết ta sẽ thống kê dữ liệu thiếu và đưa ra giải pháp xử lý các giá trị thiếu.
Thống kê dữ liệu thiếu

Data missing:

	Missing Count	Missing Percent (%)
date	0	0.00
cloud_cover	19	0.12
sunshine	0	0.00
global_radiation	19	0.12
max_temp	6	0.04
mean_temp	36	0.23
min_temp	2	0.01
precipitation	6	0.04
pressure	4	0.03
snow_depth	1441	9.39

Hình 7: Thông tin dữ liệu thiếu trong bộ dữ liệu

Nhận xét

- Đối với bộ dữ liệu có độ dài lớn (15341 dữ liệu), dữ liệu thiếu ở mức thấp (Cao nhất là 9,39% ở cột `snow_depth`. Các đặc trưng còn lại có dữ liệu thiếu chiếm chưa đến 1% trên tổng số dữ liệu.

Xử lý các giá trị thiếu

Có nhiều cách xử lý dữ liệu thiếu trong bài toán dự báo như thay dữ liệu thiếu = 0, bằng trung bình hoặc nội suy. Tuy nhiên, không phải phương pháp nào cũng phù hợp với mô hình nhất định. Ví dụ, nội suy có thể phù hợp với Prophet, nhưng chưa chắc hợp với các mô hình Linear Regression, XGBOOST, ... Do đó, với mỗi mô hình, em sẽ xử lý dữ liệu thiếu sao cho phù hợp với tính chất mô hình nhằm tối ưu hóa mô hình. Mặc dù dữ liệu thiếu không nhiều, nhưng việc xử lý dữ liệu thiếu không phù hợp với mô hình sẽ gây ra tác động ít nhiều đến hiệu suất mô hình. Cách xử lý dữ liệu thiếu sẽ được nói rõ ở mỗi mô hình cụ thể trong phần **4** và **5**.

Dữ liệu sau khi xử lý sẽ được kiểm tra như sau:

Data after handle:

	Missing Count	Missing Percent (%)
date	0	0.0
cloud_cover	0	0.0
sunshine	0	0.0
global_radiation	0	0.0
max_temp	0	0.0
mean_temp	0	0.0
min_temp	0	0.0
precipitation	0	0.0
pressure	0	0.0
snow_depth	0	0.0

Hình 8: Dữ liệu sau xử lý

Có thể thấy, ta đã thành công xử lý dữ liệu thiếu, tránh bất lợi khi xây dựng mô hình và huấn luyện sau này.

3.2 Thống kê dữ liệu mẫu

Ta thực hiện thống kê các chỉ số dữ liệu như trung bình, trung vị,...:

	cloud_cover	sunshine	global_radiation	max_temp	mean_temp	min_temp	precipitation	pressure	snow_depth
count	15322.000000	15341.000000	15322.000000	15335.000000	15305.000000	15339.000000	15335.000000	15337.000000	13900.000000
mean	5.268242	4.350238	118.756951	15.388777	11.475511	7.559867	1.668634	101536.605594	0.037986
std	2.070072	4.028339	88.898272	6.554754	5.729709	5.326756	3.738540	1049.722604	0.545633
min	0.000000	0.000000	8.000000	-6.200000	-7.600000	-11.800000	0.000000	95960.000000	0.000000
25%	4.000000	0.500000	41.000000	10.500000	7.000000	3.500000	0.000000	100920.000000	0.000000
50%	6.000000	3.500000	95.000000	15.000000	11.400000	7.800000	0.000000	101620.000000	0.000000
75%	7.000000	7.200000	186.000000	20.300000	16.000000	11.800000	1.600000	102240.000000	0.000000
max	9.000000	16.000000	402.000000	37.900000	29.000000	22.300000	61.800000	104820.000000	22.000000

Hình 9: Các chỉ số dữ liệu

3.3 Chuyển đổi dữ liệu

1. Chuyển đổi dữ liệu cột `date` sang định dạng `datetime`.
2. Đổi tên các cột theo yêu cầu mô hình
3. Đánh chỉ mục (`index`) theo yêu cầu mỗi mô hình.

4. Chuẩn hóa dữ liệu

4 Đánh giá mô hình

4.1 Lựa chọn các chỉ số đánh giá

Ta sử dụng những chỉ số đánh giá đặc trưng cho những mô hình học máy:

- Hệ số xác định R^2 (R - Squared)
- Sai số trung bình tuyệt đối MAE
- Sai số bình phương trung bình MSE (Hoặc RMSE)
- Sai số phần trăm tuyệt đối trung bình MAPE

4.1.1 Hệ số xác định R^2

Hệ số xác định R^2 là một chỉ số thống kê đo lường mức độ mà mô hình hồi quy tuyến tính giải thích được phương sai của biến phụ thuộc. Nó phản ánh mức độ phù hợp của mô hình: mô hình càng giải thích được nhiều phương sai trong dữ liệu thì R^2 càng cao. Hay nói cách khác, R^2 cho biết bao nhiêu phần trăm biến thiên của biến phụ thuộc Y được giải thích bởi các biến độc lập X.

Công thức tính R^2 là:

$$R^2 = 1 - \frac{SSR}{SST} = \frac{SSM}{SST}$$

Trong đó:

- SSR (Residual Sum of Squares): Tổng bình phương phần dư
- SST (Total Sum of Squares): Tổng bình phương tổng thể
- SSM (Model Sum of Squares): Tổng bình phương do mô hình giải thích

Ngoài ra, với mô hình hồi quy đơn, ta còn có:

$$R^2 = \left(\text{hệ số tương quan Pearson giữa } y \text{ và } \hat{y}\right)^2$$

Giá trị của R^2 nằm trong khoảng từ 0 đến 1:

- $R^2 = 0$: Mô hình không giải thích được gì về biến phụ thuộc
- $R^2 = 1$: Mô hình giải thích hoàn toàn biến thiên của biến phụ thuộc (trường hợp lý tưởng)
- Ví dụ: Nếu $R^2 = 0.85$, nghĩa là 85% phương sai của biến Y được mô hình giải thích

4.1.2 Sai số tuyệt đối trung bình MAE

Sai số tuyệt đối trung bình, ký hiệu là **MAE (Mean Absolute Error)**, là một chỉ số đo lường mức độ sai lệch trung bình giữa giá trị thực tế và giá trị dự đoán của mô hình hồi quy. MAE được tính bằng cách lấy trung bình các sai số tuyệt đối, phản ánh mức độ sai lệch trung bình của mô hình so với thực tế mà không quan tâm đến chiều hướng sai số. Đây là một chỉ số đơn giản, trực quan, và có cùng đơn vị với biến phụ thuộc.

Công thức tính MAE được cho bởi:

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

Trong đó:

- y_i: Giá trị thực tế
- $\hat{y}_i$: Giá trị dự đoán từ mô hình
- n: Số lượng quan sát

Giá trị MAE càng nhỏ thì mô hình dự đoán càng chính xác. Ví dụ, nếu MAE = 2.1 (giả sử đơn vị là triệu đồng), điều đó có nghĩa là, trung bình mỗi dự đoán sai lệch khoảng 2.1 triệu đồng so với giá trị thực tế.

4.1.3 Sai số bình phương trung bình MSE

Sai số bình phương trung bình, ký hiệu là **MSE (Mean Squared Error)**, là một chỉ số phổ biến dùng để đo lường mức độ sai lệch trung bình bình phương giữa các giá trị dự đoán và giá trị thực tế trong mô hình hồi quy. MSE cho biết mức độ phân tán trung bình của sai số bằng cách bình phương độ lệch giữa giá trị dự đoán và giá trị thực tế. Việc bình phương giúp nhấn mạnh các sai số lớn, do đó MSE rất nhạy với các điểm ngoại lai.

Công thức tính MSE được cho bởi:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Trong đó:

- y_i: Giá trị thực tế
- $\hat{y}_i$: Giá trị dự đoán từ mô hình
- n: Số lượng quan sát

Giá trị MSE càng nhỏ thì mô hình càng chính xác. Tuy nhiên, do có bình phương sai số, đơn vị của MSE là bình phương đơn vị của biến phụ thuộc, điều này đôi khi gây khó khăn trong việc diễn giải trực tiếp. Ví dụ, nếu biến đầu ra là "triệu đồng", thì MSE sẽ có đơn vị là "triệu đồng bình phương".

4.1.4 Sai số phần trăm tuyệt đối trung bình MAPE

Sai số phần trăm tuyệt đối trung bình, ký hiệu là **MAPE (Mean Absolute Percentage Error)**, là một chỉ số đánh giá mức độ sai lệch trung bình giữa giá trị dự đoán và giá trị thực tế, được biểu diễn dưới dạng phần trăm. MAPE đo lường sai số tương đối của mô hình, phản ánh mức độ sai lệch trung bình của dự đoán so với thực tế. Đây là một chỉ số dễ hiểu và không phụ thuộc vào đơn vị đo lường của dữ liệu.

Công thức tính MAPE được cho bởi:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Trong đó:

- y_i: Giá trị thực tế
- $\hat{y}_i$: Giá trị dự đoán từ mô hình
- n: Số lượng quan sát

MAPE càng nhỏ thì mô hình càng chính xác. Ví dụ, nếu $MAPE = 6.2\%$, điều đó có nghĩa là mô hình dự đoán sai trung bình 6.2% so với giá trị thực tế. Tuy nhiên, MAPE không xác định nếu có $y_i = 0$, và có thể bị ảnh hưởng lớn khi y_i tiến gần đến 0, khiến cho sai số phần trăm bị khuếch đại. Do đó, MAPE không phù hợp trong các bài toán mà biến mục tiêu có thể bằng hoặc gần bằng 0.

