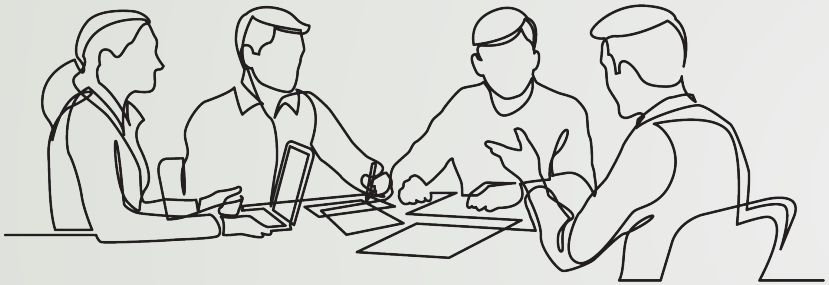
BÁO CÁO
ĐỒ ÁN THỰC HÀNH

Chủ đề: Thời tiết

Nhập môn khoa học dữ liệu - 20_21

Nhóm 09	2022



THÀNH VIÊN NHÓM

MSSV	HỌ VÀ TÊN
20120109	Trương Ngọc Huy
20120125	Bùi Anh Kiệt
20120598	Dương Tấn Tồn
20120614	Nguyễn Anh Tuấn

NỘI DUNG TRÌNH BÀY

01 THU THẬP DỮ LIỆU

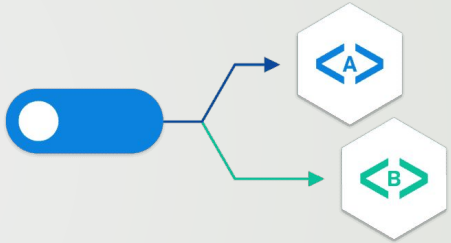
02 KHÁM PHÁ DỮ LIỆU

03 ĐẶT CÂU HỎI CÓ Ý NGHĨA

04 MÔ HÌNH HÓA DỮ LIỆU

01 THU THẬP DỮ LIỆU

- Trang web crawl data (tại đây)
- Dự định ban đầu crawl dựa vào html
- Phát hiện web có sử dụng api để lấy dữ liệu
- Cuối dùng crawl bằng api
- Có sử dụng key để lấy



Mô tả về API

Request

- Phương thức GET
- Key: lấy ở trang web
- Ngôn ngữ: English
- Location: HCM city
- Ngày lấy

Response

- Dự liệu nằm trong object observations:
- Trả về một ngày: gồm 48 object con nằm trong object observations. Nữa tiếng lấy một lần.
- Trả về tất cả các dự liệu bao gồm những trường không dùng đến => tiền xử lý dữ liệu.

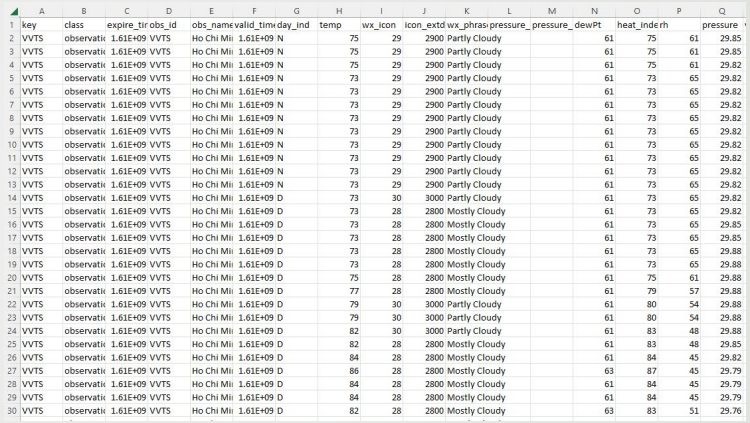
```
def getDataByDate(location,startDate,endDate):
    units = "e"
    apiKey = "e1f10a1e78da46f5b10a1e78da96f525"

    api = "https://api.weather.com/v1/location/"+location+"/observations/historical.json?apiKey="+apiKey+"&units="+units+"&startDate="+startDate+"&enc
    x = requests.get(api)

    return x.json()["observations"]
```

Một số trường dữ liệu quan trọng

- temp: nhiệt độ
- pressure_desc: áp suất
- rh: độ ẩm
- vis: sứcgió
- wspd: tốc độ gió
- precip_hrly: lượng mưa hàng giờ
- primary_wave_period: thủy triều
- uv_index: chỉ số uv

Dữ liệu sau khi được thu thập

- Dữ liệu sau khi được thu thập được lưu vào file weather-2021.csv
- Trong đo gồm dữ liệu thời tiết của năm 2021 qua tường ngày
- Mỗi ngày khoảng 48 dòng,.
- Có khoảng 17,500 dòng.
- Mỗi dòng khoảng 45 thuộc tính



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	key	class	expire_tir	obs_id	obs_name	valid_time	day_ind	temp	wx_icon	icon_extd	wx_phras	pressure_	pressure_	dewPt	heat_inde	rh	pressure
2	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	75	29	2900	Partly Cloudy			61	75	61	29.85
3	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	75	29	2900	Partly Cloudy			61	75	61	29.85
4	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	75	29	2900	Partly Cloudy			61	75	61	29.82
5	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
6	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
7	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
8	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
9	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
10	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
11	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
12	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
13	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	N	73	29	2900	Partly Cloudy			61	73	65	29.82
14	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	73	30	3000	Partly Cloudy			61	73	65	29.82
15	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	73	28	2800	Mostly Cloudy			61	73	65	29.82
16	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	73	28	2800	Mostly Cloudy			61	73	65	29.85
17	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	73	28	2800	Mostly Cloudy			61	73	65	29.85
18	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	73	28	2800	Mostly Cloudy			61	73	65	29.88
19	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	73	28	2800	Mostly Cloudy			61	73	65	29.88
20	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	75	28	2800	Mostly Cloudy			61	75	61	29.88
21	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	77	28	2800	Mostly Cloudy			61	79	57	29.88
22	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	79	30	3000	Partly Cloudy			61	80	54	29.88
23	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	79	30	3000	Partly Cloudy			61	80	54	29.88
24	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	82	30	3000	Partly Cloudy			61	83	48	29.88
25	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	82	28	2800	Mostly Cloudy			61	83	48	29.85
26	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	84	28	2800	Mostly Cloudy			61	84	45	29.82
27	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	86	28	2800	Mostly Cloudy			63	87	45	29.79
28	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	84	28	2800	Mostly Cloudy			61	84	45	29.79
29	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	84	28	2800	Mostly Cloudy			61	84	45	29.79
30	VVTS	observatio	1.61E+09	VVTS	Ho Chi Mi	1.61E+09	D	82	28	2800	Mostly Cloudy			63	83	51	29.76

02 KHÁM PHÁ DỮ LIỆU

- Tiền xử lý dữ liệu
- Kháp phá gom nhóm dữ liệu
- Xử lý nhiễu

Các câu hỏi đặt ra

- Đọc dữ liệu, tính số dòng và số cột
- Mỗi dòng có ý nghĩa gì? Có vấn đề các dòng có ý nghĩa khác nhau không?
- Dữ liệu có các dòng bị lặp không?
- Tỉ lệ giá trị thiếu của từng cột
- Mỗi cột có ý nghĩa gì?
- Mỗi cột hiện đang có kiểu dữ liệu gì? Có cột nào có kiểu dữ liệu chưa phù hợp để có thể xử lý tiếp hay không?
- Với mỗi cột có kiểu dữ liệu số, các giá trị phân bố như thế nào?
- Với mỗi cột có kiểu dữ liệu không phải dạng số, các giá trị được phân bố như thế nào?

