

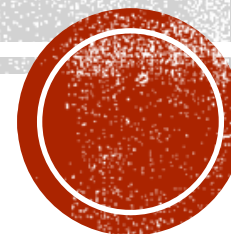
KHOA HỌC DỮ LIỆU

ĐỒ ÁN CUỐI KỲ

DỰ ĐOÁN THỜI TIẾT

18120356 – PHAN ANH HÀO

18120292 – NGUYỄN DƯƠNG BINH



NỘI DUNG

- - GIỚI THIỆU ĐỒ ÁN
- - THU THẬP DỮ LIỆU
- - KHÁM PHÁ DỮ LIỆU
- - TIỀN XỬ LÝ DỮ LIỆU
- - XÂY DỰNG MÔ HÌNH
- - ĐÁNH GIÁ KẾT QUẢ
- - NHÌN LẠI QUÁ TRÌNH
- - TÀI LIỆU THAM KHẢO




GIỚI THIỆU ĐỒ ÁN

- **Câu hỏi:** Dự đoán thời tiết có mưa hay không tại khu vực thành phố Hồ Chí Minh ?
- **Input:** Các thông số thời tiết
- **Output:** 0: Không mưa, 1: Có khả năng mưa, 2: Mưa
- **Ý nghĩa trong thực tế:** Nếu biết trời mưa, khi đi ra ngoài ta sẽ cầm theo dù (ô) hay là áo mưa, giúp cho các bác nông dân tránh bị "chạy thóc" khi gặp phải ngày mưa,...
- **Cảm hứng:** Vô tình vào tuần sau khi được nghỉ các môn học, em có về quê chơi với ông bà, vô tình vào 1 hôm em đi chơi xung quanh xóm thì trời bất ngờ đổ mưa, em chạy về nhà vô tình trên đường thấy nhiều nhà cô chú làm nông phải vội vã "chạy thóc", nếu không kịp sẽ bị mưa cuốn trôi, ảnh hưởng đến nhiều thứ nên bọn em đã quyết định đặt câu hỏi liên quan tới thời tiết.



THU THẬP DỮ LIỆU

Dữ liệu được lấy trên trang: [API - MetaWeather](#)

 MetaWeather beta

HomeLanguage ▾MapAPIAbout










Search

API

MetaWeather provides an API that delivers JSON over HTTPS for access to our data.

Drop me an email if you're going to make more than maybe a request a minute to this. We also ask that you link back to [MetaWeather.com](#) where appropriate, in a sensible way that's useful to the user.

Weather States

Name	Abbreviation	Icon
Snow	sn	
Sleet	sl	
Hail	h	
Thunderstorm	t	
Heavy Rain	hr	
Light Rain	lr	
Showers	s	
Heavy Cloud	hc	
Light Cloud	lc	



THU THẬP DƯ LIỆU

Đây là dữ liệu hợp pháp, hình bên dưới là file robots.txt

The image is a highly detailed ASCII art representation of a BMW car, viewed from a side profile. The car's body is outlined using a combination of dots (.), dashes (-), and various letters (J, L, F, Y, P, q, d, b, o, Y, P, q, d, b, o). The front of the car features a prominent grille with a circular BMW roundel logo at the top center. The logo consists of a circle with a cross inside, and the letters 'J', 'L', 'J', 'L', 'J', 'L' arranged around it. The car's wheels are depicted with a grid-like pattern of dots and dashes. The side of the car shows the 'BMW' logo on the front fender and a large, detailed wheel with a grid-like pattern. The overall style is reminiscent of early computer graphics or text-based art.

User-agent: *



KHÁM PHÁ DỮ LIỆU

Dữ liệu thô (chưa xử lý) bao gồm:

+ 71475 mẫu

+ 13 thuộc tính

ID	State Name	State Abbreviation	Wind Direction	Created	Applicable Date	Min Temp	Max Temp	The Temp	Wind Speed	Air Pressure	Humidity	Visibility	Predict
4581443174924288	Heavy Cloud	hc	102.747804	2016-01-28T15:50:47.001230Z	2016-01-28	24.7025	34.0925	33.79	8.587820	1013.0	59.0	11.895841	
6336625583849472	Heavy Cloud	hc	91.709815	2016-01-28T12:50:47.353250Z	2016-01-28	24.9525	34.0725	33.79	8.470320	1013.0	59.0	11.895841	
4559383551803392	Heavy Cloud	hc	91.709815	2016-01-28T09:50:46.720960Z	2016-01-28	24.9525	34.0725	33.79	8.470320	1013.0	59.0	11.895841	
4623773701505024	Light Cloud	lc	95.785326	2016-01-28T06:50:47.016290Z	2016-01-28	24.9225	35.4250	33.86	9.630337	1013.0	57.0	11.991221	
5238239505940480	Light Cloud	lc	95.785326	2016-01-28T03:51:02.272730Z	2016-01-28	24.9225	35.4250	33.86	9.630337	1013.0	57.0	11.991221	