4.2 Mô hình sử dụng

Trong phần này, ta sử dụng mô hình Prophet làm mô hình Baseline.
Thông thường, ta sẽ sử dụng các biến đầu vào như đã trình bày ở phần **2.1**, tuy nhiên, để so sánh hiệu suất giữa một mô hình sử dụng đa biến đầu vào với mô hình đơn biến đầu vào, ở phần này ta tạm thời chưa thêm các biến đầu vào như đã trình bày.
Phương pháp xử lý giá trị thiếu: Phương pháp **Forward Fill** (`ffill`)
Đổi tên các cột theo yêu cầu Prophet:

- `date` -> `ds`
- `mean_temp` -> `y`

Prophet là một mô hình dự báo chuỗi thời gian được phát triển bởi Facebook (nay là Meta) và được công bố lần đầu vào năm 2017. Mục tiêu của Prophet là cung cấp một công cụ dễ sử dụng, hiệu quả và linh hoạt để dự báo các chuỗi thời gian thực tế có tính chất phức tạp như có tính mùa vụ, xu hướng không tuyến tính, và có các dị thường (outliers) hoặc dữ liệu bị thiếu.

Để dự báo nhiệt độ theo thời gian, mô hình Prophet biểu diễn nhiệt độ tại thời điểm t theo cấu trúc cộng như sau:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{1}$$

Trong đó:

- $y(t)$ là nhiệt độ quan sát tại thời điểm t.
- $g(t)$ là thành phần xu hướng (trend), mô tả biến động dài hạn của nhiệt độ như xu hướng nóng lên hoặc lạnh đi theo thời gian.
- $s(t)$ là thành phần mùa vụ (seasonality), phản ánh các chu kỳ biến đổi định kỳ của nhiệt độ như chu kỳ ngày–đêm hoặc mùa trong năm.
- $h(t)$ là thành phần hiệu ứng sự kiện đặc biệt, chẳng hạn các dị thường thời tiết như bão hoặc đợt rét/hạn kéo dài (có thể tùy chọn sử dụng).
- ε_t là nhiễu ngẫu nhiên, biểu diễn phần sai số không giải thích được từ mô hình.

Trong Prophet, xu hướng $g(t)$ có thể được mô hình hóa bằng hàm tuyến tính hoặc logistic, tùy theo đặc điểm giới hạn vật lý của nhiệt độ. Thành phần mùa vụ $s(t)$ được xây dựng từ chuỗi Fourier để phản ánh tính chu kỳ. Prophet cũng có khả năng tự động phát hiện các điểm thay đổi xu hướng (changepoints) và xử lý tốt dữ liệu nhiệt độ thiếu, nhiễu hoặc không đều.
Kích thước tập huấn luyện và tập kiểm tra: 14245 - 1096
Ta đánh giá mô hình thông qua các chỉ số đánh giá cơ bản (Chưa tinh chỉnh, tối ưu hóa):

- **Mean Absolute Error (MAE)**: 2.29 °C
- **Mean Squared Error (MSE)**: 8.31
- **Mean Absolute Percentage Error (MAPE)**: 40.93%
- **Hệ số xác định (R^2)**: 0.76

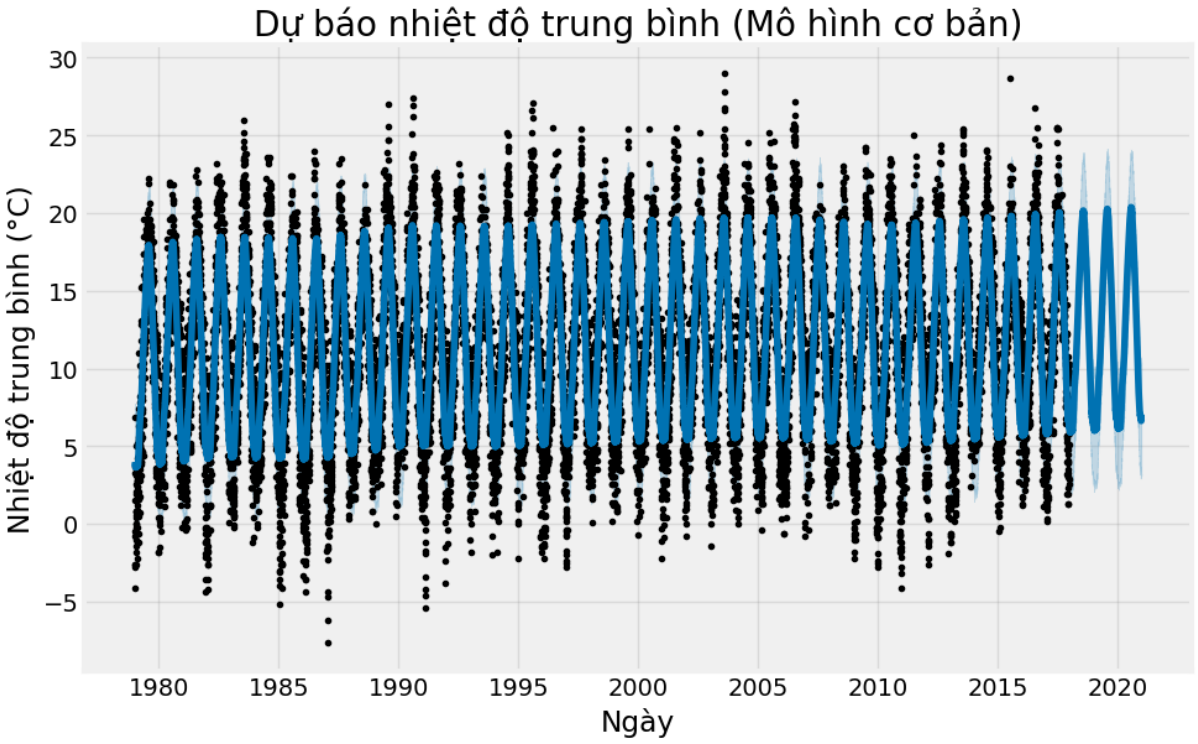
4.3 Thống kê và phân tích lỗi

Sau khi huấn luyện mô hình Prophet cơ bản trên tập huấn luyện và đánh giá trên tập kiểm tra từ năm 2018, các chỉ số đánh giá thu được như sau:

- **Mean Absolute Error (MAE)**: 2.29 °C
- **Mean Squared Error (MSE)**: 8.31
- **Mean Absolute Percentage Error (MAPE)**: 40.93%
- **Hệ số xác định (R^2)**: 0.76

Phân tích lỗi:

- Mặc dù hệ số R^2 đạt 0.76 cho thấy mô hình giải thích được phần lớn phương sai của dữ liệu, tuy nhiên chỉ số MAPE cao (hơn 40%) phản ánh rằng mô hình dự báo chưa chính xác, đặc biệt ở các thời điểm nhiệt độ thấp.
- Giá trị MAE và MSE cho thấy sai số trung bình tuyệt đối khoảng 2.29 °C, và tồn tại những sai lệch lớn tại một số thời điểm bất thường.
- Nguyên nhân chính có thể do mô hình Prophet cơ bản chưa tận dụng các yếu tố ảnh hưởng đến nhiệt độ như độ phủ mây, bức xạ, lượng mưa,... khiến mô hình không theo sát được biến động ngắn hạn.



Hình 10: Mô hình dự báo cơ bản

5 Cải tiến mô hình

Nhằm cải thiện kết quả đánh giá mô hình, cũng như xem xét hiệu suất các mô hình khác nhau đối với bài toán dự báo đặt ra, ta sẽ thử nghiệm với 8 mô hình, bao gồm cả mô hình Prophet đã cải tiến.

5.1 Tổng hợp các mô hình được sử dụng

- Mô hình Prophet (cải tiến)
- Mô hình hồi quy tuyến tính - Linear Regression
- Mô hình Random Forest
- Mô hình XGBOOST
- Mô hình LSTM
- Mô hình SVR
- Mô hình LightGBM
- Mô hình Bayesian Ridge Regression