Các câu hỏi đặt ra

Đọc dữ liệu, tính số dòng và số cột

- Import các thư viện cần thiết → pandas, numpy, …
- Đọc dữ liệu từ file sau khi cào về bằng pd.read_csv() và lưu vào biến df
- Tính số dòng và số cột thông qua df.shape()

Mỗi dòng có ý nghĩa gì? Có vấn đề các dòng có ý nghĩa khác nhau không?

- Mỗi dòng trong tập dữ liệu là thông tin và các chỉ số thời tiết của quận Tân Bình (TP.HCM) tại thời điểm nhất định (cập nhật 30 phút/lần)
- Có vẻ không có vấn đề các dòng có ý nghĩa khác nhau, tức là không có dòng nào bị “lạc loài”

Các câu hỏi đặt ra

Dữ liệu có các dòng bị lặp không?

- Sử dụng duplicated() và any() để kiểm tra xem có dòng nào bị lặp không. Kết quả sẽ trả về giá trị True nếu dữ liệu có dòng bị lặp, ngược lại trả về False → df.duplicated().any()

Tỉ lệ giá trị thiếu của từng cột

- Sử dụng isnull() để biết được giá trị thiếu, sau đó dùng sum() để tính tổng số giá trị thiếu theo từng cột. Cuối cùng chia cho tổng số dòng của dữ liệu để tính tỉ lệ giá trị thiếu của từng cột và lưu vào biến missing_ratio
- Loại bỏ một số cột không cần thiết. Có rất nhiều cột không có giá trị (tỉ lệ dữ liệu thiếu là 100%) → làm gọn → bỏ những cột không có ý nghĩa đó → lưu tên những cột vào list del_cols, sau đó dùng df.drop(del_cols, axis=1) để bỏ những cột đó đi

Các câu hỏi đặt ra

Mỗi cột có ý nghĩa gì?

- Dựa vào thông tin mô tả ở Thu thập dữ liệu, ta chọn ra những cột liên quan và cần thiết
    - Thời điểm biểu diễn các chỉ số thời tiết → ‘valid_time_gmt’
    - Thông tin về chỉ số thời tiết → 'temp', 'wx_phrase', 'dewPt', 'heat_index', 'rh', 'pressure', 'vis', 'wspd', 'uv_desc', 'feels_like', 'uv_index'
    - Như vậy các cột bị loại bỏ bao gồm → 'key', 'class', 'expire_time_gmt', 'obs_id', 'obs_name', 'day_ind', 'wx_icon', 'icon_extd', 'wc', 'wdir', 'wdir_cardinal', 'clds'
- Loại bỏ những cột không cần thiết . Lưu tên các cột cần bỏ vào list del_cols, sau đó df.drop(del_cols, axis=1)

Các câu hỏi đặt ra

Mỗi cột có ý nghĩa gì?

- Để dễ dàng ghi nhớ ý nghĩa của từng cột, tiến hành đổi tên cột bằng rename()
    - 'valid_time_gmt' → 'Time', temp' → 'Temperature', 'wx_phrase' → 'Condition', 'dewPt' → 'Dew Point', 'heat_index' → 'Heat Index', 'rh' → 'Humidity', 'pressure' → 'Pressure', 'vis' → 'Wind Force', 'wspd' → 'Wind Speed', 'uv_desc' → 'UV Description', 'feels_like' → 'Temperature Feels Like', 'uv_index' → 'UV Index'
- Thống kê mô tả của từng cột: dùng describe() để tính thống kê mô tả của các cột numeric

Các câu hỏi đặt ra

Mỗi cột có kiểu dữ liệu gì? Có cột nào có kiểu dữ liệu chưa phù hợp để có thể xử lý tiếp hay không?

- Sử dụng info() để kiểm tra thông tin của các cột
- Nhận xét: Cột ‘Time’ nên là dữ liệu datetime, nhưng hiện tại có kiểu numeric → Đưa giá trị cột ‘Time’ về dạng datetime
    - Sử dụng datetime.fromtimestamp để đưa dữ liệu số về dạng datetime
    - Sử dụng apply() để áp dụng cho toàn bộ dữ liệu trong cột ‘Time’
- Xem xét tập giá trị của các thuộc tính phân loại
    - Xem xét mỗi thuộc tính phân loại có bao nhiêu giá trị phân biệt bằng set()
    - Sau khi áp dụng cho 2 cột ‘UV Description’ và ‘Condition’, nhận thấy rằng:

Các câu hỏi đặt ra

Mỗi cột có kiểu dữ liệu gì? Có cột nào có kiểu dữ liệu chưa phù hợp để có thể xử lý tiếp hay không?

- ‘UV Description’: bình thường
- Condition’: có nhiều loại điều kiện thời tiết (24 loại), nhưng xuất hiện nhiều loại có thể xếp chung vào 1 nhóm

- Xem xét tập giá trị của các thuộc tính phân loại
  - Phân nhóm ‘Conditon’: phân chia các loại vào 6 nhóm: 'Cloudy', 'Fair', 'Fog / Haze', 'Rain', 'T-Storm', 'Thunder'

Các câu hỏi đặt ra

Với mỗi cột có kiểu dữ liệu số, các giá trị phân bố như thế nào?

- Với các cột có kiểu dữ liệu số, ta sẽ tính:
  - Tỉ lệ % (từ 0 đến 100) các giá trị thiếu
  - Giá trị min
  - Giá trị lower quartile (phân vị 25)
  - Giá trị median (phân vị 50)
  - Giá trị upper quartile (phân vị 75)
  - Giá trị max
- Lưu kết quả vào DataFrame num_col_info_df, trong đó:Xem xét tập giá trị của các thuộc tính phân loại
  - Tên của các cột là tên của các cột số trong df
  - Tên của các dòng là: missing_ratio, min, lower_quartile, median, upper_quartile, max

Các câu hỏi đặt ra

Với mỗi cột có kiểu dữ liệu số, các giá trị phân bố như thế nào?

row_name	Temperature	Heat Index	Temperature Feels Like	Dew Point	Humidity	Wind Force	Wind Speed	Pressure	UV Index
missing_ratio	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.10	0.0
min	64.0	64.0	64.0	46.0	21.0	0.37	0.0	29.52	0.0
lower_quartile	79.0	82.0	82.0	72.0	66.0	6.00	3.0	29.70	0.0
median	82.0	88.0	88.0	75.0	79.0	6.00	6.0	29.76	0.0
upper_quartile	86.0	95.0	95.0	77.0	89.0	6.00	8.0	29.82	4.0
max	99.0	114.0	114.0	84.0	100.0	6.00	31.0	29.97	15.0

Các câu hỏi đặt ra

Với mỗi cột có kiểu dữ liệu không phải dạng số, các giá trị được phân bố như thế nào?

