



**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Nguyễn Tiến Khôi  
Tống Duy Tân  
Nguyễn Quốc Tuấn**

**BÁO CÁO HỌC MÁY DỰ ĐOÁN NHIỆT ĐỘ THỜI  
TIẾT CAO NHẤT TRONG NGÀY**

**BÁO CÁO DỰ ÁN CUỐI KỲ 2 NĂM HỌC 2023**

**Ngành: Trí tuệ nhân tạo**

**HÀ NỘI - 2023**

## TÓM TẮT

Dự báo nhiệt độ từ trước khi có công nghệ hiện đại đã được nghiên cứu rất nhiều, không chỉ bởi các nhà khoa học mà còn bởi những người nông dân. Trong dân gian đã có nhiều quy luật mà ông cha khái quát lên để dự báo thời tiết, nhiệt độ ngày mai. Nhận thấy rằng công cụ này rất hữu ích và đóng góp rất lớn cho đời sống nhân dân nên nhóm chúng tôi được truyền cảm hứng. Từ đó, chúng tôi thấy có sự tương đồng giữa sự đúc rút kinh nghiệm của cha ông ngàn năm với khả năng học được từ dữ liệu của học máy. Cùng thời gian đang trong quá trình học Học Máy của cô Diệp, chúng tôi quyết định làm một báo cáo về việc so sánh các mô hình dự đoán nhiệt độ cao nhất trong ngày tiếp theo, nhận định xem các mô hình học chúng tôi làm có điểm gì nổi bật và tối ưu hơn không. Đồng thời cũng rèn luyện kỹ năng phân tích dữ liệu, thiết kế bài toán học máy hợp lý, khả năng diễn giải được kết quả và ưu nhược điểm của mô hình trong bài toán thực tế, .... Bài toán học máy này bỏ qua rất nhiều quy luật về hiện tượng thời tiết nên chúng tôi hi vọng vào một kết quả tương đối khi dự đoán kết quả nhiệt độ cao nhất ngày kế tiếp.

Chính vì vậy, trong đề tài này, chúng em đã thực hiện quá trình cào dữ liệu từ trang <https://www.visualcrossing.com/> để lấy dữ liệu, nghiên cứu rồi phân tích thông tin thu được để làm 1 bài báo cáo đề tài này. Do lần đầu được làm báo cáo nên còn nhiều thiếu sót, chúng em mong nhận được sự góp ý, phê bình của thầy để báo cáo của chúng em được hoàn thiện hơn.

Từ khóa: Học Máy, dự báo thời tiết

# Phần 1: Thu thập dữ liệu từ Visual Crossing về và xử lý dữ liệu

(Lưu vào file hanoiweather.csv)

## I, Thu thập dữ liệu

- Trang <https://www.visualcrossing.com/> là trang web chuyên cung cấp về dữ liệu thời tiết và API uy tín và có bộ dataset đầy đủ nên chúng em chọn web này để thu thập dữ liệu
- Địa điểm thu thập dữ liệu là Hà Nội, thời gian sẽ thu thập từ năm 1990 đến hiện tại và lấy theo ngày
- Mỗi ngày mỗi tài khoản sẽ được crawl dữ liệu khoảng tầm 1 năm nên chúng em 3 người đã thu thập dữ liệu trong khoảng 10 ngày.
- Phần chia công việc thu thập dữ liệu
  - Nguyễn Quốc Tuấn: 1/1/1990 - 31/12/1999
  - Tống Duy Tân: 1/1/2000 - 31/12/2009
  - Nguyễn Tiến Khôi: 1/1/2010 – đến nay

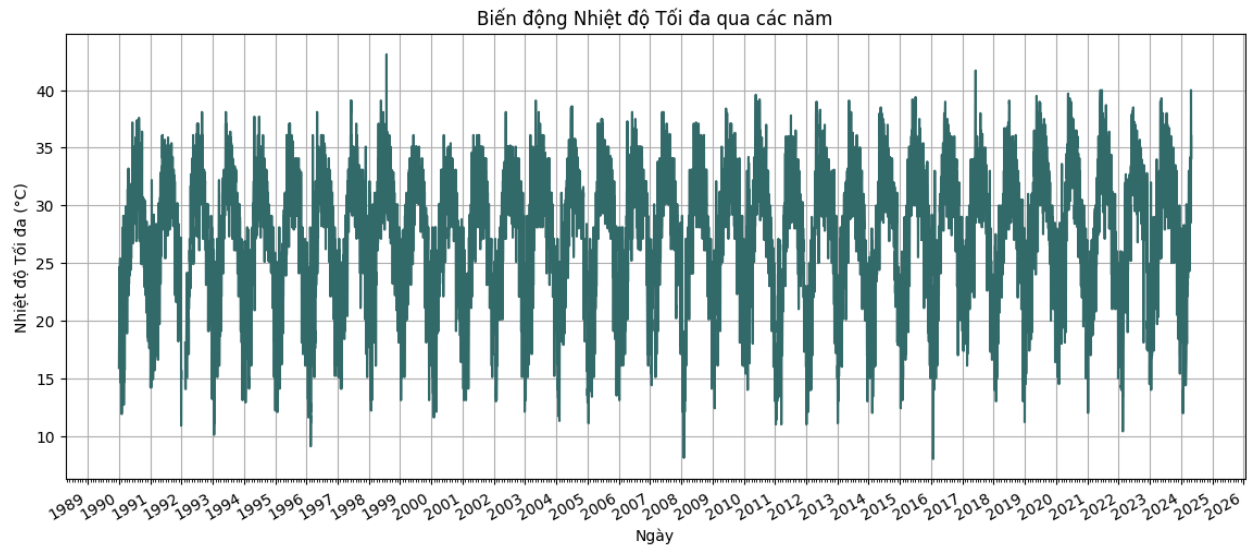
datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	dew	humidity	precip	precipprob	precipcover	precipitype	snow	snowdepth	windgust	windspeed	winddir	pr
2024-04-30	37	27	31.1	50.2	31.2	40.1	26.8	78.8	0	0	0		0	0	35.3	18.4	84.6	99
2024-05-01	28	24	26	31.5	24	27.1	22.8	83	6.5	100	62.5	rain	0	0	36.7	18.4	41.9	10
2024-05-02	26.2	22	24.1	26.2	22	24.1	22.5	91.2	32.82	100	8.33	rain	0	0	24.5	16.6	61.7	10
2024-05-03	27.7	23	25.1	30.9	23	25.6	23.6	91.7	34.457	100	12.5	rain	0	0	39.6	22.3	103	10
2024-05-04	33	25	28.2	42.1	25	32.1	25.1	84.2	0.2	100	8.33	rain	0	0	29.5	22.3	117.8	10
2024-05-05	28.7	24	26.5	31.4	24	27.8	22.2	77.5	2.6	100	37.5	rain	0	0	25.6	16.6	53.5	10
2024-05-06	28.8	24.2	26	32.2	24.2	26.9	23.5	86.6	2.241	100	4.17	rain	0	0	24.5	16.6	52.8	10

Bộ dữ liệu khá nhiều cột và đầy đủ (37 cột)