5.1.1 Mô hình Prophet

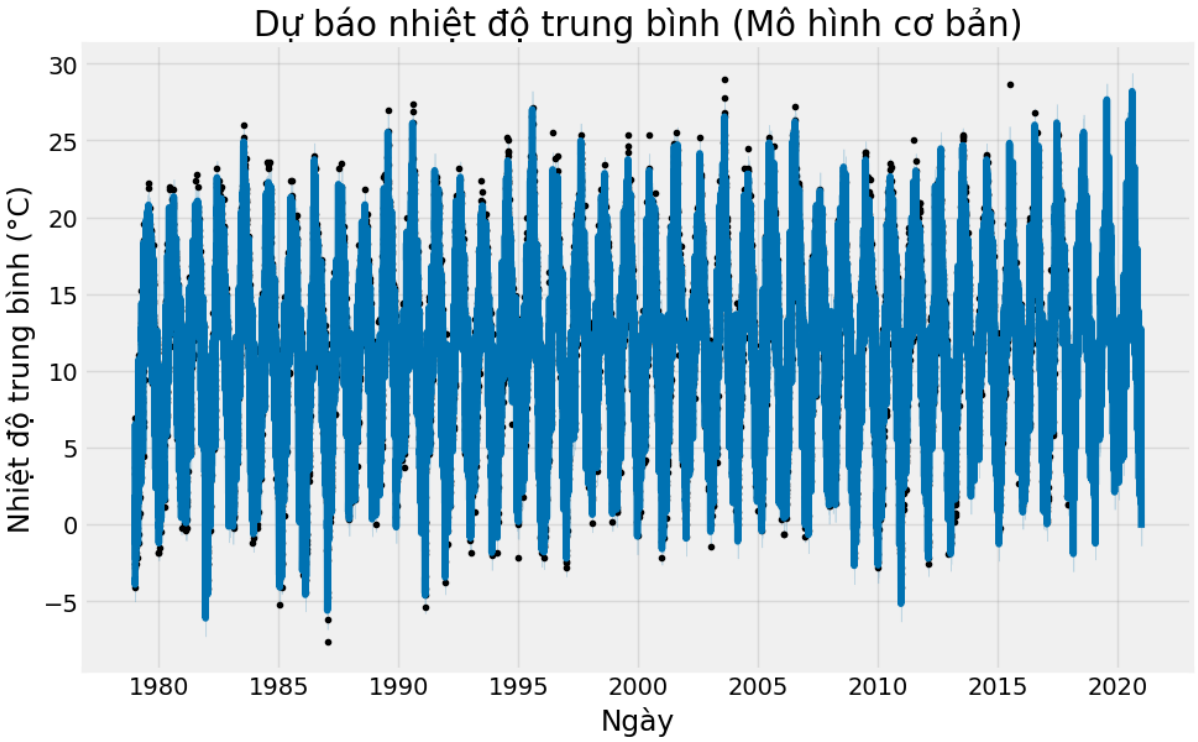
Như nhận xét phía trên, mô hình Prophet dự báo chưa thực sự tốt với những tham số cơ bản. Để cải tiến mô hình, ta sẽ thêm các biến ngoại sinh (Regressors) như phần **2.1** vào mô hình.
Mô hình Prophet có thể được mở rộng bằng cách bổ sung các biến phụ (external regressors), nhằm cải thiện khả năng dự báo khi có thêm thông tin liên quan đến biến mục tiêu. Khi đó, công thức tổng quát của mô hình được mở rộng như sau:

$$y(t) = g(t) + s(t) + h(t) + \sum_{j=1}^{J} \beta_j x_j(t) + \varepsilon_t \tag{2}$$

Trong đó:

- $x_j(t)$ là giá trị của biến phụ thứ j tại thời điểm t, với $j = 1, 2, \ldots, J$.
- β_j là hệ số hồi quy tương ứng với biến phụ $x_j(t)$.

Việc bổ sung các biến phụ như nhiệt độ cực đại, nhiệt độ cực tiểu, độ ẩm, lượng mưa, bức xạ mặt trời, v.v., có thể giúp mô hình học được mối quan hệ tiềm ẩn giữa các yếu tố khí tượng và biến cần dự báo (ví dụ: nhiệt độ trung bình).
Kết quả mô hình nâng cao thu được:



Hình 11: Trực quan hóa kết quả dự báo

Các chỉ số đánh giá:

- **Mean Absolute Error (MAE)**: 0.78
- **Mean Squared Error (MSE)**: 1.17
- **R-squared (R^2)**: 0.97
- **Mean Absolute Percentage Error - MAPE**: 9.64 %

Có thể thấy, mô hình đã cải thiện kết quả rất tốt thông qua việc thêm vào các biến ngoại sinh.

5.1.2 Mô hình hồi quy tuyến tính – Linear Regression

Hồi quy tuyến tính là một trong những phương pháp cơ bản và phổ biến nhất trong thống kê và học máy, được sử dụng để mô hình hóa mối quan hệ tuyến tính giữa một biến phụ thuộc (biến mục tiêu) và một hoặc nhiều biến độc lập (biến giải thích).

Mô hình tổng quát

Mô hình hồi quy tuyến tính nhiều biến được biểu diễn dưới dạng:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, 2, \ldots, n \tag{3}$$

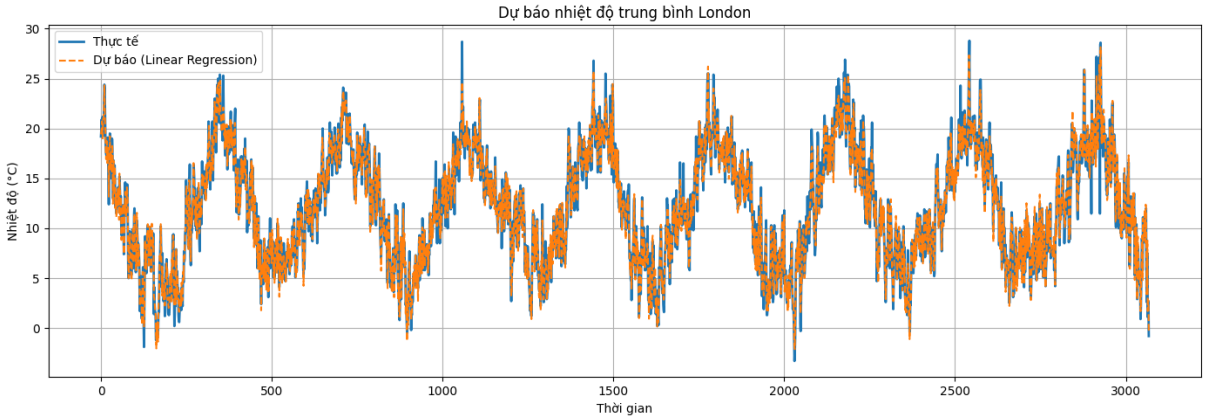
Hoặc dưới dạng ma trận:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \tag{4}$$

Trong đó:

- y_i: giá trị của biến phụ thuộc tại quan sát thứ i.
- x_{ij}: giá trị của biến độc lập thứ j tại quan sát thứ i.
- β_0: hệ số chặn (intercept).
- β_j: hệ số hồi quy tương ứng với biến x_j, phản ánh mức độ ảnh hưởng của biến đó lên y.
- ε_i: sai số ngẫu nhiên, thường giả định tuân theo phân phối chuẩn $\mathcal{N}(0, \sigma^2)$.
- $\mathbf{y} \in \mathbb{R}^{n\times 1}$: vector các giá trị mục tiêu.
- $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$: ma trận thiết kế (design matrix) bao gồm một cột giá trị 1 (cho hệ số chặn) và các biến giải thích.
- $\beta \in \mathbb{R}^{(p+1)\times 1}$: vector hệ số hồi quy.

Phương pháp xử lý giá trị thiếu: Điền trung bình
Ở mô hình hồi quy tuyến tính, ta sẽ tạo thêm 3 đặc trưng độ trễ cho biến mục tiêu làm đầu vào (Input) nhằm tối ưu hóa thêm đầu vào.
Chia tập huấn luyện: Huấn luyện mô hình theo tỷ lệ 80 - 20.
Trực quan hóa dự báo



Hình 12: Trực quan hóa kết quả dự báo

Các chỉ số đánh giá mô hình

- **Mean Absolute Error (MAE)**: 0.696
- **Mean Squared Error (MSE)**: 0.868
- **R-squared (R^2)**: 0.973
- **Mean Absolute Percentage Error - MAPE**: 9.672 %

5.1.3 Mô hình Random Forest

Random Forest là một mô hình học máy thuộc nhóm phương pháp **ensemble learning** (học tập tổ hợp), cụ thể là dạng *bagging* của cây quyết định (decision trees). Mô hình được giới thiệu bởi Leo Breiman vào năm 2001 và đã chứng minh hiệu quả vượt trội trong nhiều bài toán phân loại và hồi quy.

Nguyên lý hoạt động

Ý tưởng chính của Random Forest là xây dựng một tập hợp các cây quyết định (decision trees) huấn luyện trên các mẫu con khác nhau của dữ liệu (tạo ra từ kỹ thuật bootstrapping), sau đó tổng hợp kết quả của các cây để đưa ra dự đoán cuối cùng.

Trong bài toán hồi quy, dự báo của Random Forest được tính trung bình từ dự báo của từng cây con:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(\mathbf{x}) \tag{5}$$

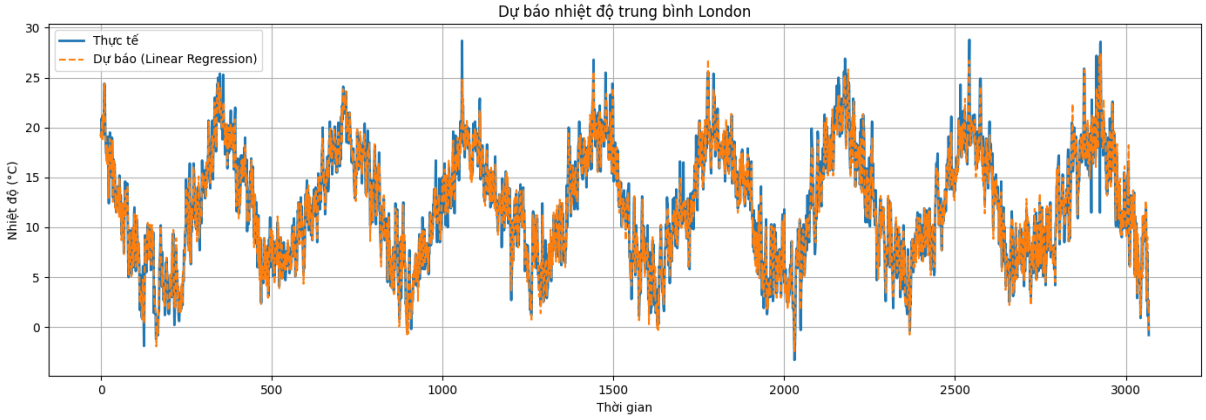
Trong đó:

- T: số lượng cây trong rừng.
- $h_t(\mathbf{x})$: đầu ra dự báo của cây quyết định thứ t với đầu vào $\mathbf{x}$.
- $\hat{y}$: giá trị dự đoán trung bình cuối cùng.