- Thực hiện thống kê và lưu vào một dataframe với các dòng là đại diện cho các giá trị như sau:
  - Tỉ lệ % (từ 0 đến 100) các giá trị thiếu (missing_ratio).
  - Số lượng các giá trị khác nhau (không xét giá trị thiếu) (num_values).
  - Tỉ lệ % (từ 0 đến 100) của mỗi giá trị được sort theo tỉ lệ % giảm dần (không xét giá trị thiếu, tỉ lệ là tỉ lệ so với số lượng các giá trị không thiếu): dùng dictionary để lưu, key là giá trị, value là tỉ lệ % (value_ratios).

row_name	UV Description	Condition
missing_ratio	0.0	0.0
num_values	5	6
value_ratios	{'Low': 70.6, 'Moderate': 11.6, 'High': 9.0, '...	{'Cloudy': 73.9, 'Fair': 18.8, 'Rain': 5.7, 'T...

05

ĐẶT CÂU HỎI CÓ Ý NGHĨA

- Đặt câu hỏi
- Trực quan hóa dữ liệu.

Câu hỏi 1

Mối tương quan giữa nhiệt độ và chỉ số UV là gì?

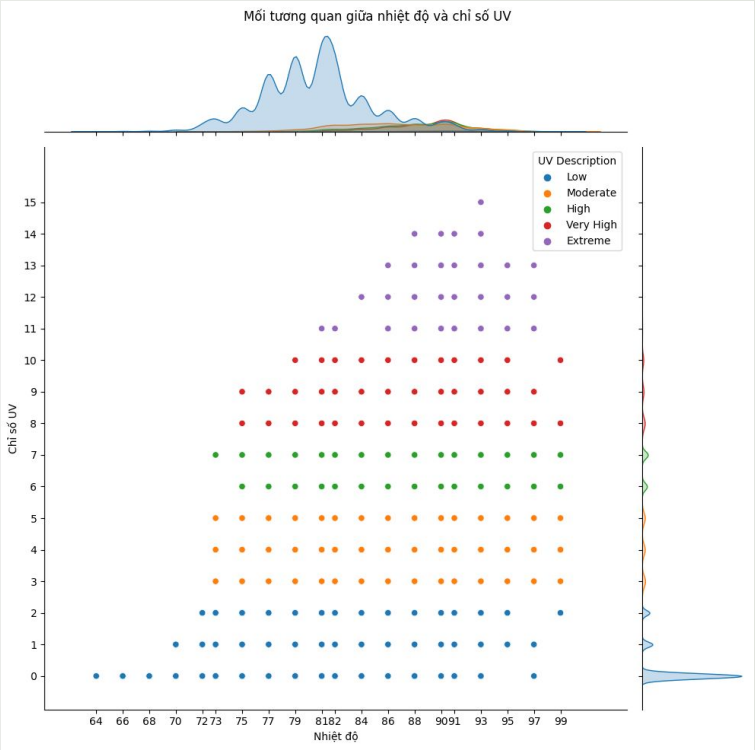
Lợi ích:

Giúp chúng ta nắm rõ được điều kiện thời tiết như thế nào trong năm để có thể có sự chuẩn bị trước cho thời tiết sắp tới. Đặc biệt là khách du lịch hoặc những người mới chuyển vào sinh sống ở đây.

Nguồn cảm hứng:

Nhóm tự suy nghĩ từ trong quá trình khám phá dữ liệu.

Trực quan hóa dữ liệu:



Nhận xét:

Mối tương quan chung giữa nhiệt độ và chỉ số UV là chỉ số UV càng tăng thì khoảng nhiệt độ càng thu hẹp và có xu hướng lệch về phía nhiệt độ cao.

Các mức độ UV:

- Thấp (Low):
  + Khoảng chỉ số UV từ 0 đến 2.
  + Khoảng nhiệt độ từ 64°F đến 99°F.
  + Chỉ số UV bằng 0 có khoảng nhiệt độ trải dài nhất và hơn hẳn các chỉ số còn lại.
- Vừa (Moderate), Cao (High):
  + Khoảng chỉ số UV lần lượt từ 3 đến 5 và từ 6 đến 7.
  + Khoảng nhiệt độ từ 73°F đến 99°F.
- Rất Cao (Very High):
  + Khoảng chỉ số UV lần lượt từ 8 đến 10.
  + Khoảng nhiệt độ từ 73°F đến 99°F.
- Cực Độ (Extreme):
  + Khoảng chỉ số UV lần lượt từ 11 đến 15.
  + Khoảng nhiệt độ từ 81°F đến 97°F và thu hẹp đáng kể theo chiều tăng chỉ số UV.

Câu hỏi 2

Nhiệt độ trung bình trong ngày theo từng tháng như thế nào?

Lợi ích:

Giúp ta biết được trung bình nhiệt độ theo giờ trong ngày theo từng tháng. Từ đó, đưa ra kết luận về khoảng thời gian nóng hay lạnh trong ngày theo từng tháng.

Nguồn cảm hứng:

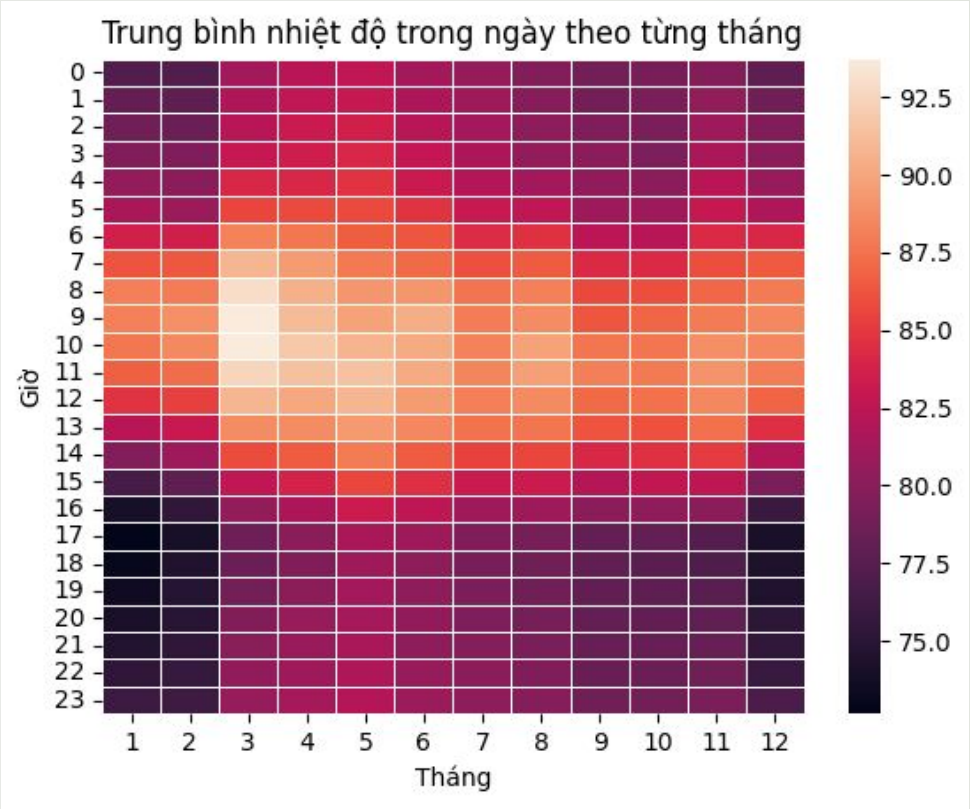
Từ Lab 3 môn Nhập môn Khoa học dữ liệu.