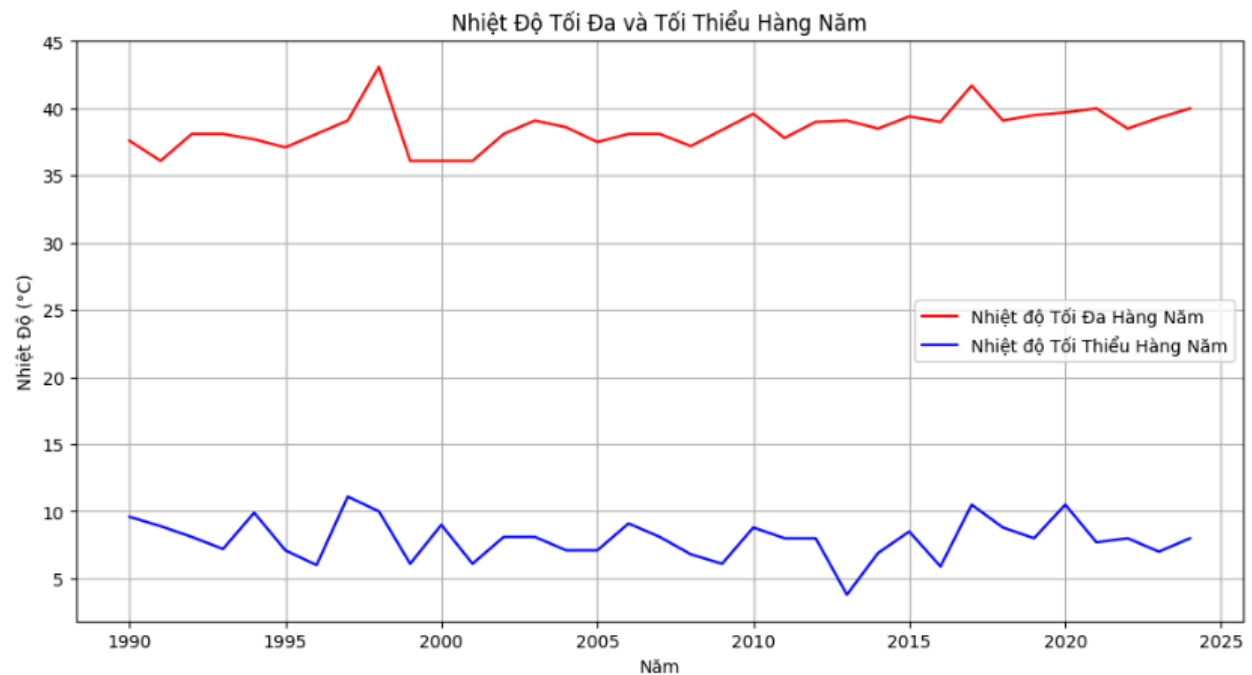
## II, Xử lý dữ liệu

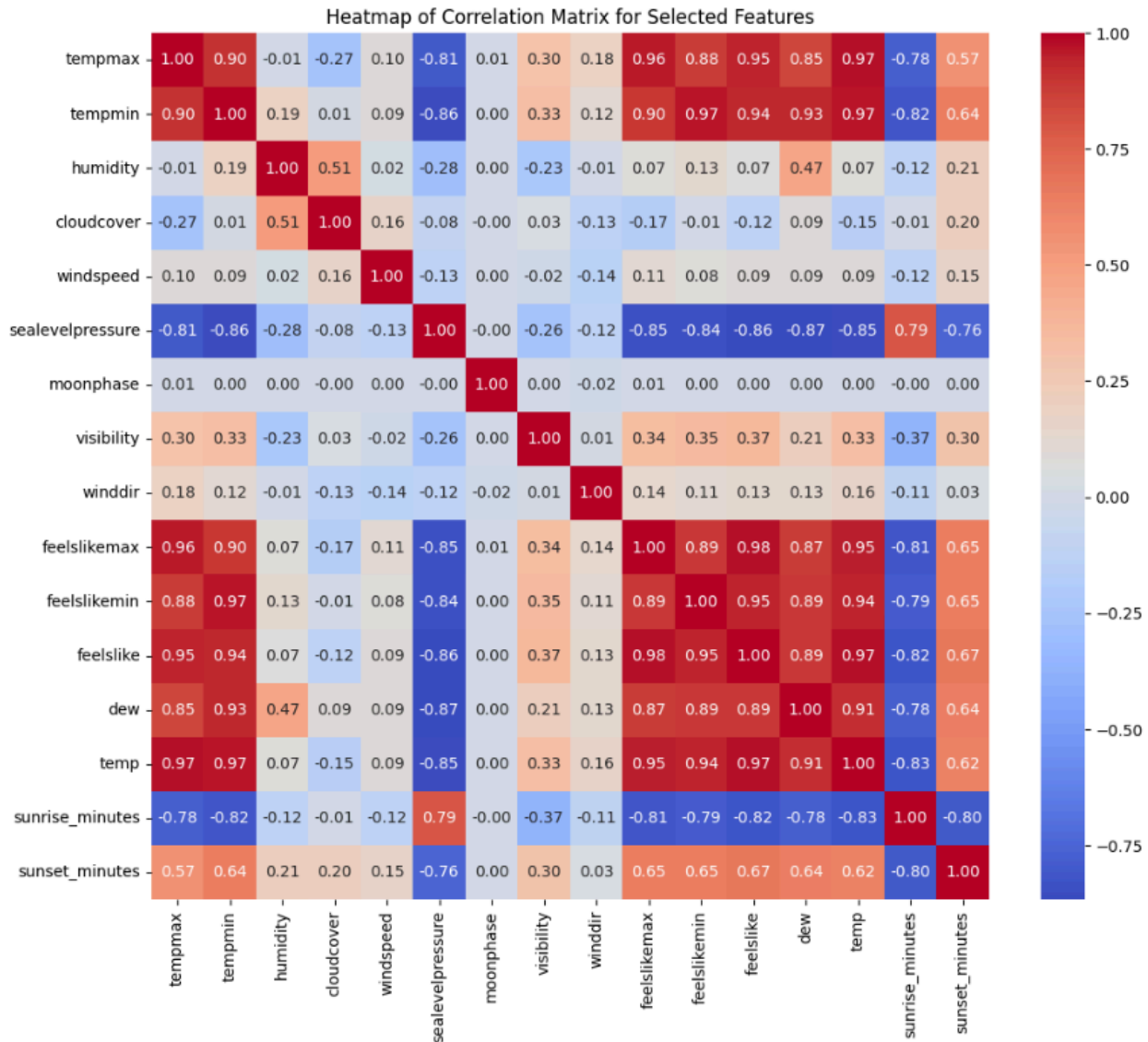
- Để tận dụng hết các thông tin của cột, em biến đổi thời gian sunrise, sunset trong ngày từ "%Y-%m-%dT%H: %M: %S" sang dạng phút.
- Thêm 1 cột tempmax\_next để dự đoán nhiệt độ cao nhất trong ngày (mục tiêu)
- Vì weather['conditions'] trong dữ liệu được miêu tả như 'Overcast', 'Partially cloudy' hay 'Rain, Partially cloudy' nên sẽ tạo luôn các cột tương ứng với điều kiện thời tiết như vậy và đánh giá true hay false cho tất cả các ngày như là 'cond\_Overcast', 'cond\_Partially cloudy', hay 'cond\_Rain, Partially cloudy']
- Lấp các giá trị khuyết trong bộ dữ liệu (57/12538) với tỷ lệ chỉ khoảng 0.475% không đáng kể đến chất lượng tổng thể của bộ dữ liệu. Phương pháp "forward fill" lấp đầy các giá trị NaN bằng cách sử dụng giá trị cuối cùng không phải là NaN trước chúng.

## Phần 2: Phân tích dữ liệu và đánh giá



- Nhiệt độ tối đa trong bộ dữ liệu là: 43.1 °C
- Nhiệt độ tối thiểu trong bộ dữ liệu là: 3.8 °C





- Phân tích: Từ biểu đồ này, ta có thể phân tích sâu hơn về mối quan hệ giữa các biến, chẳng hạn như khám phá nguyên nhân của các tương quan cao hoặc thấp và cách chúng ảnh hưởng đến các yếu tố thời tiết khác: **tempmax**, **tempmin**, **temp**, **feelslikemax**, **feelslikemin**, **feelslike**, **sunrise\_minutes** và **sealevelpressure** đều có tương quan rất cao với nhau, điều này có ý nghĩa vì chúng đều liên quan trực tiếp đến nhiệt độ. **temp** (nhiệt độ trung bình) cũng có tương quan rất cao với **tempmax**, **tempmin**, và các biến "feels like", thể hiện rõ mối liên hệ giữa nhiệt độ trung bình và các chỉ số nhiệt độ khác.