Phương pháp xử lý giá trị thiếu: Điền giá trị trung bình như hồi quy tuyến tính.

Tương tự, ta cũng tạo độ trễ làm đặc trưng đầu vào như hồi quy tuyến tính **5.1.2**.

Trực quan hóa kết quả dự báo



Hình 13: Trực quan hóa kết quả dự báo

Các chỉ số đánh giá

- **Mean Absolute Error (MAE)**: 0.706
- **Mean Squared Error (MSE)**: 0.945

- **R-squared (R^2)**: 0.971
- **Mean Absolute Percentage Error - MAPE**: 10.141 %

5.1.4 Mô hình XGBOOST

XGBoost (Extreme Gradient Boosting) là một mô hình học máy tiên tiến thuộc nhóm **ensemble learning**, được phát triển bởi Tianqi Chen và Carlos Guestrin vào năm 2016. XGBoost là một cải tiến hiệu quả và mạnh mẽ của kỹ thuật *Gradient Boosting Decision Tree (GBDT)*, nổi bật nhờ khả năng huấn luyện nhanh, chính xác và tối ưu tài nguyên.

Nguyên lý hoạt động

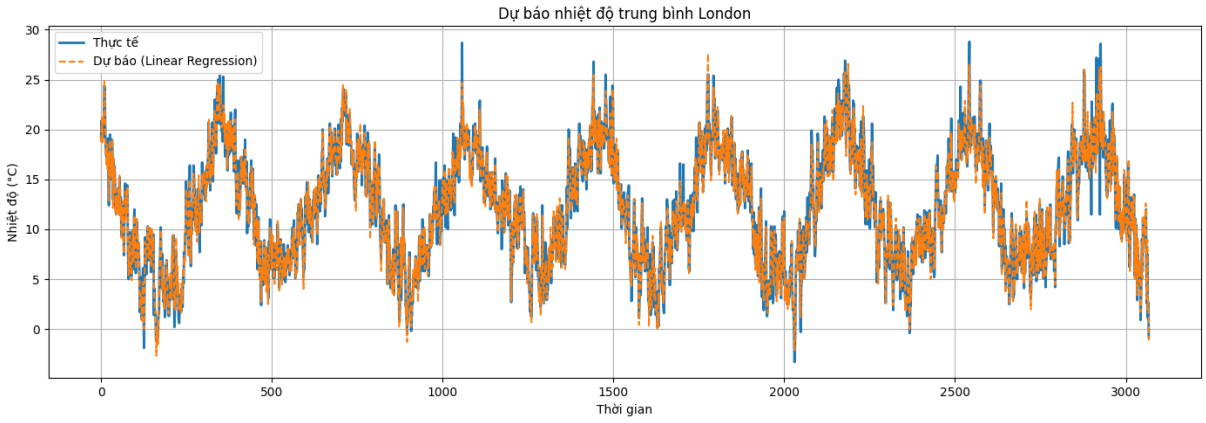
XGBoost hoạt động theo nguyên lý boosting – xây dựng mô hình bằng cách huấn luyện liên tiếp các cây quyết định sao cho mỗi cây mới tập trung vào việc sửa lỗi của các cây trước. Mục tiêu là tối thiểu hóa hàm mất mát theo hướng đạo hàm bậc hai (*second-order Taylor expansion*):

$$\mathscr{L}^{(t)} = \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \tag{6}$$

Trong đó:

- l: hàm mất mát (ví dụ: MSE cho hồi quy).
- $g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}$: gradient.
- $h_i = \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$: hessian (đạo hàm bậc hai).
- f_t: cây quyết định tại vòng lặp thứ t.
- $\Omega(f_t)$: hàm phạt độ phức tạp của cây, giúp tránh overfitting.

Trực quan hóa kết quả dự báo



Hình 14: Trực quan hóa kết quả dự báo

Các chỉ số đánh giá mô hình

- **Mean Absolute Error (MAE)**: 0.734
- **Mean Squared Error (MSE)**: 1.024
- **R-squared (R^2)**: 0.968
- **Mean Absolute Percentage Error - MAPE**: 10.357 %

5.1.5 Mô hình LSTM

LSTM (Long Short-Term Memory) là một kiến trúc mạng nơ-ron hồi tiếp (Recurrent Neural Network – RNN) được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997. Mô hình này được thiết kế để giải quyết nhược điểm **vanishing gradient** của RNN truyền thống khi xử lý chuỗi dài, nhờ đó học được các mối quan hệ lâu dài trong dữ liệu chuỗi thời gian.

Kiến trúc LSTM cơ bản

Một khối (cell) LSTM gồm 3 cổng chính:

- **Forget gate** f_t: xác định phần nào của trạng thái trước đó nên được quên.
- **Input gate** i_t: xác định thông tin mới nào sẽ được lưu vào trạng thái.
- **Output gate** o_t: xác định thông tin nào từ trạng thái ẩn sẽ được truyền ra ngoài.

Công thức toán học cho một khối LSTM tại thời điểm t:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{9}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{10}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$
$$h_t = o_t \odot \tanh(C_t) \tag{12}$$

Trong đó:

- x_t: đầu vào tại thời điểm t.
- h_t: trạng thái ẩn (hidden state).
- C_t: trạng thái bộ nhớ (cell state).
- W và b: các trọng số và hệ số điều chỉnh.
- σ: hàm sigmoid, $\odot$: phép nhân từng phần tử (Hadamard product).

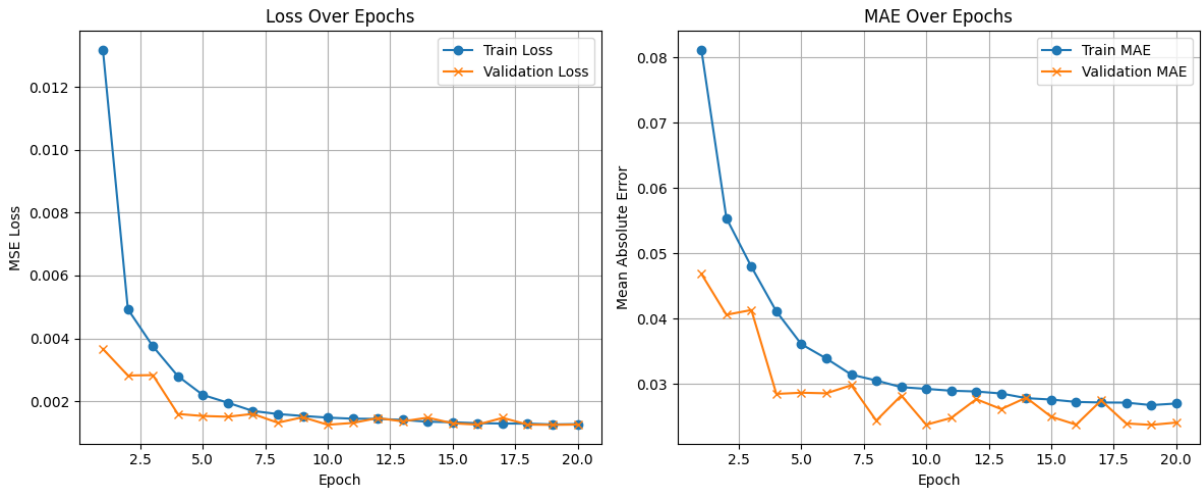
Ở LSTM, ta không tạo đặc trưng độ trễ mà sử dụng các đặc trưng có sẵn trong bộ dữ liệu, như đã trình bày ở **2.1**.

Sử dụng `MinMaxScaler()` để chuẩn hóa dữ liệu.

Chia tập huấn luyện: 70 (Train) - 15 (Validation) - 15 (Test).

Ta sử dụng thêm tối ưu Adam cho mô hình, chọn hàm mất mát MSE.