Phân tích và trích xuất dữ liệu:

Tạo dataframe là một ma trận 24x12 chứa trung bình nhiệt độ theo giờ trong ngày theo từng tháng.

	1	2	…	12
0	77.145161	77.178571	…	77.866667
1	78.193548	77.928571	…	78.716667
…	…	…	…	…
23	76.100000	76.185185	…	76.883333

Trực quan hóa dữ liệu:



Nhận xét:

- Khoảng thời gian nóng trong ngày (trung bình từ 85°F) phình to từ tháng 3 đến tháng 8, cho thấy tại những tháng này thời gian nóng trong ngày là dài nhất.
Đồng thời, cũng ở những tháng này, nhiệt độ trung bình trong ngày không dưới 80°F nên có thể kết luận đây là những tháng nóng trong năm.
- Khoảng thời gian nóng trong ngày co lại ở những tháng còn lại. Đặc biệt, các tháng 1, 2 và 12 có những thời điểm nhiệt độ trung bình trong ngày hạ xuống dưới 75°F, ta nhận định đây là những tháng lạnh trong năm.
- Nhiệt độ trung bình cao nhất năm rơi (trên 92,5°F) vào từ 9 đến 10 giờ ở tháng 3.
- Nhiệt độ trung bình thấp nhất năm rơi (dưới 75°F) vào từ 17 đến 19 giờ ở tháng 1.

Câu hỏi 3

Sự tương quan giữa nhiệt độ không khí, nhiệt độ điểm sương và độ ẩm?

Lợi ích:

Giúp ta biết được sự tương quan chung giữa 3 chỉ số thời tiết quan trọng: nhiệt độ không khí, nhiệt độ điểm sương và độ ẩm.

Nguồn cảm hứng:

Nhóm tự nghĩ ra khi khám phá dữ liệu.

Phân tích và trích xuất dữ liệu:

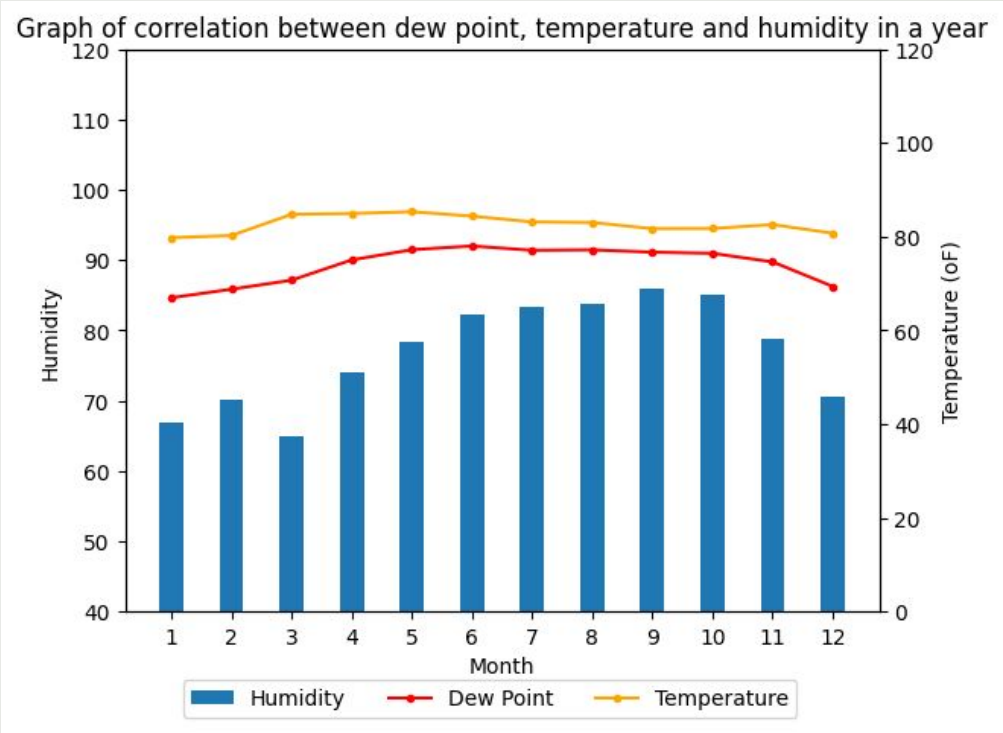
- Tính nhiệt độ trung bình mỗi tháng.

- Tính điểm sương trung bình mỗi tháng.

- Tính độ ẩm trung bình mỗi tháng.

Phương thức chung: Tính tổng giá trị của từng thuộc tính theo từng tháng rồi chia lại số lượng để lấy số trung bình.

Trực quan hóa dữ liệu:



Nhận xét:

- Nhiệt độ phân bố tương đối đều giữa các tháng, độ chênh lệch giữa tháng có nhiệt độ cao nhất và tháng có nhiệt độ thấp nhất là rất thấp.

- Độ ẩm trong không khí cao, hầu hết các tháng đều có độ ẩm trung bình trên 70% (trừ tháng 1 và tháng 3).

- Nhiệt độ điểm sương gần như là tỉ lệ thuận với nhiệt độ không khí.

Khi độ ẩm càng cao thì nhiệt độ điểm sương càng gần với nhiệt độ không khí.

Câu hỏi 4

Các điều kiện thời tiết phân bố như thế nào trong năm?

Lợi ích:

Giúp chúng ta nắm rõ được điều kiện thời tiết như thế nào trong năm để có thể có sự chuẩn bị trước cho thời tiết sắp tới. Đặc biệt là khách du lịch hoặc những người mới chuyển vào sinh sống ở đây.

Nguồn cảm hứng:

Trải nghiệm của bản thân trong năm đầu tiên vào học và sinh sống ở TPHCM.