- Chọn đặc trưng cho mô hình học máy:
  - Tương quan giữa các đặc trưng có thể giúp chúng ta chọn lọc các đặc trưng quan trọng hoặc loại bỏ các đặc trưng thừa cho các mô hình học máy, nhằm cải thiện hiệu quả và độ chính xác của mô hình

## Phần 3: Tạo các bộ dữ liệu dành cho training, validating và testing.

Thông nhất chia bộ dữ liệu như sau:

- Bộ dữ liệu test tính từ ngày 2024-05-01 - 2024-05-16
- Bộ dữ liệu còn lại dùng để huấn luyện và đánh giá:
  - chia tập đánh giá val từ 2018-12-31 - 2024-04-30

Kết quả dùng để so sánh các mô hình với nhau là dựa trên đánh giá tập test và tham khảo kết quả từ tập đánh giá.

## Phần 4: Thực nghiệm và tối ưu mô hình

### I, Mô tả bài toán học máy

Bài toán học máy của nhóm là dự báo nhiệt độ cao nhất của ngày kế tiếp dựa trên dữ liệu của các ngày trước đó, là một bài toán thuộc lĩnh vực hồi quy.

Để tiện cho việc đánh giá, nhóm đã gắn nhãn cho mỗi ngày là nhiệt độ cao nhất của ngày kế tiếp.

Các mô hình xây dựng sẽ dựa trên:

- fb prophet
- random forest
- ridge regression
- support vector machine

Các phương pháp đo lường và đánh giá bao gồm:


- Mean-absolute-error: đo trung bình chênh lệch tuyệt đối giữa nhiệt độ dự đoán và thực tế là cách đơn giản nhất để đánh giá mô hình nhưng nó vẫn không làm nổi bật được ưu điểm của mô hình so với cách tiếp cận ngây thơ.
- Mean-squared-error: Khoảng chênh lệch nhiệt bình phương giữa nhiệt độ dự đoán và nhiệt độ thật sẽ làm nổi bật được những điểm có chênh lệch cao tới bất thường trong tập dự đoán, giúp ta đưa ra cái cách đánh giá đúng đắn về mô hình hơn.
- Mean-absolute-percentage-error: là trung bình của các tỷ lệ phần trăm tuyệt đối của chênh lệch giữa giá trị dự đoán và giá trị thực tế.

### Xác định baseline của bài toán học máy:

Chưa cần nhìn bộ dữ liệu, theo thực tế khách quan, Hà Nội là vùng đồng bằng nằm trung tâm miền bắc, nên nơi này không có thiên tai, từ đó ta luôn suy đoán rằng thời tiết Hà Nội hai ngày liên tiếp thường không thể thay đổi quá nhiều được.

Vì vậy, cho rằng nhiệt độ ngày hôm sau chính bằng nhiệt độ ngày hôm trước, ta có được baseline của bài toán với các số đo lỗi như sau:

mse,mae naive approach on val


```
✓ 0s  se = ((val['y'] - val['tempmax']) ** 2)  
print(se.mean())
```

 6.666623376623377

```
✓ 0s [52] ae = abs((val['y'] - val['tempmax']))  
print(ae.mean())
```


 1.8844155844155843

mse,mae naive approach on test

```
✓ 0s  se = ((test['y'] - test['tempmax']) ** 2)  
print(se.mean())
```

 5.047499999999999

```
✓ 0s [54] ae = abs((test['y'] - test['tempmax']))  
print(ae.mean())
```

 1.6999999999999997

## II, Thực nghiệm

Áp dụng fb prophet xây dựng mô hình dự đoán:

### Giới thiệu:

Facebook Prophet là một công cụ dự báo chuỗi thời gian mạnh mẽ, được thiết kế bởi Facebook. Đây là **mô hình additive**, nghĩa là nó phân tách các thành phần khác nhau của chuỗi thời gian như xu hướng, mùa vụ và ngày lễ rồi dự báo chúng một cách riêng biệt.

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

Với:

- $g(t)$  là xu hướng trong khoảng thời gian của dữ liệu
- $s(t)$  là mùa vụ hay là chu kỳ thay đổi ngắn
- $h(t)$  là ngày lễ, là những ngày đặc biệt nhưng lặp lại theo quy luật
- $e(t)$  là những thay đổi không có điều kiện xảy ra bất thường



- $y(t)$  là kết quả dự báo

Đặc điểm nổi bật của FB Prophet là khả năng xử lý các chuỗi thời gian có xu hướng phi tuyến tính và có tính mùa vụ đa dạng. Công cụ này đặc biệt hữu ích khi làm việc với các dữ liệu có khoảng thời gian thiếu hụt hoặc có sự thay đổi theo mùa phức tạp. Prophet dễ sử dụng và yêu cầu ít kiến thức chuyên môn về chuỗi thời gian, giúp các nhà phân tích và nhà khoa học dữ liệu có thể tạo ra các mô hình dự báo chính xác mà không cần quá nhiều nỗ lực. Ngoài ra, nó còn cung cấp các công cụ trực quan hóa giúp dễ dàng kiểm tra và điều chỉnh các thành phần của mô hình.

### Lý do chọn :

mô hình prophet được áp dụng vào bài toán dự báo nhiệt độ là hoàn toàn hợp lý với các tính chất của nhiệt độ, đặc biệt là nhiệt độ miền bắc có tính phân biệt rõ rệt, không chỉ vậy, còn có các hiệu ứng dòng biển nóng, lạnh tác động đến nhiệt độ làm tính mùa vụ của nhiệt độ càng thêm phức tạp. Đặc biệt, mô hình có khả năng tự điều chỉnh dựa trên tính cộng thành phần,, giúp cho việc thiết lập các thành phần biến đầu vào trở nên dễ dàng hơn. Cách tiếp cận bằng prophet sẽ đưa đến không chỉ là dự đoán nhiệt độ cao nhất mà nó cho ta một vùng mà nhiệt độ cao nhất ngày tới có khả năng rơi vào

Áp dụng ridge regression xây dựng mô hình dự đoán:

### Giới thiệu Ridge Regression:

Ridge Regression là một kỹ thuật học máy thuộc nhóm hồi quy tuyến tính, được sử dụng để phân tích và dự báo các chuỗi thời gian và dữ liệu với các đặc tính đa chiều. Mô hình Ridge Regression được thiết kế để giải quyết vấn đề quá khớp (overfitting) bằng cách thêm các thuật toán chuẩn hóa, giúp giảm thiểu độ lớn của các hệ số hồi quy. Điều này giúp mô hình trở nên ổn định hơn khi xử lý các tập dữ liệu có nhiều biến đầu vào, đồng thời cải thiện khả năng dự đoán trên dữ liệu mới. Ridge Regression rất hữu ích trong các trường hợp có đa cộng tuyến (multicollinearity) giữa các biến độc lập, bởi nó giúp làm giảm độ nhạy cảm của mô hình đối với những biến này.

### Lý do chọn Ridge Regression

Việc chọn mô hình Ridge Regression để dự báo nhiệt độ là một lựa chọn hợp lý bởi tính chất của dữ liệu nhiệt độ, đặc biệt là ở miền Bắc với sự thay đổi khá rõ rệt theo mùa. Ridge Regression giúp xử lý tốt các yếu tố đầu vào phức tạp và đa chiều, làm cho mô hình trở nên ổn định và ít bị nhiễu bởi các biến không quan trọng. Mô hình Ridge Regression cũng có khả năng điều chỉnh và tối ưu hóa các hệ số hồi quy để phù hợp với dữ liệu thực tế, giúp dự báo chính xác hơn.

Áp dụng Random Forest xây dựng mô hình dự đoán:

### Giới Thiệu :

Random Forest là một thuật toán học máy mạnh mẽ và linh hoạt được sử dụng rộng rãi cho cả các bài toán phân loại và hồi quy. Nó được xây dựng dựa trên tập hợp các cây quyết định (decision trees), nơi mỗi cây quyết định trong rừng ngẫu nhiên được xây dựng từ một mẫu ngẫu nhiên của dữ liệu

huấn luyện và một tập con ngẫu nhiên của các đặc trưng. Sau đó, các **dự đoán từ từng cây quyết định** được **kết hợp lại** để **đưa ra dự đoán cuối cùng**. Điều này giúp giảm thiểu hiện tượng quá khớp (overfitting) và cải thiện độ chính xác của mô hình.