KHÁM PHÁ DỮ LIỆU

Đây là thông tin của mỗi cột (thuộc tính)

***** DESCRIPTION WEATHER *****

Vị trí: Hồ Chí Minh City

Thời tiết 4 năm (2016,2017,2018,2019)

Predictability:

+ 80: Thunder

+ 77: Heavy Rain

+ 75: Light Rain

+ 73: Showers

+ 71: Heavy Cloud

+ 70: Light Cloud

+ 68: Clear

ID (integer): id của mỗi ngày trong từng năm

State Name (string): tên trạng thái của ngày (Clear,Light Cloud,...)

State Abbreviation (string): viết tắt của trạng thái (c,lc,...)

Wind Direction (float): Hướng gió

Created (datetime): Thời gian cụ thể trong ngày

Applicable Date (datetime): Ngày áp dụng

Min Temp (integer): Nhiệt độ tối thiểu

Max Temp (integer): Nhiệt độ tối đa

The Temp (integer): Nhiệt độ

Wind Speed (float): Tốc độ của gió

Air Pressure (float): Áp suất không khí

Humidity (float): Độ ẩm

Visibility (float): Khoảng cách nhìn thấy



KHÁM PHÁ DỮ LIỆU

- - **Bộ dữ liệu bị hiện tượng Imbalanced Dataset (Output chủ yếu là mưa)**
- - **Dữ liệu không có dòng bị trùng**
- - **Một vài cột còn bị thiếu dữ liệu ('The Temp', 'Humidity',....)**
- - **Thuộc tính dữ liệu chủ yếu là kiểu số**



TIỀN XỬ LÝ DỮ LIỆU

```
1 train_X_df.shape
```

```
(40026, 12)
```

```
1 train_y_sr.shape
```

```
(40026,)
```

```
1 val_X_df.shape
```

```
(17154, 12)
```

```
1 val_y_sr.shape
```

```
(17154,)
```

```
1 test_X_df.shape
```

```
(14295, 12)
```

```
1 test_y_sr.shape
```

```
(14295,)
```

- - Tách tập dữ liệu ban đầu thành tập train và tập test theo tỉ lệ (80/20)
- - Từ tập dữ liệu train ta tách thành 2 tập (tập train và tập validation) theo tỉ lệ (70/30)



```
*** Trước khi Undersampling tập train***  
(40026,)
```

```
: 2    79.933044  
  0    11.832309  
  1     8.234647  
Name: Predictability, dtype: float64
```

```
*** Sau khi Undersampling tập train ***  
(11328,)
```

```
0    41.807910  
2    29.096045  
1    29.096045  
Name: Predictability, dtype: float64
```

UNDERSAMPLING TẬP TRAIN



- Theo kinh nghiệm thời tiết để có thể phân tích được dữ liệu trên thì thông thường trong 1 năm, lượng mưa tập trung chủ yếu vào từ tháng 5 đến tháng 11 -> chúng ta sẽ quan tâm thêm về thông tin tháng, ngoài ra chúng ta sẽ quan tâm thêm thông tin về giờ trong ngày (vào những tháng mưa, vào các ngày trong tuần ta thấy mưa tập trung vào những khung giờ (theo kinh nghiệm em quan sát))
- Bỏ cột 'State Name' và cột 'State Abbreviation' vì đây là 2 cột tương tự giống với cột output
- Từ cột 'Created' và cột 'Applicable Date' ta rút trích dữ liệu tháng vào giờ thay vào đó chúng ta sẽ thêm cột 'Month' và cột 'Hour', sau đó xóa 2 cột 'Created' và 'Applicable Date' đi
- Bỏ cột 'Visibility' vì thiếu dữ liệu nhiều và thuộc tính này cũng không ảnh hưởng nhiều đến dự đoán của mô hình