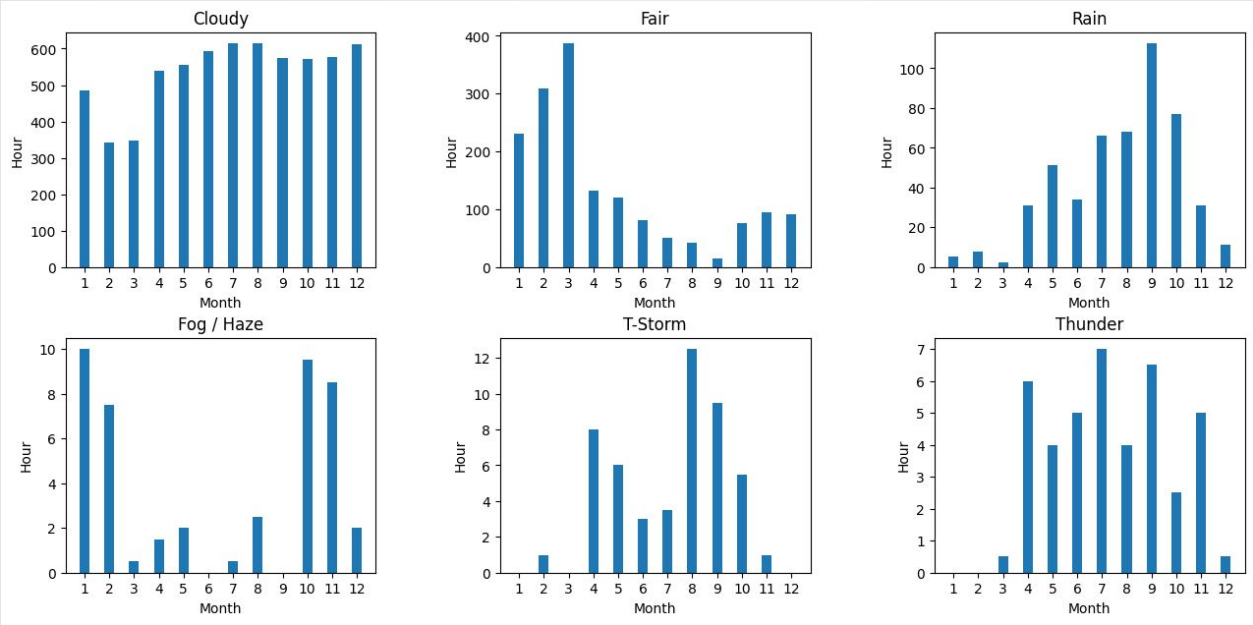
Phân tích và trích xuất dữ liệu:

- Gom nhóm các điều kiện thời tiết theo từng tháng.

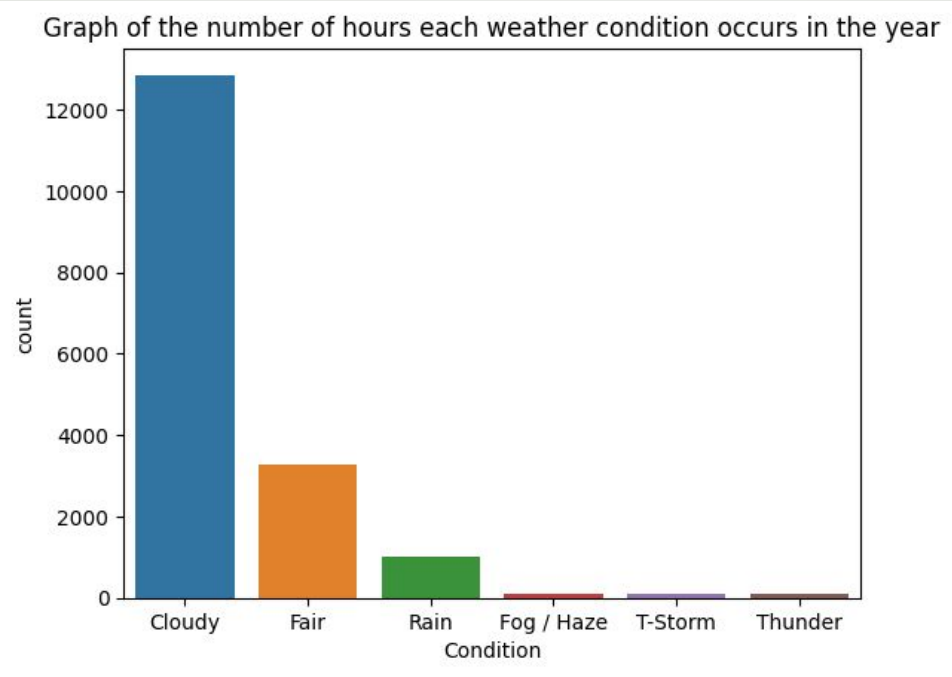
- Lưu giá trị vào một dictionary với key là các điều kiện thời tiết, value là 1 dictionary khác

với key là các tháng trong năm và value là số giờ xuất hiện điều kiện thời tiết đó trong tháng.

Trực quan hóa dữ liệu



Trực quan hóa dữ liệu



Nhận xét:

Tổng quát:

- Điều kiện thời tiết Cloudy (có mấy), Fair (đẹp) và Rain (mưa) chiếm phần lớn trong năm, đặc biệt là Cloudy với hơn 12000h.

- Các kiểu thời tiết: Fog/Haze (sương mù), T-Storm (giông), và Thunder (sấm sét) xuất hiện ít trong năm với tháng cao nhất chỉ khoảng 13h/tháng.

Nhận xét:

Cụ thể:

- Cloudy:

Xuất hiện rất thường xuyên trong năm (trên 300h mỗi tháng), chỉ có tháng 1, 2 và 3 là dưới 500h mỗi tháng, còn lại đều trên 500h.

- Fair:

+ Xuất hiện nhiều vào các tháng đầu năm (tháng 1,2,3) với hơn 200h mỗi tháng, cao điểm nhất là tháng 3 với gần 400h.

+ Các tháng còn lại trong năm thời gian xuất hiện ít, dưới 150h/tháng, chạm đáy là tháng 9 (vì lúc này là đỉnh điểm của mùa mưa).

- Rain:

- Mưa ít ở các tháng đầu năm và cuối năm (tháng 1,2,3,11,12) với thời gian mưa chỉ dưới 20h/tháng.

- Mưa nhiều bắt đầu từ tháng 7 đến tháng 10 → mùa mưa bắt đầu, đỉnh điểm là tháng 9 với gần 120h mưa.

Nhận xét:

Cụ thể:

- Fog / Haze:
  + Sương mù nhiều vào các tháng đầu và cuối năm (tháng 1,2,10,11) với thời gian trên 6h/tháng.
  + Sương mù xảy ra ít vào các tháng giữa năm (từ tháng 3 đến tháng 9).

- T-Storm:
  + Mưa giông xảy ra nhiều từ tháng 4 đến tháng 10, nhiều nhất vào tháng 8 và tháng 9, đặc biệt là tháng 8.
  → Nguyên nhân: lúc đó là thời gian bước vào mùa mưa cho nên giông xuất hiện nhiều.