### Lý Do Chọn :

Random Forest được áp dụng vào bài toán dự báo nhiệt độ là hoàn toàn hợp lý với các tính chất của nhiệt độ, đặc biệt là khi mô hình cần xử lý nhiều đặc trưng (features) khác nhau như nhiệt độ tối thiểu, độ ẩm, tốc độ gió, áp suất mực nước biển và các điều kiện thời tiết khác. Random Forest có khả năng xử lý dữ liệu phi tuyến tính và tương tác phức tạp giữa các đặc trưng, giúp nó trở nên linh hoạt hơn so với các mô hình hồi quy tuyến tính truyền thống. Mô hình này cũng có khả năng xử lý tốt với dữ liệu thiếu và các ngoại lệ, điều này rất quan trọng trong việc dự báo nhiệt độ.

Áp dụng SVM xây dựng mô hình dự đoán:

### Giới thiệu:

Support Vector Machines (SVM) là một thuật toán học máy có khả năng xử lý cả dữ liệu tuyến tính và phi tuyến. Đặc điểm nổi bật của SVM là khả năng tạo ra các ranh giới phân chia tối ưu giữa các nhóm dữ liệu khác nhau trong không gian đa chiều.

Đối với dữ liệu phi tuyến tính, SVM sử dụng một kỹ thuật gọi là "kernel trick" để ánh xạ các điểm dữ liệu từ không gian ban đầu sang một không gian đặc biệt (cao chiều) mà trong đó các lớp có thể được phân tách tốt hơn. Các hạt nhân (kernels) khác nhau như đa thức (polynomial), Radial Basis Function (RBF), hay sigmoid cho phép SVM hoạt động hiệu quả trên các dữ liệu có độ phức tạp cao, gồm những dữ liệu mà không thể phân tách tuyến tính.

### Lý do chọn :

Dữ liệu về nhiệt độ là một dãy dữ liệu phi tuyến tính với các đặc trưng quan hệ phức tạp nên ta chọn svm có thể là một lựa chọn tốt.

## III, Tối Ưu

### Tối Ưu cho mô hình Prophet:

Mô hình prophet được tinh chỉnh cho nhiệm vụ dự báo nhiệt độ cao nhất trong ngày như sau:

- Mô hình đã được loại bỏ tính thời vụ theo ngày và tuần, giữ lại tính thời vụ theo năm.
- Thêm tính thời vụ dài hơn cho mô hình để biểu diễn chu kỳ nóng-lạnh của dòng biển mà luôn có tác động mạnh đối với khí hậu của Đông Nam Á.
- Tiến hành thêm các biến đầu vào để cải thiện độ chính xác của mô hình

Sử dụng grid-search để tìm ra các siêu tham số tối ưu cho mô hình bao gồm:

- **n\_changepoints**: chọn ra tổng số điểm thay đổi trong dữ liệu
- tham số **changepoint\_prior\_scale** kiểm soát độ nhạy của mô hình đối với các điểm thay đổi (changepoints) trong xu hướng của dữ liệu.

- **period**: là siêu tham số được thêm vào để kiểm soát thời vụ dài hơn năm của thời tiết dựa trên biến đổi khí hậu
- **fourier\_order**: kiểm soát sự biến đổi của tính thời vụ period, fourier\_order càng cao biến động càng mạnh:

$$S(t) = \sum_{n=1}^N [a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P})]$$

N: là fourier order cần được điều chỉnh

P: là period

- tham số **seasonality\_prior\_scale** kiểm soát độ nhạy của mô hình đối với các biến động mùa vụ trong dữ liệu.

```
from sklearn.model_selection import ParameterGrid
params_grid = {
    'changepoint_prior_scale': [0.002, 0.004, 0.008],
    'n_changepoints': [10, 25, 50],
    'seasonality_prior_scale': [0.05, 0.1, 0.2],
    'period': [720, 1080, 1440, 1800],
    'fourier_order': [1, 2, 3, 4, 5],
}
grid = ParameterGrid(params_grid)
cnt = 0
for p in grid:
    cnt = cnt + 1

print('Total Possible Models', cnt)
```

Total Possible Models 540

Sau khi khởi tạo 540 bộ siêu tham số, ta chọn được hyperparameter như sau:

- changepoint\_prior\_scale: 0.002,
- fourier\_order: 1,
- n\_changepoints: 50,
- period: 1800,
- seasonality\_prior\_scale: 0.05

### Tối ưu mô hình Ridge Regression:

Sử dụng Pipeline từ scikit-learn để gói gọn quy trình chuẩn hóa và huấn luyện mô hình vào một đối tượng duy nhất, giúp mã ngắn gọn và dễ bảo trì hơn. Pipeline gồm các bước: chuẩn hóa dữ liệu với StandardScaler, chuẩn hóa L2 với Normalizer, và mô hình Ridge Regression.

Chuẩn hóa dữ liệu với StandardScaler: StandardScaler chuẩn hóa các đặc trưng (features) sao cho chúng có phân phối với giá trị trung bình (mean) bằng 0 và độ lệch chuẩn (standard deviation) bằng 1. Nó sẽ giúp thuật

toán học máy như Ridge Regression hoạt động tốt hơn khi các đặc trưng có cùng đơn vị và phạm vi giá trị. Điều này giúp tăng tốc độ hội tụ và cải thiện hiệu suất mô hình.

Chuẩn hóa L2 với Normalizer: Normalizer chuẩn hóa các đặc trưng của mỗi mẫu (sample) sao cho tổng bình phương các giá trị bằng 1 (L2 norm). Việc chuẩn hóa L2 giúp giảm ảnh hưởng của độ lớn của các giá trị đặc trưng, đảm bảo rằng mọi đặc trưng đều đóng góp đều nhau vào mô hình.

### Tối ưu mô hình Random Forest:

Sử dụng GridSearchCV để tìm ra bộ **tham số tối ưu** cho mô hình Random Forest. Grid Search giúp đảm bảo rằng mô hình không chỉ phù hợp với tập huấn luyện mà còn có khả năng tổng quát hóa tốt trên dữ liệu mới.

```
param_grid = {  
    'n_estimators': [100, 200, 300, 400, 500],  
    'max_features': ['auto', 'sqrt', 'log2'],  
    'max_depth': [None, 10, 20, 30],  
    'min_samples_split': [2, 5, 10],  
    'min_samples_leaf': [1, 2, 4]  
}
```

Đây là các yếu tố đặc trưng của Random Forest:

**n\_estimators**: Số lượng cây quyết định trong rừng ngẫu nhiên.

**max\_features**: Số lượng đặc trưng để xem xét khi tìm kiếm phân chia tốt nhất.

**max\_depth**: Độ sâu tối đa của cây.

**min\_samples\_split**: Số lượng mẫu tối thiểu cần thiết để chia nút.

**min\_samples\_leaf**: Số lượng mẫu tối thiểu tại nút lá.

```
from sklearn.model_selection import GridSearchCV  
  
param_grid = {  
    'n_estimators': [100, 200, 300, 400, 500],  
    'max_features': ['auto', 'sqrt', 'log2'],  
    'max_depth': [None, 10, 20, 30],  
    'min_samples_split': [2, 5, 10],  
    'min_samples_leaf': [1, 2, 4]  
}  
  
rf = RandomForestRegressor(random_state=42)  
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3, n_jobs=-1, verbose=2, scoring='neg_mean_squared_error')  
grid_search.fit(X_train, y_train)
```

Sau khi khởi tạo 1620 bộ tham số, ta chọn được hyperparameter như sau:

- max\_depth: 20

- max\_features: 'sqrt',
- min\_samples\_leaf: 1,
- min\_samples\_split: 5,
- n\_estimators: 500

#### Tối ưu mô hình SVM:

Cách 1: Chọn tối ưu svm theo cách tự xây một kernel mới lấy ý tưởng từ khoảng cách mahalanobis.

Đặc biệt của Mahalanobis kernel là nó có khả năng xử lý dữ liệu mà các đặc trưng (features) không chỉ có tính chất phân phối không đồng nhất (không cùng phương sai và hiệp phương sai) mà còn có mối quan hệ tuyến tính phức tạp.