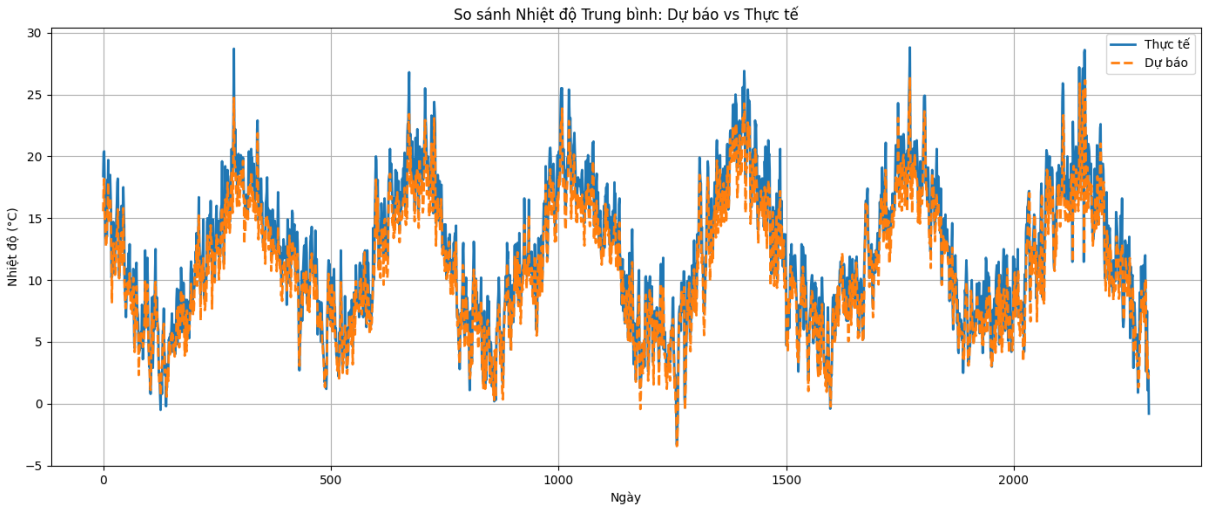
Trực quan hóa hàm mất mát và quá trình huấn luyện Nhận xét: Mô hình hội tụ tốt, hàm



Hình 15: Đánh giá quá trình huấn luyện

mất mát giảm dần và rất thấp trong quá trình huấn luyện.

Trực quan hóa kết quả dự báo



Hình 16: Trực quan hóa kết quả dự báo

Các chỉ số đánh giá mô hình

- **Mean Absolute Error (MAE)**: 1.4609
- **Mean Squared Error (MSE)**: 3.072

- **R-squared (R^2)**: 0.9021
- **Mean Absolute Percentage Error - MAPE**: 14.73 %

5.1.6 Mô hình SVR

Support Vector Regression (SVR) là một biến thể của Support Vector Machine (SVM), được thiết kế để giải quyết các bài toán hồi quy. Thay vì phân loại, SVR dự đoán một giá trị thực thông qua việc tìm một siêu phẳng (hyperplane) phù hợp nhất với dữ liệu trong không gian đặc trưng.

Nguyên lý hoạt động

Mục tiêu chính của SVR là tìm một hàm $f(x)$ sao cho sai số dự đoán không vượt quá một ngưỡng ε với hầu hết các điểm dữ liệu, đồng thời đảm bảo độ phức tạp của mô hình là nhỏ nhất. Hàm hồi quy có dạng:

$$f(x) = \langle w, \phi(x) \rangle + b \tag{13}$$

Trong đó:

- w: vector trọng số.
- b: hệ số điều chỉnh (bias).
- $\phi(x)$: ánh xạ dữ liệu đầu vào sang không gian đặc trưng có chiều cao hơn.
- $\langle \cdot, \cdot \rangle$: tích vô hướng.

Hàm mục tiêu cần tối ưu là:

$$\min_{w,\xi,\xi^*} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{14}$$

$$\text{với điều kiện:} \quad \begin{cases} y_i - \langle w, \phi(x_i) \rangle - b \leq \varepsilon + \xi_i \\ \langle w, \phi(x_i) \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{15}$$

Trong đó:

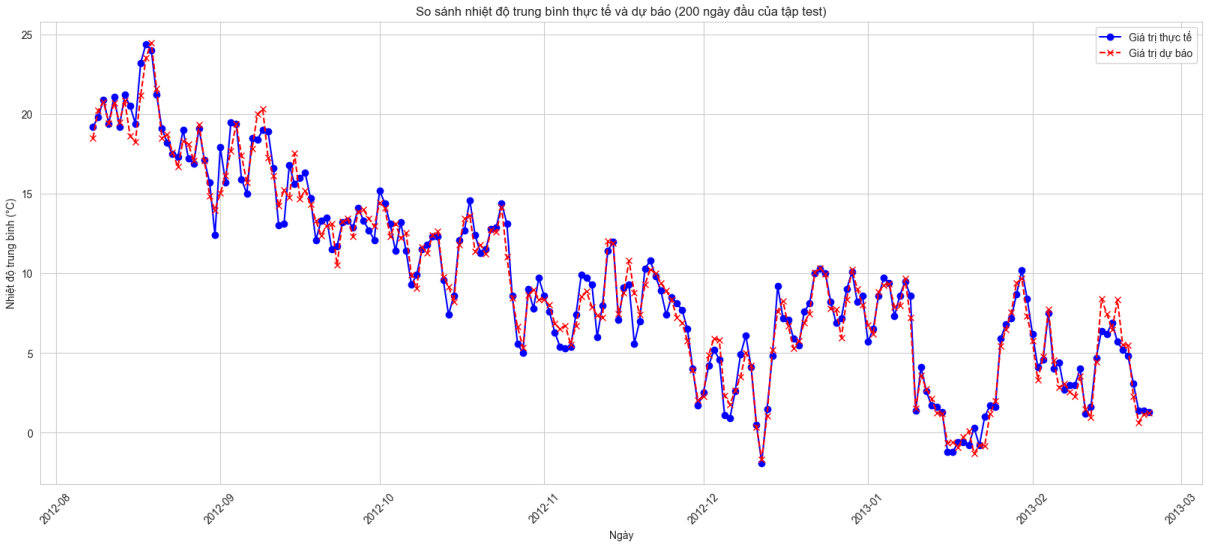
- ε: ngưỡng sai số cho phép.
- ξ_i, ξ_i^*: biến trễ dùng khi điểm dữ liệu nằm ngoài biên ε.
- C: hệ số điều chỉnh giữa độ chính xác và độ phức tạp của mô hình.

Các hàm kernel phổ biến

SVR sử dụng các hàm kernel để ánh xạ dữ liệu phi tuyến sang không gian tuyến tính, bao gồm:

- Hàm tuyến tính: $K(x_i, x_j) = x_i^T x_j$
- Hàm Gaussian RBF: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
- Hàm polynomial: $K(x_i, x_j) = (x_i^T x_j + r)^d$

Ta sử dụng đầu vào và xử lý dữ liệu tương tự các mô hình trên, không tạo đặc trưng độ trễ.
Chia tập huấn luyện theo tỉ lệ 80 – 20
Chuẩn hóa dữ liệu bằng `StandardScaler()`
Để làm rõ hơn xu hướng và mức độ chính xác, ta thực hiện trực quan ít hơn kết quả dự báo (Chỉ mang tính chất trực quan rõ ràng hơn)



Hình 17: Trực quan hóa kết quả dự báo

Các chỉ số đánh giá

- **Mean Absolute Error (MAE)**: 0.90
- **Mean Squared Error (MSE)**: 1.81
- **R-squared (R^2)**: 0.94
- **Mean Absolute Percentage Error - MAPE**: 11.29 %

5.1.7 Mô hình LightGBM

LightGBM (Light Gradient Boosting Machine) là một framework học máy mạnh mẽ do Microsoft phát triển, dựa trên thuật toán boosting theo dạng cây quyết định. Mô hình này được tối ưu hóa để xử lý các bài toán hồi quy, phân loại và xếp hạng với tốc độ huấn luyện nhanh và khả năng mở rộng cao.
Nguyên lý hoạt động

Tương tự như các mô hình boosting khác như XGBoost, LightGBM xây dựng mô hình tổng hợp bằng cách kết hợp nhiều cây quyết định (decision trees) tuần tự, trong đó mỗi cây mới học trên phần dư (residual) của cây trước để giảm sai số dự đoán.

Một số đặc điểm nổi bật

- **GOSS (Gradient-based One-Side Sampling)**: Lọc bớt những điểm có gradient nhỏ (ít thông tin) và giữ lại các điểm có gradient lớn để huấn luyện, từ đó giảm chi phí tính toán mà vẫn đảm bảo độ chính xác.
- **EFB (Exclusive Feature Bundling)**: Gộp các đặc trưng hiếm khi kích hoạt đồng thời thành một nhóm nhằm giảm số chiều của dữ liệu, tăng hiệu suất xử lý.
- **Leaf-wise tree growth**: LightGBM phát triển cây theo chiều sâu (leaf-wise), chọn nhánh lá có độ giảm lỗi lớn nhất để chia tách tiếp, giúp tăng độ chính xác so với phương pháp level-wise (chia đều theo tầng) của các thuật toán khác.