	Wind Direction	Min Temp	Max Temp	The Temp	Wind Speed	Air Pressure	Humidity	Month	Hour
0	230.000000	24.220000	31.190000	31.380	3.167974	1009.940	NaN	6	7
1	50.500000	25.060000	35.980000	NaN	5.940000	NaN	56.0	1	3
2	41.635907	21.282000	33.220000	31.980	4.504709	1020.070	51.0	2	14
3	351.000000	21.593333	29.880000	NaN	2.110000	NaN	69.0	12	15
4	141.946152	23.545000	35.630000	34.850	10.654814	1012.000	58.0	3	20
...
11323	196.906226	24.782500	30.112500	29.330	5.203319	1011.270	84.0	7	14
11324	253.000000	24.373333	29.226667	30.680	3.842162	1008.100	NaN	10	21
11325	59.000000	23.485000	30.047500	27.230	7.474200	1012.710	79.0	11	0
11326	213.743990	25.796000	31.956000	30.705	6.017684	1008.895	80.0	5	23
11327	250.700950	25.057500	30.762500	31.380	7.034392	1007.655	81.0	7	8

THÊM XÓA CỘT



XỬ LÝ GIÁ TRỊ THIẾU VÀ CHUẨN HÓA DỮ LIỆU

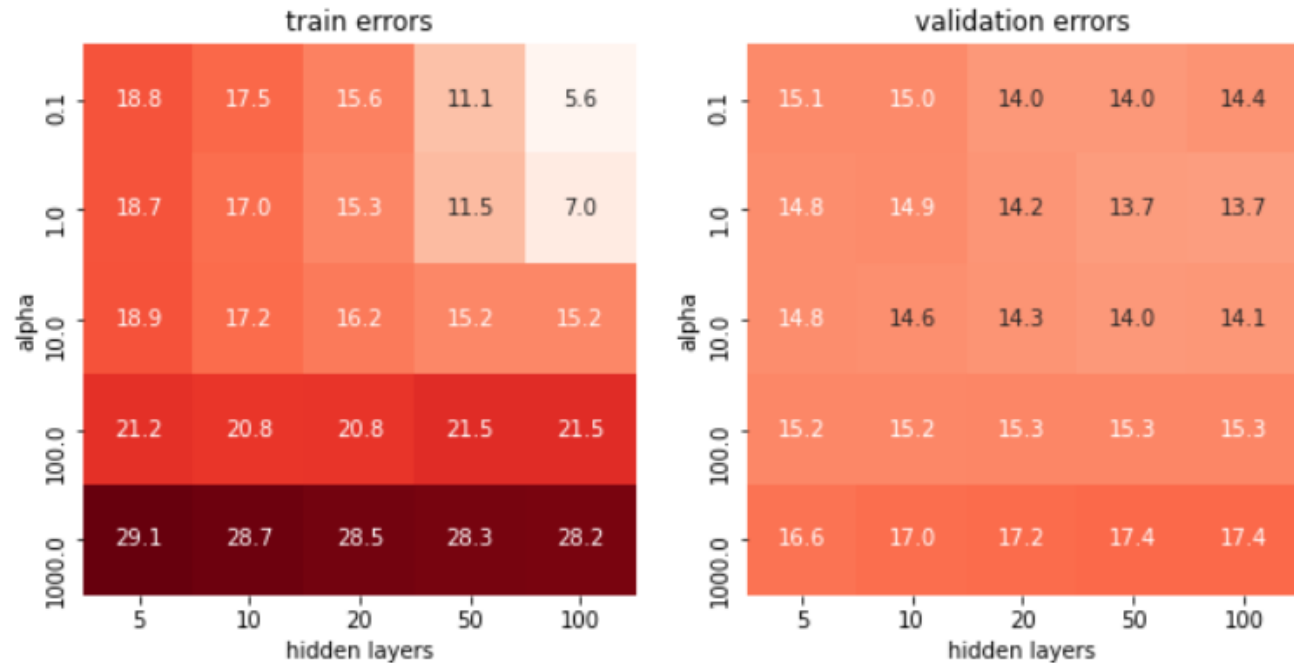
- Như ta phân tích ở trên thì thuộc tính có thuộc tính dạng số (numerical): 'Wind Direction', 'Min Temp', 'Max Temp', 'The Temp', 'Wind Speed', 'Air Pressure', 'Humidity', 'Visibility' và thuộc tính categorical: 'Month', 'Hour' (dạng số)
- + Với thuộc tính dạng kiểu số ta cần phải tính mean, với tất cả ta cần phải tính vì ta không biết được cột nào sẽ bị thiếu giá trị khi dự đoán với các véc-tơ input mới (Class MissingValues sẽ làm việc đó)
- + Với thuộc tính Categorical: Ta sẽ điền giá trị thiếu bằng mode (giá trị xuất hiện nhiều nhất) của cột, vì các cột categorical trong dữ liệu đều đã là dạng số nên ta không cần phải chuyển đổi chuẩn hóa nữa.
- Sau khi đã điền giá trị thiếu ta sẽ chuẩn hóa bằng cách trừ đi mean và chia cho độ lệch chuẩn của cột để giúp cho các thuật toán cực tiểu hóa như Gradient Descent, LBFGS, ... hội tụ nhanh hơn (Class Standarized sẽ làm việc này)