- Thunder:
  + Sấm sét phân bố khá đều ở các tháng giữa và cuối năm (từ tháng 1 đến tháng 11).
  → Nguyên nhân: Thông thường sấm sét thường đi kèm với mưa giông.
  + Xuất hiện ít vào 3 tháng đầu năm (tháng 1,2,3) và tháng 12.
  → Nguyên nhân: vào 3 tháng đầu năm là mùa xuân, thời tiết đẹp nên ít xảy ra sấm sét. Tháng 12 ít sấm sét vì lúc đó đã bắt đầu kết thúc mùa mưa chuyển sang mùa nắng.

Câu hỏi 5

Điều thời tiết nhiều mây (Cloudy) phân bố như thế nào trong khoảng thời gian từ 6h đến 18h các ngày trong tuần?

Lợi ích:

Giúp chúng ta biết được những khoảng thời gian có thời tiết thuận lợi cho các hoạt động sinh hoạt ngoài trời trong mỗi ngày của cả tuần.

Nguồn cảm hứng:

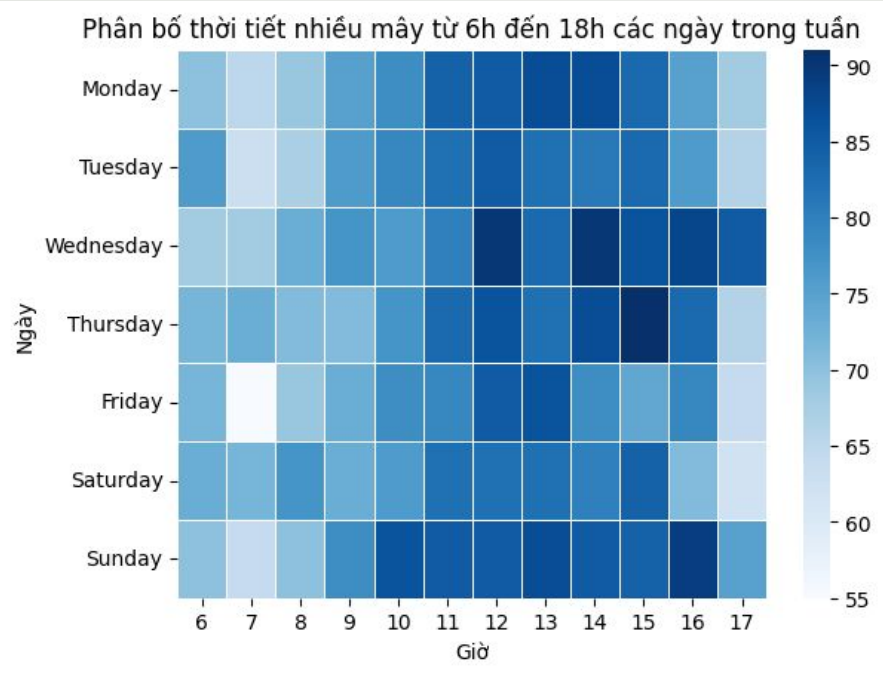
Mong muốn không bị thời tiết làm ảnh hưởng đến buổi đi chơi với bạn bè.

Phân tích và trích xuất dữ liệu:

Tạo dataframe là một ma trận 12x7 biểu diễn số lần thời tiết là nhiều mây trong khoảng thời gian từ 6h đến 18h theo từng ngày.

	6	7	…	17
Monday	70	65	…	68
Tuesday	76	63	…	66
…	…	…	…	…
Sunday	70	64	…	75

Trực quan hóa dữ liệu



Nhận xét:

- Thời tiết có nhiều mây hơn từ 9h mỗi ngày và thích hợp để bắt đầu một buổi đi chơi.
- Thứ ba và thứ bảy ít xuất hiện thời tiết nhiều mây hơn.
- Thứ tư, thứ năm và chủ nhật có tần suất xuất hiện thời tiết nhiều mây cao, đặc biệt trong khoảng thời gian từ 14h đến 17 giờ.
- Thứ hai, thứ tư, thứ năm và chủ nhật có khoảng thời gian liên tục thời tiết nhiều mây dài, kéo dài từ 11h đến 16 giờ.
- Kết luận: Thời gian phù hợp cho các hoạt động ngoài trời là buổi trưa chiều (11h đến 17h) các ngày thứ hai, thứ tư, thứ năm và chủ nhật.

Câu hỏi 6

Những thời điểm nào chỉ số UV vượt ngưỡng cho phép?

Lợi ích:

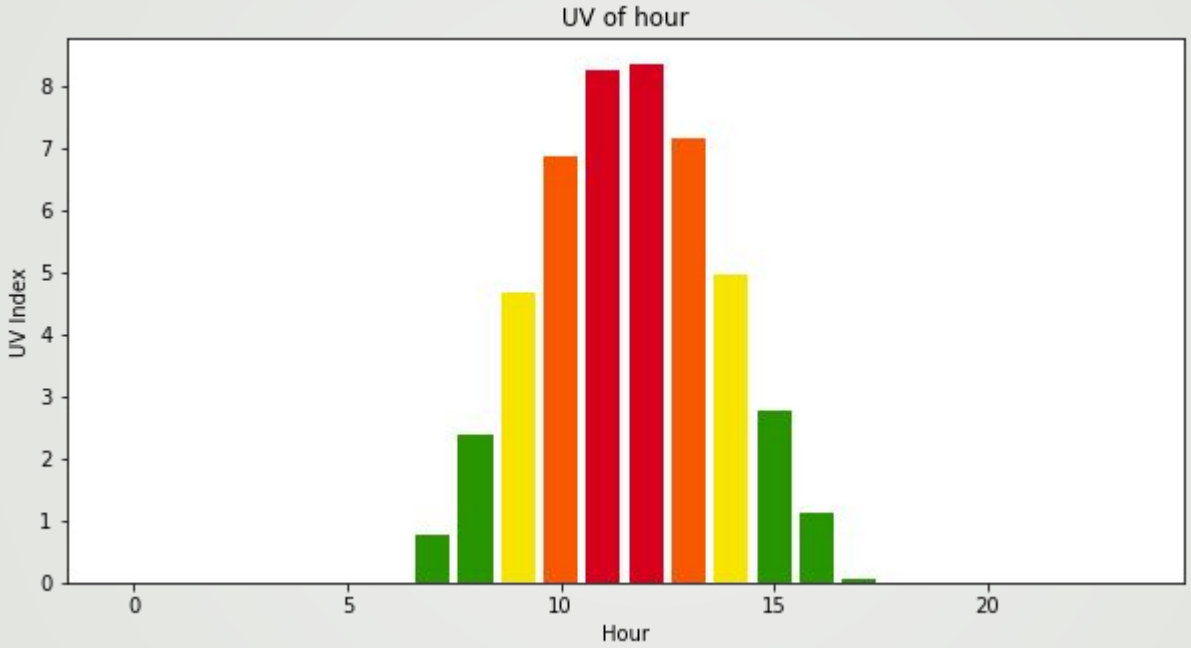
Giúp ta biết được những khoảng thời gian có chỉ số UV cao,

gây nguy hiểm với da người từ đó tránh tiếp xúc trực tiếp với ánh nắng.