Hàm mục tiêu và tối ưu hóa

Hàm mất mát tổng quát của LightGBM có dạng:

$$\mathscr{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{16}$$

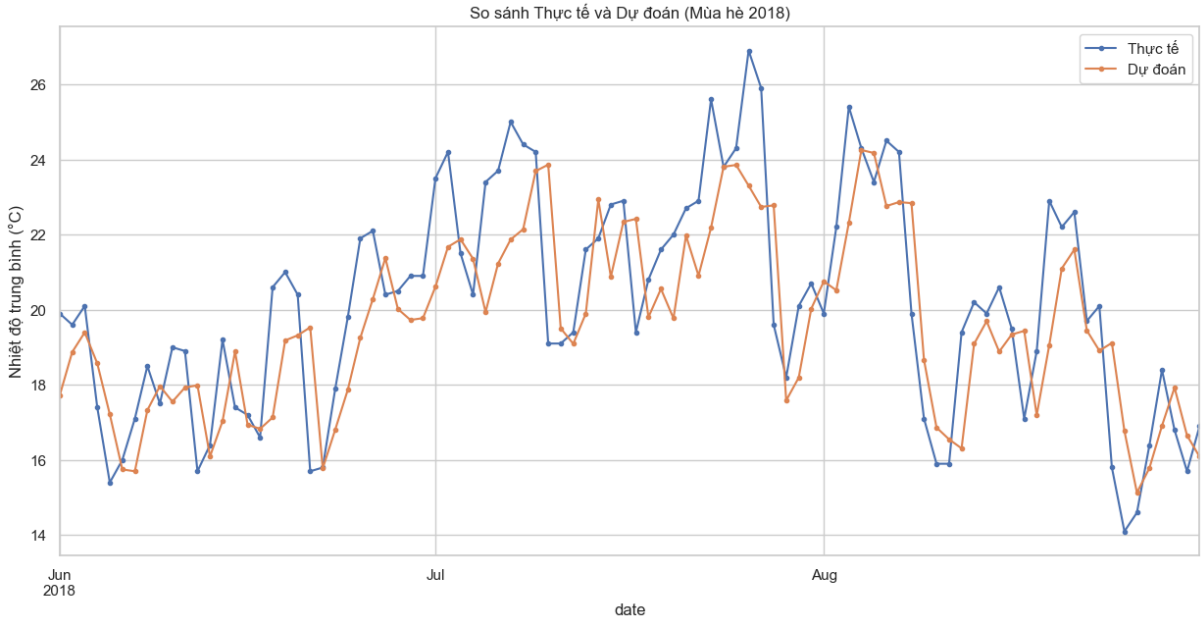
Trong đó:

- $l(\cdot)$: hàm mất mát (ví dụ: MSE cho hồi quy).
- $\hat{y}_i^{(t-1)}$: dự đoán sau $(t-1)$ cây.
- f_t: cây quyết định tại vòng thứ t.
- $\Omega(f_t)$: thành phần điều chuẩn, thường dùng để giảm overfitting.

Xử lý giá trị thiếu bằng phương pháp nội suy.

Thực hiện tạo một số đặc trưng mới cho cửa sổ trượt (Rolling Window):

- Ngày
- Tháng
- Năm
- Ngày trong tuần
- Ngày trong năm
- Tuần trong năm



Hình 18: Trực quan hóa kết quả dự báo

Ta sẽ bỏ cột `date` khỏi tập huấn luyện sau khi phân chia.
Huấn luyện đến cuối năm 2017 và dự báo từ 2018.
Trực quan hóa kết quả dự báo
Các chỉ số đánh giá mô hình

- **Mean Absolute Error (MAE)**: 1.55
- **Mean Squared Error (MSE)**: 3.87
- **R-squared (R^2)**: 0.89

5.1.8 Mô hình Bayesian Ridge Regression

Bayesian Ridge Regression (Hồi quy Ridge theo quan điểm Bayes) là một biến thể của hồi quy tuyến tính, trong đó các tham số của mô hình được xem là các biến ngẫu nhiên có phân phối xác suất thay vì các giá trị cố định. Mô hình kết hợp giữa hồi quy Ridge truyền thống và suy luận Bayes, cho phép mô hình thể hiện độ không chắc chắn (uncertainty) trong dự đoán.
Công thức tổng quát
Bayesian Ridge giả định mối quan hệ tuyến tính giữa biến đầu vào $\mathbf{X}$ và đầu ra $\mathbf{y}$ dưới dạng:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I}) \tag{17}$$

Trong đó:

- $\mathbf{X}$ là ma trận đặc trưng (feature matrix).
- $\mathbf{w}$ là vector hệ số hồi quy (tham số cần ước lượng).

- ε là nhiễu Gaussian có phương sai α^{-1}.

Thay vì ước lượng điểm cho $\mathbf{w}$, mô hình giả định:

$$\mathbf{w} \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}) \tag{18}$$

Ước lượng tham số

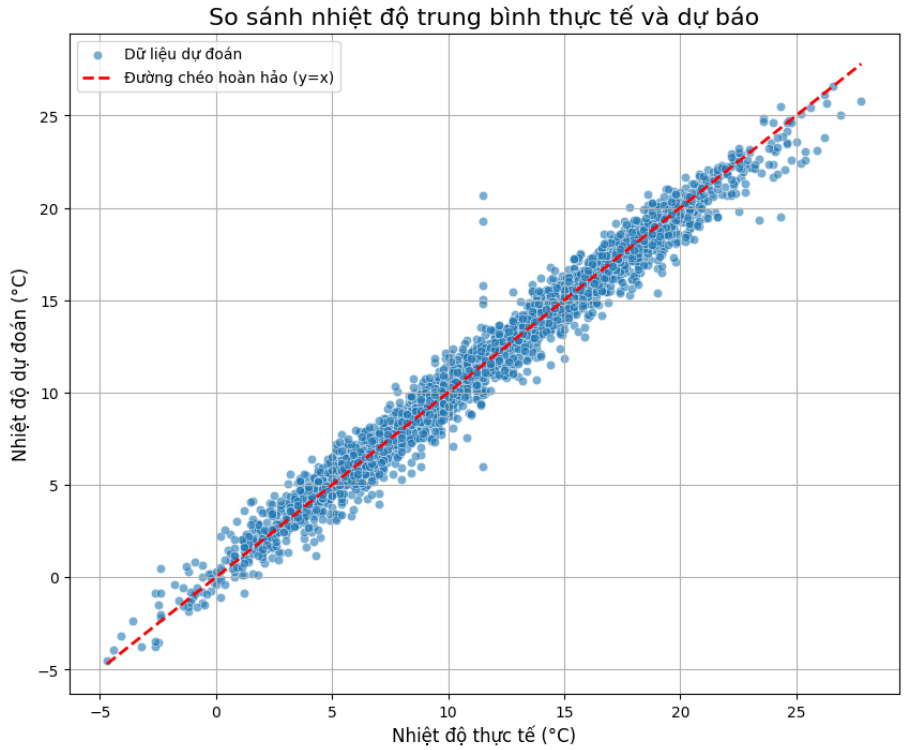
Quá trình huấn luyện sử dụng luật Bayes để cập nhật phân phối hậu nghiệm của $\mathbf{w}$ dựa trên dữ liệu quan sát. Các tham số α (precision của nhiễu) và λ (precision của prior) cũng được ước lượng bằng phương pháp tối đa hóa log-likelihood hoặc lấy kỳ vọng (Empirical Bayes).
Phân phối hậu nghiệm của $\mathbf{w}$ là:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \alpha, \lambda) \sim \mathcal{N}(\mu_w, \Sigma_w) \tag{19}$$

Trong đó:

$$\Sigma_w = (\lambda\mathbf{I} + \alpha\mathbf{X}^\top\mathbf{X})^{-1}$$
$$\mu_w = \alpha\Sigma_w\mathbf{X}^\top\mathbf{y}$$

Chia tập huấn luyện: 80 – 20
Trong phần này, ta sẽ trực quan bằng đường chéo hoàn hảo và xem mức độ phân bố dữ liệu dự báo và thực tế. **Các chỉ số đánh giá**



Hình 19: So sánh giữa thực tế và dự báo

- **Mean Absolute Error (MAE)**: 0.69

- **Mean Squared Error (MSE)**: 0.84
- **R-squared (R^2)**: 0.97
- **Mean Absolute Percentage Error - MAPE**: 10.78%

6 Đóng gói mô hình

6.1 Mô hình tiên tiến được sử dụng (3 năm trở lại đây)

Mô hình tiên tiến được sử dụng (3 năm trở lại) mà em lựa chọn là mô hình LightGBM và XGBOOST. MẶc dù ra mắt năm 2016, nhưng phiên bản mới của 2 mô hình đã được cập nhật trong khoảng 3 năm trở lại đây.

Tổng quan các cải tiến của LightGBM và XGBoost (2022–2025)

Trong vòng ba năm trở lại đây, hai mô hình học máy phổ biến là **LightGBM** và **XGBoost** đã có nhiều bước tiến vượt bậc cả về mặt thuật toán lẫn ứng dụng thực tiễn. Dưới đây là một số điểm nổi bật:

6.1.1 LightGBM

- **TDA-LightGBM**: Han Yang et al. (2023) đã kết hợp LightGBM với Topological Data Analysis (TDA) nhằm tăng cường khả năng kháng nhiễu khi xử lý ảnh, cho kết quả chính xác cao hơn khoảng 3% so với LightGBM truyền thống.
- **Hybrid-LightGBM trong Y học**: Xiaoyan Sun et al. (2025) tích hợp Gradient Harmonization Loss, Jacobian Regularization và Whale Optimization để huấn luyện LightGBM trong chẩn đoán ung thư vú, cải thiện độ nhạy với dữ liệu mất cân bằng.
- **Ứng dụng trong khai thác khoáng sản**: Một nghiên cứu năm 2024 trên *Scientific Reports* kết hợp LightGBM với thuật toán Enhanced Whale Optimization Algorithm (EWOA) và chiến lược cân bằng dữ liệu, giúp tăng Recall và F1-Score trong bài toán phân loại đá.
- **Tối ưu siêu tham số và giải thích mô hình**: Nghiên cứu trên *Springer* (2025) đã sử dụng Optuna để tối ưu siêu tham số và kết hợp SHAP để phân tích tầm quan trọng biến, giúp tăng hiệu suất và khả năng giải thích của mô hình.