XÂY DỰNG MÔ HÌNH

- Với bộ dữ liệu này, ta sẽ thử với 4 mô hình
- + Mô hình Neuret Net
- + Mô hình Adaboost Classifier
- + Mô hình Decision Tree Classifier
- + Mô hình Logistic Regression

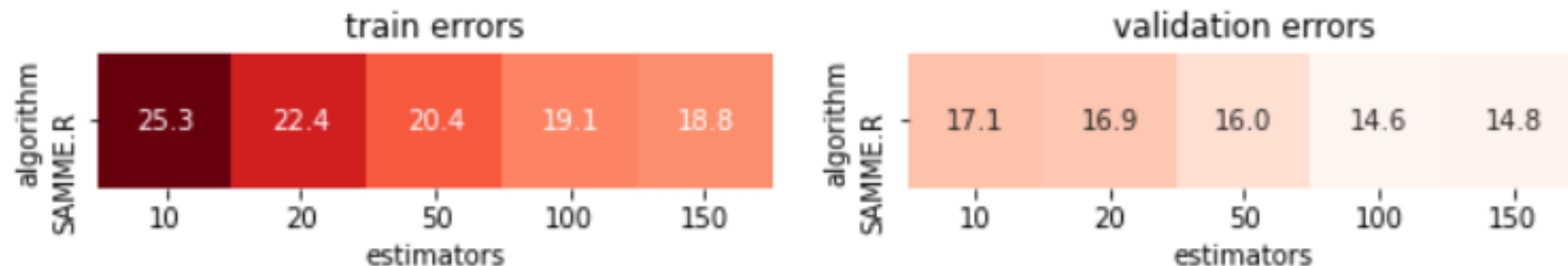




MÔ HÌNH NEURAL NET

- Ta sẽ sử dụng mô hình Neural Net để phân lớp. Ta sẽ tạo ra một pipeline từ đầu đến cuối bao gồm: các bước tiền xử lý ở trên + Neural Net (với các siêu tham số `hidden_layer_sizes=(20)`, `activation='tanh'`, `solver='lbfgs'`, `random_state=0`, `max_iter=2000`)
- Thử nghiệm mô hình với 2 tham số: `alpha`, `hidden layer`
- Đạt độ chính xác trên tập test: 86.85543





MÔ HÌNH ADABOOST CLASSIFIER

- Ta sẽ sử dụng mô hình AdaBoost Classifier để phân lớp. Bạn sẽ tạo ra một pipeline từ đầu đến cuối bao gồm: các bước tiền xử lý ở trên + AdaBoost Classifier
- Thử nghiệm với tham số estimator
- Đạt độ chính xác trên tập test: 84.1762



DECISION TREE VÀ LOGISTIC REGRESSION

- Ta sẽ thử nghiệm 2 mô hình này với tham số mặc định
- Độ chính xác của mô hình Decision Tree trên tập test: 87.023
- Độ chính xác của mô hình Logistic Regression trên tập test: 84.596



ĐÁNH GIÁ KẾT QUẢ

- - Mô hình Neuret Net
 - Neural chạy khá lâu (khoảng 15p) !!!
 - Bên train_errors, ta có thể thấy độ lỗi của mô hình dựa vào hidden layers , hidden layers càng cao thì càng fit mô hình , dễ bị overfitting
 - Dựa vào heatmap bên trên, mặc dù validation errors (hidden_layers = 100, alpha = 1.0) có độ lỗi nhỏ nhất nhưng khi nhìn qua ta có thể thấy bên train errors thì quá fit so với mô hình, tương tự với alpha = 0.1 thì ta cũng có thể thấy bị overfitting thay vào đó ta sẽ chọn hidden_layers = 50 , alpha = 1.0 làm tham số để chạy với bộ dữ liệu test