Nguồn cảm hứng:

Mong muốn thời tiết không ảnh hưởng đến làn da cũng như sức khỏe chúng ta.

Trực quan hóa dữ liệu



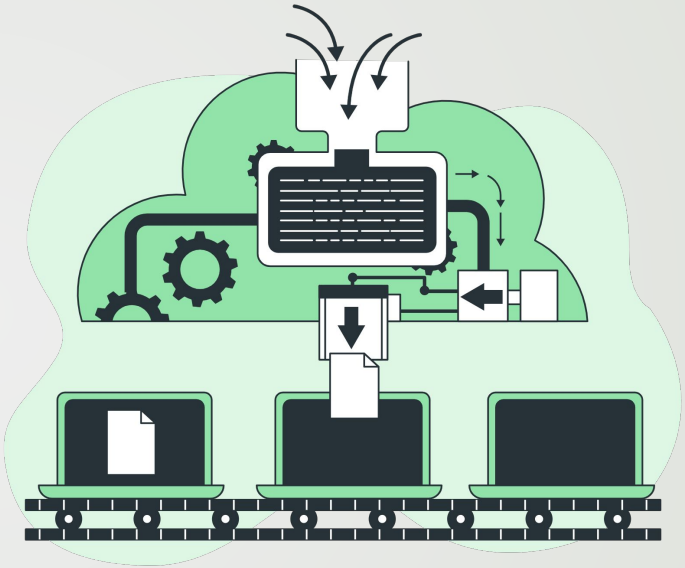
Nhận xét:

- Thời tiết trở nên khắc nghiệt hơn trong lúc từ 9 đến 14 giờ.
- Vào khoảng 9 và 14 giờ chỉ số UV nằm ở mức nguy cơ gây hại trung bình.
- Vào khoảng 10 và 13 giờ chỉ số UV nằm ở mức nguy cơ gây hại cao.
- Vào khoảng 11 và 12 giờ chỉ số UV nằm ở mức nguy cơ gây hại nằm ở mức rất cao, gây hại đến cơ thể.
- Kết luận: Không nên ra ngoài vào khoảng thời gian 11h - 13h nếu thật sự không cần thiết vì tia UV lúc này có nguy cơ gây hại rất cao, vào khoảng 10h và 14h nếu cần ra ngoài thì phải có che chắn để đảm bảo an toàn cho cơ thể mình.

06

DATA MODELING/

- Lựa chọn hình mô hình
- Huấn luyện và đánh giá.



Bài toán đặt ra

Dự đoán điều kiện thời tiết là nhiều mây hay không?

Lựa chọn thuật toán máy học

Thuật toán phân lớp (Classification Algorithm)

- Phân lớp (classification) là một tiến trình xử lý nhằm xếp các mẫu dữ liệu hay các đối tượng vào một trong các lớp đã được định nghĩa trước. Các mẫu dữ liệu hay các đối tượng được xếp về các lớp dựa vào giá trị của các thuộc tính (attributes) của mẫu dữ liệu hay đối tượng đó. Sau khi đã xếp tất cả các đối tượng đã biết trước vào các lớp tương ứng, lúc này mỗi lớp được đặc trưng bởi tập các thuộc tính của các đối tượng chứa trong lớp đó.

- Về bài toán "Dự đoán điều kiện thời tiết là nhiều mây hay không?", ở đây ta có 2 loại dữ liệu "có mây" và "không mây". Vì thế chọn Classification Algorithm là phù hợp nhất.

Tiền xử lý dữ liệu đầu vào mô hình

- Tiến hành thay đổi tập giá trị ở cột `Condition` thành ‘Cloudy’ và ‘Not Cloudy’.
- Ở cột `Time` ta chỉ lấy tháng.
- `UV Description` và `UV Index` có mức độ tương quan nhất định, chỉ số UV có thể biểu thị cho mức độ UV nên việc có cột `UV Description` trong quá trình này là không cần thiết.
- Cột `Condition` là dạng categorical nên trước khi đưa vào mô hình, ta phải đưa về dạng numerical.

	Time	Temperature	Heat Index	Temperature Feels Like	Dew Point	Humidity	Wind Force	Wind Speed	Pressure	UV Index	Condition
0	1	75	75.0	75.0	61.0	61.0	6.0	8	29.85	0	0
1	1	75	75.0	75.0	61.0	61.0	6.0	7	29.85	0	0
2	1	75	75.0	75.0	61.0	61.0	6.0	7	29.82	0	0
3	1	73	73.0	73.0	61.0	65.0	4.0	6	29.82	0	0
4	1	73	73.0	73.0	61.0	65.0	4.0	6	29.82	0	0
...	...	...	...	...	...	...	...	...	...	...	...
17385	12	81	82.0	82.0	64.0	58.0	6.0	8	29.94	0	0
17386	12	79	81.0	81.0	64.0	61.0	6.0	6	29.94	0	0
17387	12	79	81.0	81.0	64.0	61.0	6.0	6	29.94	0	0
17388	12	79	81.0	81.0	64.0	61.0	6.0	8	29.94	0	0
17389	12	79	81.0	81.0	64.0	61.0	6.0	8	29.94	0	0

17390 rows × 11 columns

Huấn luyện và đánh giá mô hình

- Tách tập dữ liệu thành feature và label.
- Tách tập dữ liệu thành tập huấn luyện (training set) và tập kiểm tra (test set).

```
Training set shape:  (13042, 10) (13042,)
Testing set shape:  (4348, 10) (4348,)
```