Khoảng cách Mahalanobis của vector  $x = (x_1, x_2, x_3, \dots, x_N)^T$  so với một nhóm có trung bình là  $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_N)^T$  và ma trận hiệp phương sai  $S$  được định nghĩa:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}$$

Cách 2: dùng rbf kernel

rbf kernel là lựa chọn phù hợp nhất đối với dữ liệu nhiệt độ sau mahalanobis

## Phần 5: Đánh giá mô hình

### Đánh giá mô hình sử dụng fb prophet:

Điểm mạnh:

- Mô hình có thể nắm bắt được tính thời vụ và xu hướng của dữ liệu và tự điều chỉnh dựa trên các biến đầu vào. Các siêu tham số đã được tinh chỉnh để phù hợp với nhiệm vụ dự báo nhiệt độ.
- Tốc độ tính toán của mô hình rất nhanh, thời gian chạy tìm kiếm tham số tối ưu 540 bộ ước tính mất khoảng 1.5 h (chạy trên google colab)
- Về sai số của mô hình đã vượt qua được baseline cả trên tập đánh giá và tập test.

Kết quả sai số trên tập train

```
mse(predictions)
mae(predictions)
mape(predictions)
```

```
MSE: 4.9259334351398625
MAE: 1.731258214795954
MAPE: 6.62%
```

Kết quả sai số khi cross validation

```
mse(cv, actual_label="y")
mae(cv, actual_label='y')
mape(cv, actual_label = 'y')
```

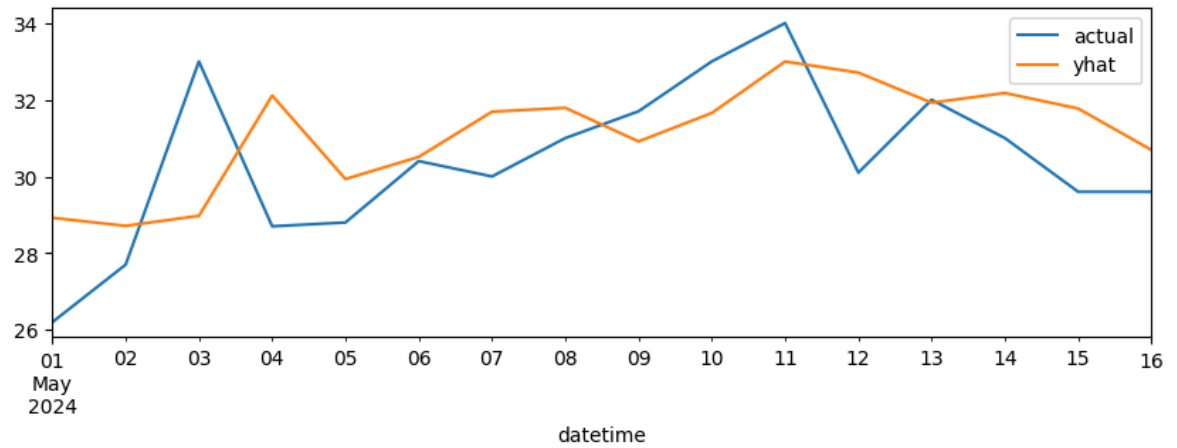
```
4.8935578100056
1.7228681633437897
MAPE: 6.74%
```

Kết quả sai số trên tập test

```
mse(forecast_test)
mae(forecast_test)
mape(forecast_test)
```

```
MSE: 3.6606572794880514
MAE: 1.571775702984708
MAPE: 5.24%
```

- Trong dự báo khoảng 16 ngày test , mô hình thể hiện được khả năng vượt trội khi dự báo chỉ lệch nhiều nhất ngày 2024-05-04 là khoảng 4°C và các ngày sau lệch khoảng ~ 1°C(Xét mse là 5.6 trên sai số se của ngày 2024-05-01 là 16 và mae là 1.83)



- Ngoài ra, khi sử dụng mô hình tạo bởi fb prophet, ta có được khoảng biến động của nhiệt độ dự đoán, đây là điều mà các mô hình khác không làm được:

	trend	yhat_lower	yhat_upper	yhat	actual
datetime					
2024-04-30	26.651611	31.379402	36.865708	34.106812	28.0
2024-05-01	26.651646	26.311599	31.891197	28.959551	26.2
2024-05-02	26.651681	25.928408	31.427492	28.716433	27.7
2024-05-03	26.651715	26.172210	31.736582	29.011944	33.0
2024-05-04	26.651750	29.296772	34.921204	32.122532	28.7
2024-05-05	26.651785	27.028958	32.713172	29.946533	28.8
2024-05-06	26.651819	27.747784	33.228393	30.494852	30.4
2024-05-07	26.651854	28.794711	34.485513	31.674982	30.0
2024-05-08	26.651889	28.937331	34.472807	31.772724	31.0
2024-05-09	26.651923	28.207162	33.789060	30.909025	31.7
2024-05-10	26.651958	28.716999	34.401169	31.690181	33.0
2024-05-11	26.651993	30.112369	35.688910	33.002592	34.0
2024-05-12	26.652027	29.936484	35.414360	32.723331	30.1
2024-05-13	26.652062	28.968456	34.828058	31.950452	32.0
2024-05-14	26.652097	29.187931	35.150726	32.207195	31.0
2024-05-15	26.652131	28.898848	34.694243	31.815038	29.6
2024-05-16	26.652166	27.746520	33.399431	30.615450	29.6

Điểm yếu:

- Tuy nhiên Prophet gặp khó khăn trong việc ước lượng và mô hình các thay đổi nhiệt độ nhanh chóng và đột ngột, chẳng hạn như các cơn gió mạnh, mưa bất ngờ, hoặc những biến đổi thời tiết không thường xuyên. Mặc dù khi dự báo cho Hà Nội là vị trí rất ít biến động nhưng vẫn không hạn chế được điều này. Ví dụ là ngày 2024-05-01.
- Prophet cho phép chia thời gian nhỏ đến từng giờ để có thể nắm bắt xu hướng và tính thời vụ vì vậy với dữ liệu lấy theo ngày, mô hình chưa phát huy được hết khả năng của nó.
- Mô hình được tinh chỉnh để phù hợp với nắm bắt xu hướng nhiệt độ lấy theo ngày nên nếu ta có đầy thời điểm dữ liệu hơn, quá trình tối ưu hoá sẽ mất rất nhiều thời gian do có quá nhiều siêu tham số có thể được thay đổi.



## Đánh giá mô hình sử dụng Ridge Regression:

Điểm mạnh:

- Mô hình có thể nắm bắt được tính thời vụ và xu hướng của dữ liệu và tự điều chỉnh dựa trên các biến đầu vào. Các siêu tham số đã được tinh chỉnh để phù hợp với nhiệm vụ dự báo nhiệt độ.
- Tốc độ tính toán của mô hình khá nhanh.
- Về sai số của mô hình đã vượt qua được baseline cả trên tập đánh giá và tập test.

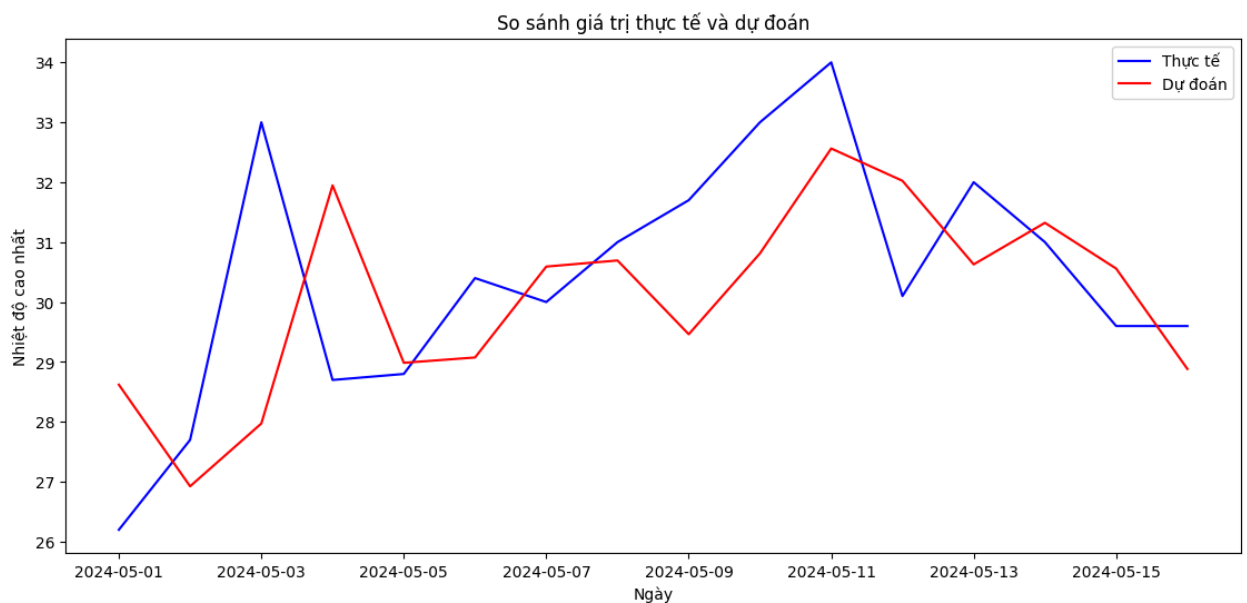
Kết quả sai số khi cross validation

```
MAE: 1.7882972375419406
MSE: 5.345551432466003
MAPE: 0.06847424798933173
```