ĐÁNH GIÁ KẾT QUẢ

- - **Mô hình Adaboost Classifier**
 - Ở mô hình này ta có thể thấy, độ lỗi của mô hình giảm khi ta tăng chỉ số **estimators** (cả **train errors** và **validation errors**) Lý do **validation errors** thấp hơn so với **train errors** có thể do 1 phần lớn ảnh hưởng bởi việc **undersampling** tập train còn tập **validation** thì không
 - Thì em hiểu đây là một thuật toán học tăng cường, **n estimators** chính là số lượng 'học viên' đạt trọng số thấp cần phải đào tạo lại có thể kết hợp với nhiều bộ phân loại (với tham số **base_estimator=None**, mặc định là **DecisionTreeClassifier**)



ĐÁNH GIÁ KẾT QUẢ

- - Mô hình Decision Tree
 - Theo như quan sát độ lỗi của 2 tập (train và validation) thì với mô hình Decision Tree Classifier bị overfitting

Độ lỗi trên tập train: 0.0

Độ lỗi trên tập validation: 14.066689984843183



ĐÁNH GIÁ KẾT QUẢ

- - Mô hình Logistic Regression
 - Độ lỗi của tập train nhỏ hơn độ lỗi của tập validation, theo như em nghĩ ở trên thì có vẻ là do undersampling bộ train và giữ nguyên bộ validation

Độ lỗi trên tập train: 22.98728813559322

Độ lỗi trên tập validation: 15.337530605106686



NHÌN LẠI QUÁ TRÌNH

- - **Khó khăn:**
 - **Lúc tìm dữ liệu thời tiết, tuy nhiều trang API cung cấp dữ liệu về thời tiết nhưng có giới hạn về số lượng dữ liệu được lấy (thời gian) , lấy trang metaweather nhưng có vẻ dữ liệu thời tiết của nó không được đúng cho lắm :v**
 - **Trong quá trình làm, nhóm em có khó khăn trong vấn đề bị imbalanced dataset. (Nhóm em đã thử làm mà không cần under/oversampling dữ liệu mà chỉ điều chỉnh trọng số lớp trong thuật toán, tuy nhiên theo các bạn đóng góp cho đồ án của nhóm em và em cũng suy nghĩ mình cũng nên thử với việc undersampling dữ liệu coi có sự thay đổi gì không)**
 - **Tuy có cảm hứng làm với thời tiết nhưng nhóm em lại muốn làm gì đó nó mới mẻ hơn , có thử nghĩ qua một vài chủ đề khác nhưng cái cảm hứng của nhóm em nó lặn ất rồi ạ :))**



NHÌN LẠI QUÁ TRÌNH

- - **Những điều học được:**
 - **Biết thêm nhiều về Git , Github**
 - **Biết thêm về file markdown**
 - **Tăng khả năng làm việc nhóm**
 - **Biết được thêm nhiều mô hình dữ liệu hơn**
 - **Biết được thêm quá trình xử lý nếu dữ liệu bị mất cân bằng**
 - **Suy ngẫm về nhiều điều trong thực tế có thể mô hình hóa hay không ? (Tự đặt câu hỏi, thấy nhiều bài toán khá thú vị)**
 - **Ngoài ra còn được học hỏi được nhiều từ góp ý của các bạn nhóm khác, nhóm em thấy thầy làm thêm cái góp ý (issue) trong đồ án này khá hay**
- - **Nếu có thêm thời gian thì bọn em sẽ dành thời gian nhiều hơn cho việc suy nghĩ ý tưởng hơn và thử qua nhiều mô hình hơn**



TÀI LIỆU THAM KHẢO

- www.metaweather.com
- scikit-learn.org
- imbalanced-learn.org
- <https://datascience.stackexchange.com/questions/61858/oversampling-undersampling-only-train-set-only-or-both-train-and-validation-set>
- <https://datascience.stackexchange.com/questions/8895/with-unbalanced-class-do-i-have-to-use-under-sampling-on-my-validation-testing>



**CẢM ƠN THẦY VÀ CÁC BẠN ĐÃ ĐÓNG
GÓP CHO ĐỒ ÁN NHÓM MÌNH**