6.1.2 XGBoost

- **XGBoost-NDT**: Puli Vilash & Mohd Abdul Hameed (2025) mở rộng XGBoost với cây quyết định dạng neural (Neural Decision Tree), tăng độ chính xác từ 82% lên 90% trong dự đoán sảy thai.
- **Tối ưu hóa đa mục tiêu**: Một nghiên cứu sử dụng kết hợp thuật toán mô phỏng tôi luyện (Simulated Annealing) và di truyền (Genetic Algorithm) để tối ưu XGBoost trong dự đoán độ thấm địa chất.
- **XGBoost phiên bản 3.0.0 (2025)**: Bản cập nhật cải thiện khả năng xử lý dữ liệu lớn qua External Memory, hỗ trợ tốt hơn cho biến phân loại và thêm các phương pháp hồi quy như Quantile Regression.

- **Ứng dụng trong an ninh mạng**: CrowdStrike phát triển hàm mất mát tùy chỉnh nhằm giảm false positive/false negative trong hệ thống phát hiện mối đe dọa dùng XGBoost.

Kết luận
Các cải tiến trong LightGBM và XGBoost từ 2022–2025 đã giúp hai mô hình này:

- Tăng hiệu quả xử lý dữ liệu lớn và mất cân bằng.
- Mở rộng sang các ứng dụng thực tế như y học, địa chất, an ninh mạng.
- Nâng cao khả năng diễn giải mô hình với SHAP và các công cụ phân tích đặc trưng.

Những bước tiến này củng cố vị thế hàng đầu của hai mô hình trong các bài toán hồi quy và phân loại hiện đại.

6.2 Khả năng ứng dụng vào 1 ngữ cảnh cụ thể

Một trong những mô hình tiên tiến nổi bật hiện nay là **LightGBM kết hợp tối ưu siêu tham số bằng Optuna** và giải thích mô hình với **SHAP (SHapley Additive exPlanations)**. Mô hình này đặc biệt phù hợp cho các bài toán phân tích hành vi khách hàng trong lĩnh vực **ngân hàng** và **tài chính tiêu dùng**.

6.2.1 Bối cảnh ứng dụng cụ thể

- **Bài toán nghiệp vụ:** Dự đoán khả năng khách hàng sẽ *vỡ nợ (default)* trong 6 tháng tới, từ đó hỗ trợ ra quyết định cấp tín dụng.
- **Người sử dụng:** Các chuyên viên phân tích rủi ro tín dụng tại ngân hàng hoặc công ty tài chính.
- **Nguồn dữ liệu:** Hồ sơ tín dụng của khách hàng (thu nhập, lịch sử trả nợ, dư nợ hiện tại, tần suất sử dụng thẻ, v.v.)

6.2.2 Ưu điểm mô hình

- **Hiệu quả cao:** LightGBM cho tốc độ huấn luyện nhanh, xử lý tốt dữ liệu không đồng nhất và mất cân bằng.
- **Tối ưu tự động:** Optuna tìm siêu tham số tối ưu như learning rate, số cây, độ sâu cây mà không cần thử thủ công.
- **Giải thích dễ dàng:** SHAP giúp phân tích rõ ràng vì sao mô hình đánh giá khách hàng A có khả năng vỡ nợ cao — nhờ đó hỗ trợ giải trình trước kiểm toán và các bộ phận kiểm soát rủi ro.

Ý nghĩa nghiệp vụ Mô hình này giúp ngân hàng:

- Giảm tỷ lệ nợ xấu bằng cách từ chối cấp tín dụng cho khách hàng có nguy cơ cao.
- Tối ưu hóa chiến lược marketing và chính sách lãi suất đối với từng phân khúc khách hàng.
- Nâng cao uy tín khi ra quyết định dựa trên mô hình có thể giải thích rõ ràng và công khai.

6.3 Khả năng ứng dụng vào thực tế

Mô hình LightGBM được áp dụng vào bài toán dự báo nhiệt độ trung bình tại London đã cho thấy hiệu quả cao thông qua các chỉ số đánh giá như sau:

- **Mean Absolute Error (MAE)**: 1.55 °C
- **Mean Squared Error (MSE)**: 3.87
- **Hệ số xác định (R-squared - R^2)**: 0.89

Kết quả này cho thấy mô hình đạt độ chính xác tốt, với sai số trung bình thấp và khả năng giải thích được 89% phương sai trong dữ liệu thực tế. Với $R^2 > 0.85$ và MAE < 2 °C, mô hình đủ điều kiện để triển khai trong các hệ thống dự báo thời tiết thực tế.
Khả năng ứng dụng: Mô hình có thể áp dụng trong các ngữ cảnh sau:

- *Cơ quan khí tượng*: sử dụng để dự báo nhiệt độ phục vụ cảnh báo thời tiết.
- *Ngành nông nghiệp*: hỗ trợ lập kế hoạch gieo trồng, tưới tiêu, thu hoạch.
- *Logistics và vận tải*: tối ưu hóa lộ trình vận chuyển theo điều kiện thời tiết.

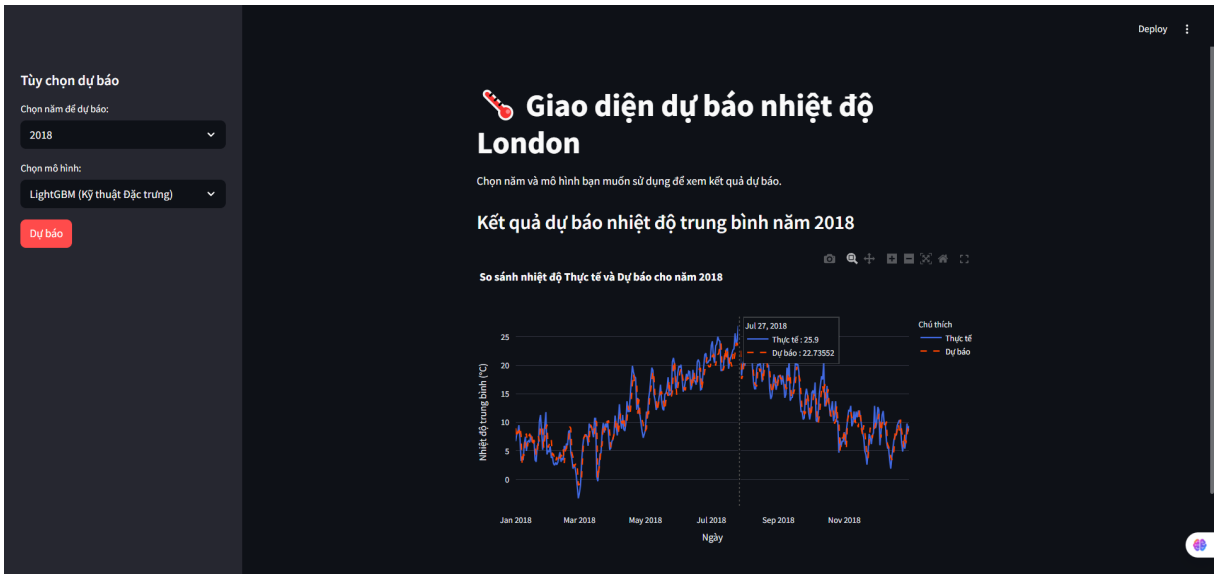
Như vậy, LightGBM không chỉ hiệu quả về mặt mô hình hóa mà còn có tiềm năng cao trong triển khai thực tế với dữ liệu thời tiết.

6.4 Đóng gói giao diện demo chương trình

Dưới đây là giao diện dự báo thời tiết (Nhiệt độ trung bình) của London:
Các bước dự báo trong giao diện như sau:

1. Chọn năm muốn thực hiện dự báo
2. Chọn mô hình muốn sử dụng để dự báo
3. Dự báo

Sau khi chọn, giao diện sẽ hiện biểu đồ dự báo và thực tế như hình. Khi ta di chuyển chuột vào 1 điểm trên biểu đồ, ta sẽ có thông tin dữ liệu nhiệt độ dự báo và nhiệt độ chính xác.



Hình 20: Giao diện dự báo thời tiết London

Hiện tại giao diện có 2 mô hình chủ yếu để minh họa là LightGBM và Bayesian Ridge Regression

7 Kết luận

7.1 Checklist công việc

STT	STT	Loại yêu cầu	Yêu cầu	Điểm chữ	Điểm số	Check	Minh chứng
1	1	Xử lý dữ liệu (2 điểm)	Mô tả bài toán, đầu vào, đầu ra, yêu cầu xử lý	A	1	X	Trang 4 - 6
2	2		Đánh nhãn & Tiền xử lý dữ liệu	A	1	X	Trang 8 - 10
3	3		Thống kê dữ liệu mẫu	A	1	X	Trang 10
4	4		Xử lý mất cân bằng dữ liệu (cho bài toán phân lớp) hoặc Chuyển đổi dữ liệu (cho bài toán hồi quy)	A	1	X	Trang 10 -11 Có trình bày chuyển đổi dữ liệu ở mỗi mô hình
5	5	Đánh giá mô hình (1 điểm)	Đề xuất và lựa chọn các tiêu chí đánh giá (về độ chính xác, tốc độ, khả năng ứng dụng,...)	A	1	X	Trang 12 - 14
6	6		Thống kê và phân tích lỗi	A	1	X	Trang 15 - 16
7	7	Cải tiến mô hình (4 điểm)	Kiến trúc mô hình 1	A	1	X	Trang 14 - 15 Trang 17 - 18 (Cải tiến)
8	8		Kiến trúc mô hình 2	A	1	X	Trang 18 - 19
9	9		Kiến trúc mô hình 3	A	1	X	Trang 20 - 21
10	10		Kiến trúc mô hình 4	A	1	X	Trang 21 - 22
11	11		Kiến trúc mô hình 5	A	1	X	Trang 22 - 23
12	12		Kiến trúc mô hình 6	A	1	X	Trang 24 - 25
13	13		Kiến trúc mô hình 7	A	1	X	Trang 25 - 27
14	14		Kiến trúc mô hình 8	A	1	X	Trang 27 - 29
15	15	Đóng gói mô hình (3 điểm)	Có sử dụng mô hình tiên tiến trong 3 năm trở lại đây (chỉ ra paper liên quan)	A	1	X	Trang 21 - 22 Trang 25 - 27 Trang 30 - 31
16	16		Có khả năng ứng dụng vào một ngữ cảnh cụ thể (ứng dụng vào bài toán nghiệp vụ nào, ai là người sử dụng)	A	1	X	Trang 31 - 32
17	17		Các chỉ số đánh giá mô hình đủ điều kiện để ứng dụng vào thực tế	A	1	X	Trang 32
18	18		Đóng gói giao diện demo chương trình	A	1	X	Trang 32 - 33
19	19		Làm slide báo cáo	A	1	X	
20	20		Thuyết trình trên lớp	A	1	X	
Tổng điểm / 20						**20**	
Tổng điểm / 10						**10**	

Thống kê nhiệm vụ đã được miêu tả như trong bảng.