Kết quả sai số trên tập test

```
MAE: 1.5645834661242861
MSE: 3.9675880436918862
MAPE: 0.051148912461271415
```

- Trong dự báo khoảng 16 ngày test, mô hình thể hiện được khả năng vượt trội khi dự báo chỉ lệch nhiều nhất ngày 2024-05-03 là khoảng  $5^{\circ}\text{C}$  và các ngày sau lệch khoảng  $\sim 1 - 3^{\circ}\text{C}$ .



Điểm yếu:

- Ridge Regression không loại bỏ hoàn toàn các đặc trưng không quan trọng, mà chỉ làm giảm trọng số của chúng. Tuy đã loại bỏ ngoại lệ và chuẩn hóa dữ liệu giúp giảm bớt vấn đề này, nhưng không thể loại bỏ hoàn toàn các đặc trưng không liên quan.

- Ridge Regression cũng gặp khó khăn trong việc ước lượng và mô hình các thay đổi nhiệt độ nhanh chóng và đột ngột. Ví dụ như ngày 30/04/2024 nhiệt độ lớn nhất ở Hà Nội là 36°C thì hôm sau ngày 01/05/2024 nhiệt độ lớn nhất ở Hà Nội là 28°C chênh lệch tới 8°C. Những sự thay đổi đột ngột và thất thường này làm giảm độ chính xác của mô hình
- Mô hình cần được tinh chỉnh để phù hợp với việc nắm bắt xu hướng thay đổi nhiệt độ lấy theo ngày nên quá trình chuẩn hóa, xử lý dữ liệu và tối ưu hoá sẽ mất rất nhiều thời gian do có quá nhiều siêu tham số có thể thay đổi.

## Đánh giá mô hình sử dụng Random Forest:

Điểm mạnh:

- Mô hình có thể nắm bắt được tính thời vụ và xu hướng của dữ liệu và tự điều chỉnh dựa trên các biến đầu vào. Các siêu tham số đã được tinh chỉnh để phù hợp với nhiệm vụ dự báo nhiệt độ.
- Tốc độ tính toán của mô hình ổn, thời gian chạy tìm kiếm tham số tối ưu 1620 bộ ước tính mất khoảng 1h (chạy trên google colab)
- Về sai số của mô hình đã vượt qua được baseline cả trên tập đánh giá và tập test.

Kết quả sai số trên tập validation

```
Validation MSE: 4.8394537828719795
Validation MAE: 1.7270017301038099
Validation MAPE: 6.552794428246428
```

Kiểm tra sai số trên tập test

```
Test MSE: 4.582825374999987
Test MAE: 1.648125000000001
Test MAPE: 5.474942388834034
```

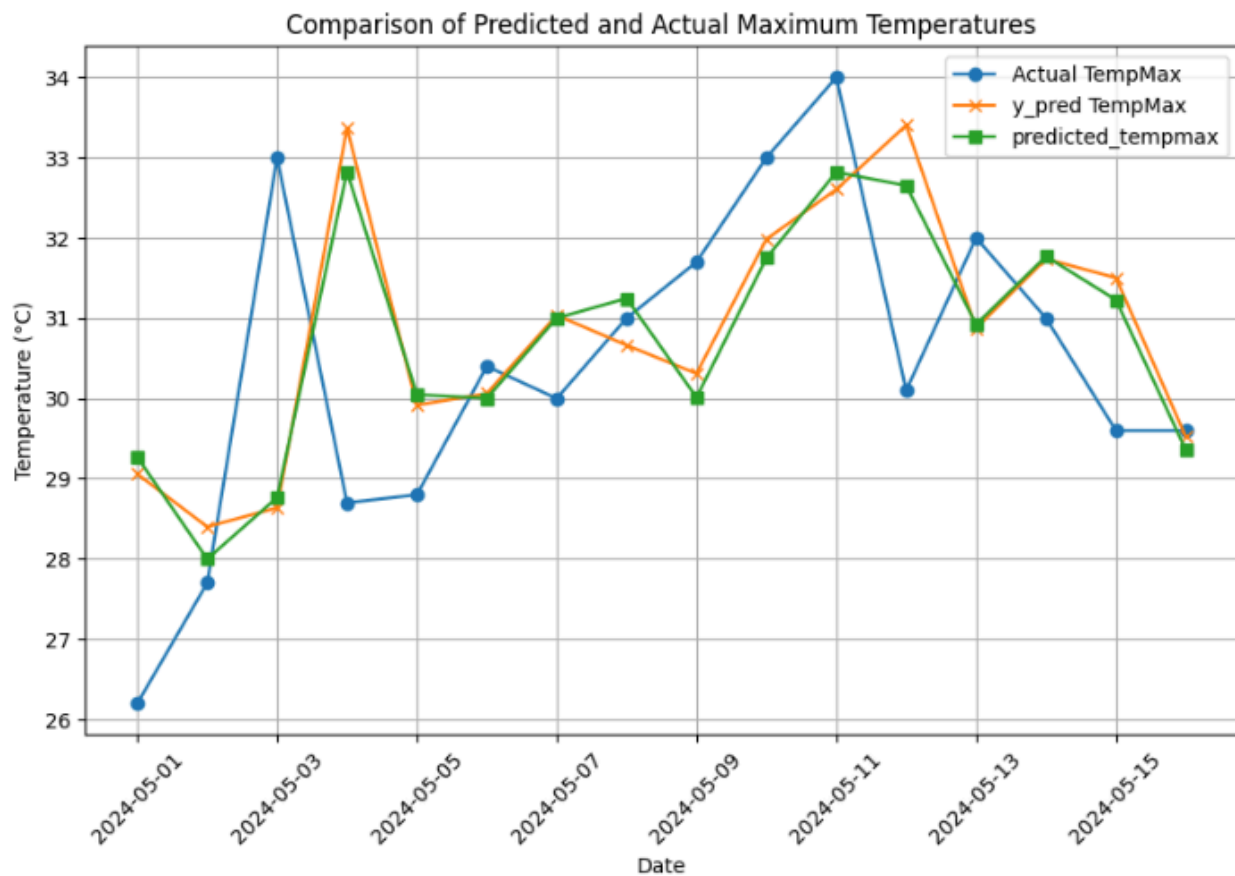
- Trong dự báo khoảng 16 ngày test, mô hình thể hiện được khả năng vượt trội khi dự báo, có những ngày kết quả dự đoán giống với tập dữ liệu thể nhưng có những ngày lệch 2 độ C (Xét mse là 4.58 trên sai số se của ngày 2024-05-04 là 16)

Điểm yếu:

- Random forest cũng gặp khó khăn trong việc ước lượng và mô hình các thay đổi nhiệt độ nhanh chóng và đột ngột, chẳng hạn như các cơn gió mạnh, mưa bất ngờ, hoặc những biến đổi thời tiết không thường xuyên. Mặc dù khi dự báo cho Hà Nội là vị trí rất ít biến động nhưng vẫn không hạn chế được điều này. Ví dụ là ngày 2024-05-01.
- Random forest đưa ra kết quả dự đoán dữ liệu không ổn định so với mục tiêu dự đoán nhiệt độ cao nhất trong ngày tiếp theo trong tập dữ liệu, lúc thì có thể đúng hoàn toàn so nhưng mà

có vài lúc lệch tận 2 độ C. Kết quả này xảy ra có thể do 1 phần ở dữ liệu, khi thời gian được thu thập theo ngày liên tục nhưng không phải theo giờ liên tục

- Mô hình được tinh chỉnh để phù hợp với nắm bắt xu hướng nhiệt độ lấy theo ngày nên nếu ta có dày thời điểm dữ liệu hơn, quá trình tối ưu hoá sẽ mất rất nhiều thời gian do có quá nhiều siêu tham số có thể được thay đổi.

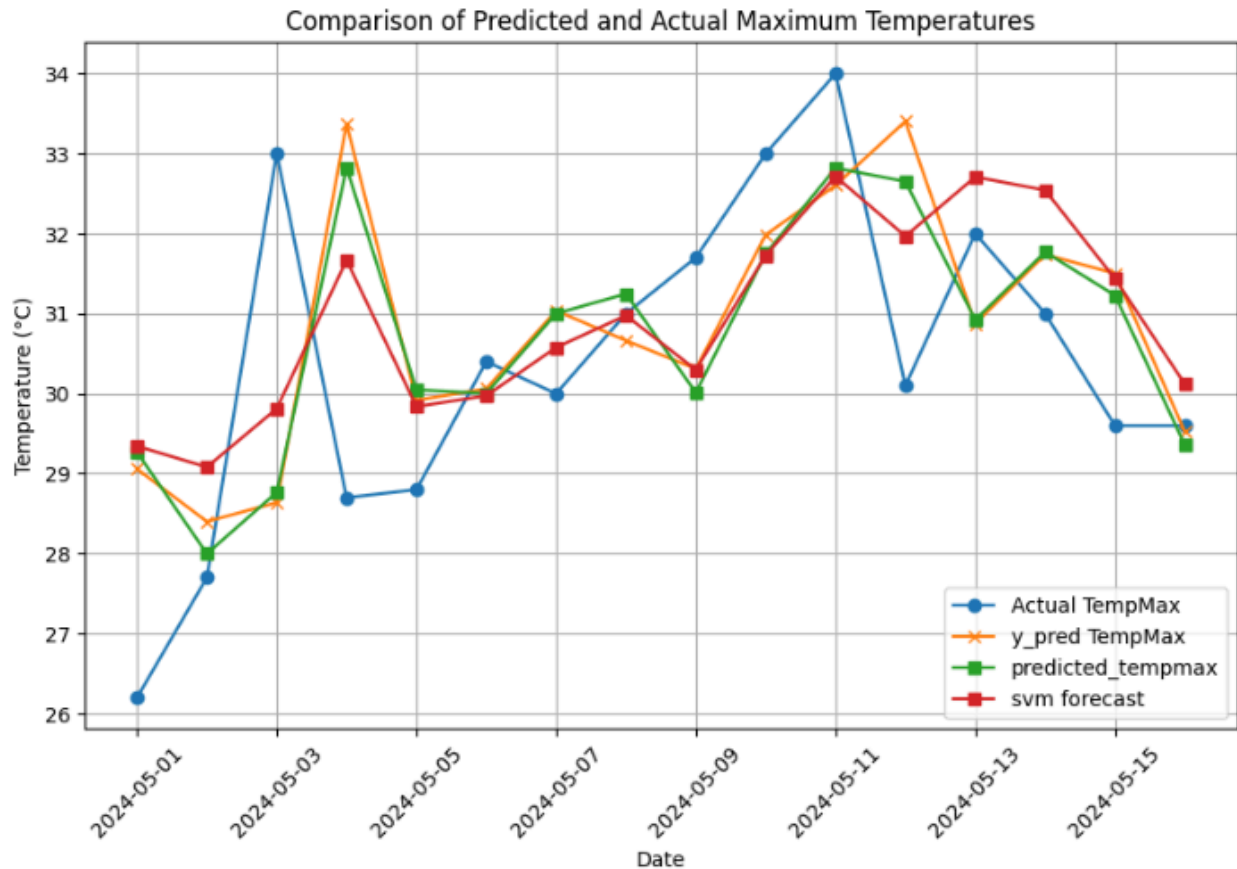


Mô hình Random Forest trải qua quá trình training xong tối ưu bằng GridSearchCV.

- Tiến hành training (y\_pred so với y\_test)  
y\_pred - Mean Absolute Error (MAE): 1.648125000000001  
y\_pred - Root Mean Squared Error (RMSE): 2.1407534596491926
- Tối ưu bằng GridSearchCV (predicted\_tempmax so với y\_test)  
predicted\_tempmax - Mean Absolute Error (MAE): 1.5630051039978468  
predicted\_tempmax - Root Mean Squared Error (RMSE): 2.000105835929

=> MAE và RMSE đã giảm đi 1 chút đúng với nguyện vọng và mục đích khi tối ưu bằng GridSearchCV. Mô hình khi tối ưu cũng bám sát với dữ liệu hơn so với y\_pred trước đó.

- Thử với mô hình SVM (Support vector machine)  
SVM Forecast- Root Mean Squared Error (RMSE): 1.4487094745676523  
SVM Forecast - Root Mean Squared Error (RMSE): 1.7248826721367774



### Đánh giá SVM rbf kernel

Đường màu đỏ là SVM forecast đường như có hiệu quả hơn so với 2 đường còn lại của model Random Forest.

- Support Vector Machine MSE: 2.97522
- Random Forest MSE: 4.58282

Có thể do sự khác nhau giữa 2 mô hình trong quá trình tối ưu hoá

- SVM tối ưu hóa biên giữa các lớp, tập trung vào các điểm dữ liệu **khó phân loại** (support vectors), điều này khiến cho SVR nắm bắt được xu hướng xuyên suốt dài thời gian và vô tình trong 15 ngày của tập test có sự biến động không nhiều so với xu hướng. Sai số của svr rbf kernel trên tập đánh giá thể hiện điều này:

```

print(mape)
print(mae)
print(mse)
31] ✓ 0.0s
0.096051026301672
2.4289386294685458
8.649584389579493

```

- Random Forest không có quá trình tối ưu hóa tương tự mà dựa vào việc **trung bình hóa** kết quả từ nhiều cây quyết định, điều này có thể dẫn đến mất thông tin chi tiết tại các điểm dữ liệu khó nhưng về dự đoán một khoảng thời gian bất kì khác dùng random forest chắc chắn sẽ đem lại hiệu quả tốt hơn SVM.
- Dữ liệu lấy từ Visual Crossing đường như có khá nhiều điểm dữ liệu khó phân loại, mà Random Forest sẽ dựa vào việc trung bình hoá kết quả nhiều nên đưa ra kết quả chưa sát, điều đó ta dễ dàng thấy thấy được qua chỉ số MSE của cả 2 mô hình SVM và Random Forest

### Đánh giá SVM mahalanobis kernel

Sai số trên tập đánh giá của mahalanobis-svm đã vượt qua baseline nhưng so với các mô hình khác đã được tối ưu thì vẫn thấp hơn rất nhiều.

```

In [24]: print(mape)
          print(mae)
          print(mse)
0.07170282458669604
1.8411209941225897
6.026884699765919

```

sai số trên tập test cũng không được như các mô hình khác

```

mape = mean_absolute_percentage_error(y_test, forecast)
mae = mean_absolute_error(y_test, forecast)
mse = mean_squared_error(y_test, forecast)
print(mape)
print(mae)
print(mse)