7.2 Bảng mô tả chi tiết mô hình

BẢNG MÔ TẢ CHI TIẾT MÔ HÌNH

STT	Tên mô hình	Điều kiện dừng	Phương pháp tối ưu hóa siêu tham số	Siêu tham số của mô hình		Kết quả đánh giá trên dữ liệu test theo các chỉ số	Chú giải
				Tên	Mô tả		
1	Prophet	Prophet sử dụng tối ưu hóa Maximum A Posteriori (MAP) với backend Stan, và dừng khi quá trình tối ưu hội tụ — tức là khi gradient đủ nhỏ hoặc sau số vòng lặp tối đa nội bộ.	Đặt thủ công	(1) changepoint_prior_scale (2) sesonality_mode (3) yearly_seasonality (4) weekly_seasonality (5) daily_seasonality	(1) Kiểm soát độ linh hoạt khi mô hình phát hiện các điểm thay đổi trong xu hướng. (2) Mùa vụ được cộng tuyến tính vào xu hướng. (3) Mùa vụ theo năm (4) Mùa vụ theo tuần (5) Mùa vụ theo ngày	MAE: 0.78 MSE: 1.17 R2_SCORE: 0.97 MAPE: 9.64%	Mô hình baseline
2	Linear Regression	Không sử dụng điều kiện dừng dựa trên vòng lặp hay hội tụ vì nó giải bài toán hồi quy bằng cách tính nghiệm đóng (closed-form solution) thông qua phép toán đại số tuyến tính	Đặt thủ công	(1) fit_intercept (2) normalize (3) copy_X (4) n_jobs	(1) Học hệ số bias/ intercept (2) Chuẩn hóa dữ liệu trước khi huấn luyện (3) Sao chép dữ liệu đầu vào (4) Số lượng luồng xử lý song song khi huấn luyện	MAE: 0.696 MSE: 0.868 R2_SCORE: 0.973 MAPE: 9.672%	Baseline tuyến tính, đơn giản, dễ diễn giải.
3	Random Forest	Mặc định dừng sau khi huấn luyện đủ 100 cây	Thử hết mọi tổ hợp giá trị **GridSearch**	(1) n_estimators (2) max_depth (3) min_samples_split (4) min_samples_leaf (5) max_features (6) boostrap (7) random_states (8) n_jobs	(1) Số cây (2) Độ sâu tối đa mỗi cây (3) Số mẫu tối thiểu để mỗi nút được chia tách (4) Số mẫu tối thiểu trong mỗi nút (5) Số lượng đặc trưng mỗi khi tách nút (6) Quyết định xem có chọn dữ liệu cho từng cây để huấn luyện không (7) Tái tạo kết quả (8) Số lượng luồng xử lý	MAE: 0.706 MSE: 0.945 R2_SCORE: 0.971 MAPE: 10.141%	Mô hình cây ngẫu nhiên (ensemble của nhiều cây).
4	XGBOOST	Sau khi huấn luyện 500 vòng	Đặt thủ công	(1) objective (2) n_estimators (3) max_depth (4) learning_rate	(1) Hàm mất mát (2) Số lượng cây (3) Độ sâu tối đa mỗi cây (4) Tốc độ học	MAE: 0.734 MSE: 1.024 R2_SCORE: 0.968 MAPE: 10.357%	Boosting tree-based model
5	LSTM	Dừng sau 20 **epochs**	Tinh chỉnh kích thước lô huấn luyện, số vòng lặp huấn luyện. Sử dụng tối ưu Adam	(1) window_size (2) LSTM units (3) Dropout rate (4) Dense units (5) activation (6) optimazer (7) batch_size (8) learning_rate (9) epochs	(1) Độ dài chuỗi đầu vào (2) Số lượng đơn vị ẩn trong kiến trúc mô hình (3) Tỷ lệ dropout nhằm tránh overfit (4) Số neuron trong lớp fully connected (5) Hàm kích hoạt (6) Phương pháp tối ưu (7) Kích thước lô huấn luyện (8) Tốc độ học (9) Số vòng huấn luyện	MAE: 1.4609 MSE: 3.072 R2_SCORE: 0.9021 MAPE: 14.73%	Dạng RNN, chuyên dùng cho chuỗi thời gian
6	SVR	Không đặt giới hạn. Chạy đến khi mô hình được tối ưu	Gán thủ công giá trị	(1) kernel (2) C (3) epsilon (4) max_iter	(1) Dùng kernel Gaussian để phù hợp với dữ liệu không tuyến tính (2) Tham số điều chuẩn (3) Xác định vùng sai số xem có lỗi không (4) Giới hạn số vòng lặp (Trường hợp này là không giới hạn)	MAE: 0.90 MSE: 1.81 R2_SCORE: 0.94 MAPE: 11.29%	Hồi quy dựa trên SVM, có thể phi tuyến với kernel.
7	LightGBM	Dừng nếu sau 100 vòng, sai số không giảm	Thực hiện thủ công	(1) objective = 'mae' (2) metric = 'mae' (3) n_estimators (4) num_leaves (5) learning_rate (6) n_jobs (7) seed	(1) Tối ưu MAE (2) Dùng MAE đánh giá trong quá trình huấn luyện (3) Tối đa 1000 cây được xây dựng (4) Số lượng lá tối đa trong mỗi cây, ảnh hưởng đến độ phức tạp mô hình (5) Tốc độ học (6) Sử dụng toàn bộ lõi CPU (7) Đặt hạt giống ngẫu nhiên để tái lập kết quả	MAE: 1.55 MSE: 3.87 R2_SCORE: 0.89	Gốc từ GBDT, nhưng tối ưu tốc độ và bộ nhớ
8	Bayesian Ridge Regression	Thuật toán sẽ dừng khi sự thay đổi giữa các vòng lặp nhỏ hơn 0.001 hoặc đạt tối đa 300 vòng lặp.	Thực hiện thủ công	(1) alpha_1 (2) alpha_2 (3) lambda_1 (4) lambda_2 (5) n_iter (6) tol (7) fit_intercept (8) nomarlized (9) compute_score	(1) Tham số prior cho sự chính xác của trọng số (2) Tham số prior cho phân phối gamma của alpha (3) Tham số proor cho độ chính xác nhiễu (4) Tham số prior cho phân phối gamma của lambda (5) Số vòng lặp tối đa của EM (6) Sai số hội tụ của EM (7) Xem có nên học hệ số chênh lệch không (8) Chuẩn hóa dữ liệu đầu vào (9) Trả về Log - Likelihood mỗi vòng lặp nếu muốn theo dõi	MAE: 0.69 MSE: 0.84 R2_SCORE: 0.97 MAPE: 10.78%	Hồi quy tuyến tính có phân phối xác suất (Bayesian).

7.3 Kết luận chung

Các công việc đã được thực hiện đầy đủ theo checklist đã yêu cầu.

Các mô hình nhìn chung có xu hướng dự báo tốt (Hầu hết lý giải được trên 85% mô hình cùng các chỉ số như đã thống kê).

Giao diện được đóng gói có thể sử dụng ổn, phù hợp và chuẩn xác các thông số mô hình đã code. Tuy nhiên do chỉ đang thử nghiệm và chưa đủ kinh phí nên chưa thể năng cấp chuyên nghiệp và đầy đủ 8 mô hình.

Qua chủ đề rất thú vị và thực tế này, em đã thu được rất nhiều trải nghiệm đáng giá, từ kỹ năng xử lý, giải quyết vấn đề, biết chia nhỏ vấn đề để giải quyết từng bước. Quan trọng hơn, em đã được làm quen với quy trình xử lý một bài toán dự báo hoàn chỉnh.

Cá nhân em xin cảm ơn thầy Trần Ngọc Thăng đã cho em cơ hội thực hiện bài toán này!

Tài liệu tham khảo

https://en.wikipedia.org/wiki/LightGBM?utm_source=chatgpt.com
https://dataaspirant.com/lightgbm-algorithm/
https://www.mdpi.com/2075-4418/13/5/842/review_report
https://ieeexplore.ieee.org/document/10594630

Link giao diện dự báo thời tiết

http://localhost:8501/