```

```

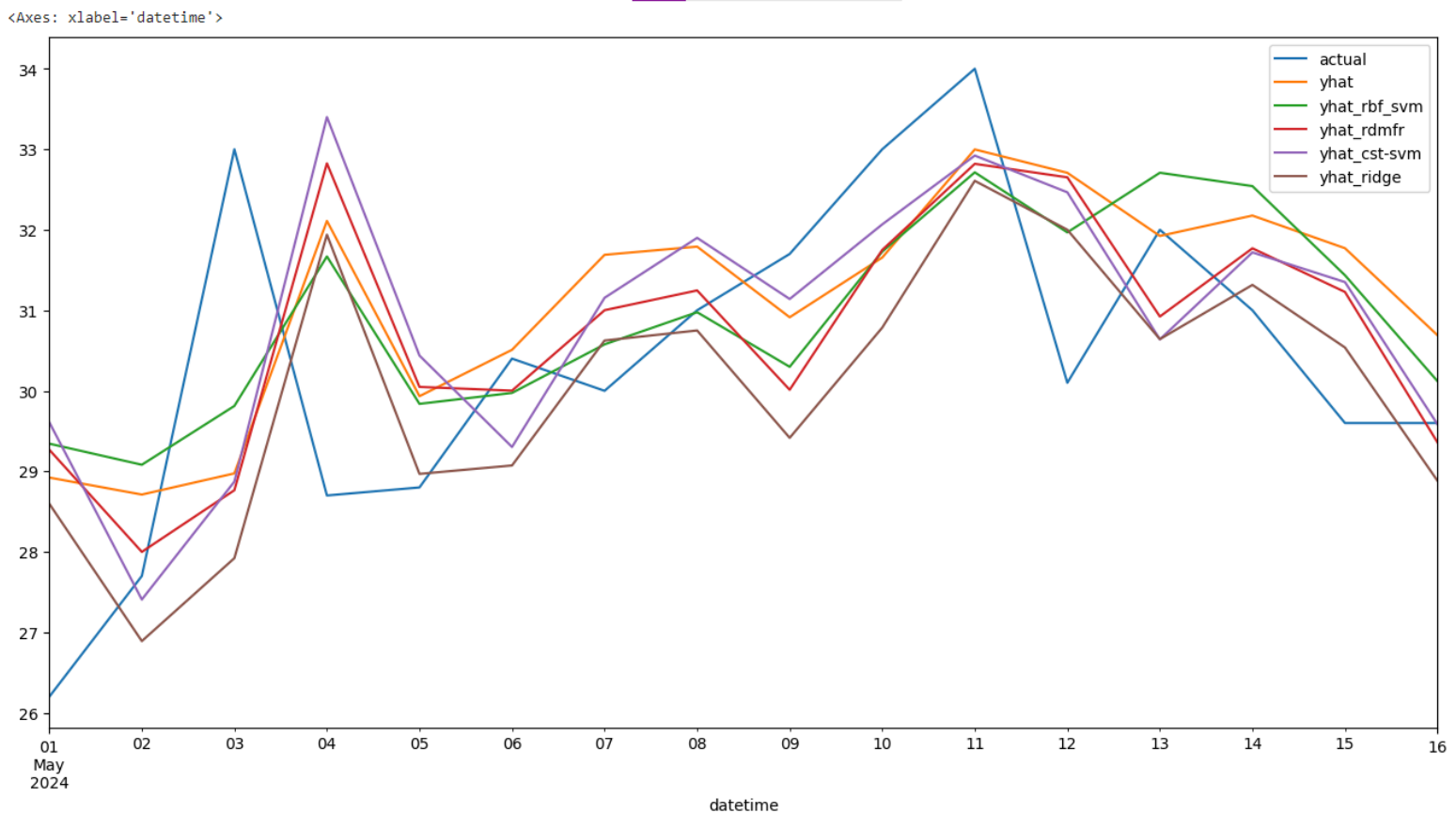
0.054481014127023206
1.6314770647924313
4.390548445143311

```

Cũng giống như rbf kernel nhưng về dự đoán khoảng thời gian dài mahalanobis kernel đã cải thiện được độ hiệu quả(kết quả trên tập đánh giá).

## Đánh giá chung toàn bộ mô hình

### Biểu đồ dự đoán trên tập test



Đường màu xanh dương (actual):

- Đây là đường biểu diễn nhiệt độ thực tế trong khoảng thời gian từ ngày 1/5/2024 đến ngày 16/5/2024.
- Đường này sẽ được sử dụng làm cơ sở để so sánh với các đường biểu diễn dự đoán.

Các đường dự đoán:

- Đường màu đỏ (yhat\_prophet): Đây là dự đoán từ một mô hình Prophet dự báo chuỗi thời gian mạnh mẽ, được thiết kế bởi Facebook. Đường này dường như khá sát với các giá trị thực tế tại một số điểm, nhưng cũng có những đoạn lệch khá xa.
- Đường màu xanh lá cây (yhat\_rbf\_svm): Đây là dự đoán từ mô hình SVM với kernel RBF. Đường này có xu hướng bám sát các giá trị thực tế khá tốt, đặc biệt là trong các khoảng biến đổi nhanh của dữ liệu.
- Đường màu cam (yhat\_rdmfr): Đây là dự đoán từ mô hình Random Forest. Đường này có xu hướng khá ổn định, nhưng đôi khi có thể không phản ứng kịp với những biến động đột ngột của dữ liệu.
- Đường màu hồng (yhat\_cst\_svm): Đây là dự đoán từ mô hình SVM với kernel mahalanobis. Mô hình này có một số điểm dự báo chính xác, nhưng cũng có những giai đoạn không theo sát dữ liệu thực tế, cho thấy nó có thể bị ảnh hưởng bởi sự lựa chọn các tham số kernel và đặc trưng của dữ liệu.
- Đường màu nâu (yhat\_ridge): Đây là dự đoán từ mô hình Ridge Regression. Đường này có thể hơi mượt hơn so với các mô hình khác, và có thể không phản ứng tốt với những biến động lớn.

Từ đồ thị ta nhận thấy:

#### **Độ chính xác:**

- Các mô hình yhat\_rbf\_svm và yhat\_cst\_svm dường như cho ra các dự đoán gần sát với giá trị thực tế trên tập test nhất. Điều này cho thấy các mô hình SVM (đặc biệt là với kernel RBF) hoạt động tốt trên tập dữ liệu này do kết quả thực tế nhiệt độ không lệch quá nhiều so với siêu mặt phẳng mà svr đã chia.
- Mô hình Random Forest (yhat\_rdmfr) cũng hoạt động tốt, đặc biệt với dự báo dài hạn, nhưng có thể kém hơn một chút so với các mô hình SVM trên tập test do việc bắt kịp các biến động đột ngột của dữ liệu.
- Mô hình Ridge Regression (yhat\_ridge) có thể mượt mà hơn, nhưng không bắt kịp tốt với những thay đổi nhanh chóng trong dữ liệu.
- Mô hình Prophet(yhat) dường như khá sát với các giá trị thực tế tại một số điểm, nhưng cũng có những đoạn lệch khá xa do thiết kế mô hình nắm bắt được trend và seasonality cùng với hệ số biến đổi nhỏ, thế nhưng lại có thể cung cấp một vùng nhiệt độ mà khả năng rất cao bao gồm cả dự báo thực tế.

#### **Phản ứng với biến động:**

- Mô hình RBF SVM phản ứng tốt với các biến động đột ngột, điều này có thể là do khả năng kernel của SVM ánh xạ dữ liệu vào không gian cao hơn.

- **Mô hình Random Forest có thể chậm hơn trong việc phản ứng với biến động do tính chất trung bình hóa của nhiều cây quyết định.**

**Xu hướng:** Tất cả các mô hình đều có xu hướng bám theo xu hướng tổng thể của dữ liệu thực tế, nhưng mức độ chính xác và khả năng phản ứng với biến động khác nhau.

#### Kết luận:

Biểu đồ trên tập test cho thấy rằng mô hình SVM với kernel RBF (`yhat_rbf_svm`) có hiệu suất dự đoán tốt nhất trong việc bắt kịp và phản ứng với các biến động của dữ liệu thực tế, trong khi các mô hình khác như Random Forest và Ridge Regression cũng cho kết quả tốt nhưng không thể bắt kịp nhanh bằng. Tuy nhiên, việc lựa chọn mô hình phù hợp còn phụ thuộc vào nhiều yếu tố khác như tính phức tạp của mô hình, yêu cầu về thời gian tính toán và khả năng diễn giải của mô hình.