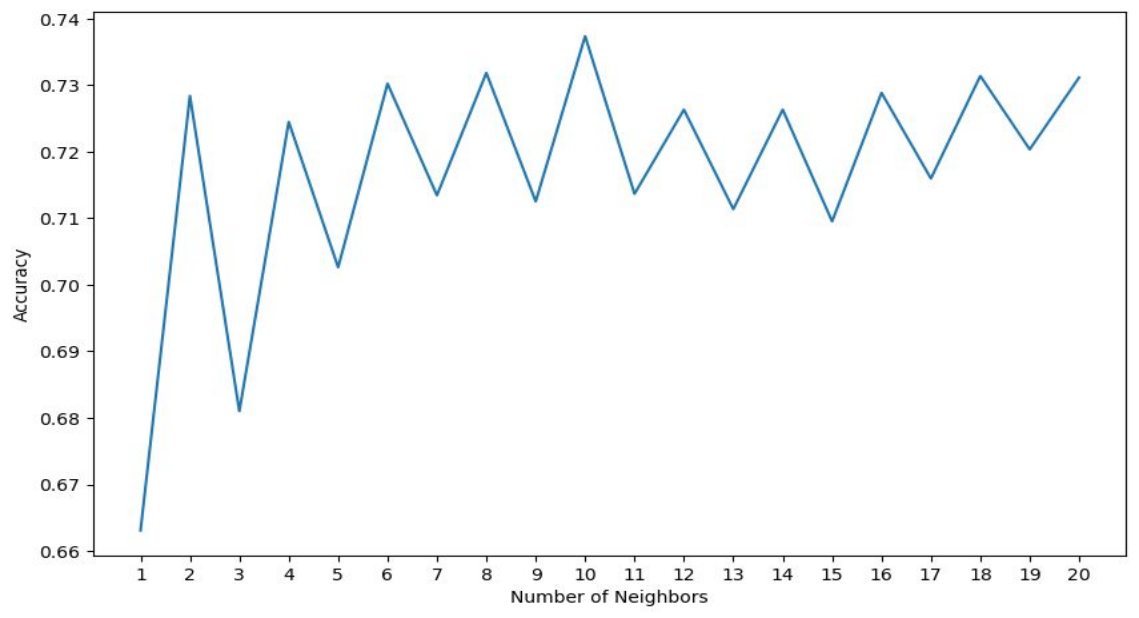
- Sau khi đã tách tập dữ liệu thành tập huấn luyện và tập kiểm tra, ta sẵn sàng tiến hành huấn luyện và đánh giá mô hình với hai mô hình phổ biến cho các phân lớp, đó chính là KNN và Naive Bayes.

KNN Classification

- Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

- Kiểm tra độ chính xác của mô hình khi huấn luyện với các giá trị khác nhau của super-parameter `n_neighbors` và sau đó chọn giá trị tốt nhất từ chúng.

Huấn luyện mô hình



Biểu đồ cho ta thấy độ chính xác cao nhất với giá trị của 'k' là 10

Cross-validation

Cross validation là một kỹ thuật lấy mẫu để đánh giá mô hình học máy trong trường hợp dữ liệu không được dồi dào cho lắm.

Kỹ thuật này thường bao gồm các bước như sau:

1. Xáo trộn dataset một cách ngẫu nhiên.
2. Chia dataset thành k nhóm.
3. Với mỗi nhóm:

- Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình.
- Các nhóm còn lại được sử dụng để huấn luyện mô hình.
- Huấn luyện mô hình

4. Đánh giá và sau đó hủy mô hình:

- Tổng hợp hiệu quả của mô hình dựa từ các số liệu đánh giá.

Xác thực siêu tham số của mô hình và báo cáo quá trình tinh chỉnh

- Tiến hành tinh chỉnh siêu tham số n_neighbors của mô hình KNN.

- Khớp mô hình trên tập huấn luyện rồi tìm `best_score_` và lấy siêu tham số (`best_params_`) với `best_score_`.

- Kết quả knn_cv.best_score_:

```
0.617022012053604
```

- Kết quả knn_cv.best_params_:

```
{'n_neighbors': 48}
```

- Khớp mô hình sau khi đã tìm được siêu tham số tốt nhất và tiến hành dự đoán.

Đánh giá mô hình

```
array([[3109,  406],
       [ 742,   91]], dtype=int64)
```

Từ Confusion Matrix trên trái, ta có:

- TP = 3503
- FP = 12
- TN = 73
- FN = 760

Classification Report:

	precision	recall	f1-score	support
0	0.81	0.88	0.84	3515
1	0.18	0.11	0.14	833
accuracy			0.74	4348
macro avg	0.50	0.50	0.49	4348
weighted avg	0.69	0.74	0.71	4348

Độ chính xác của mô hình:

```
0.6223116733755032
```

Naive Bayes Classification

- Naive Bayes là một thuật toán học có giám sát, dựa trên định lý Bayes và được sử dụng để giải các bài toán phân loại.
- Nó chủ yếu được sử dụng trong phân loại văn bản bao gồm tập dữ liệu huấn luyện chiều cao.
- Naive Bayes Classifier là một trong những thuật toán phân lớp đơn giản và hiệu quả nhất giúp xây dựng các mô hình máy học nhanh có thể đưa ra dự đoán nhanh.
- Nó là một bộ phân loại xác suất, có nghĩa là nó dự đoán trên cơ sở xác suất của một đối tượng.
- Một số ví dụ phổ biến về thuật toán Naive Bayes là lọc thư rác, phân tích tình cảm và phân loại bài viết.

Naive Bayes Classification

- `GaussianNB()` không cần tham số nên ta tiến hành huấn luyện mô hình mà không cần kiểm tra độ chính xác đối với từng `n_neighbors` khác nhau như ở mô hình KNN.
- Giá trị độ chính xác:

```
0.7012419503219871
```

Xác thực siêu tham số của mô hình và báo cáo quá trình tinh chỉnh

- Ta tiến hành xác thực kỹ lưỡng siêu tham số của mô hình cũng bằng kỹ thuật cross-validation.

- Khớp mô hình trên tập huấn luyện rồi tìm `best_score_` và lấy siêu tham số (`best_params_`) với `best_score_`.

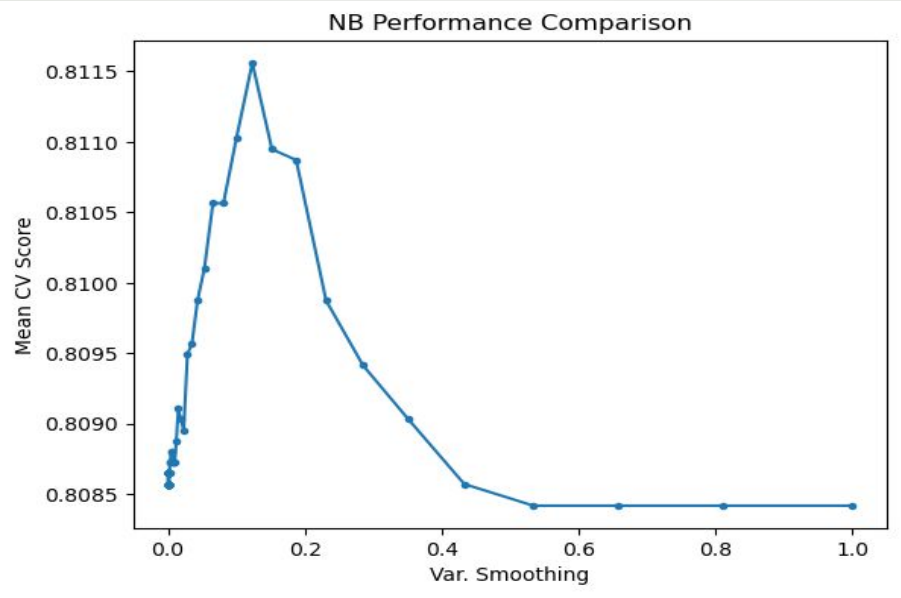
- Giá trị gs_NB.best_params_:

```
{'var_smoothing': 0.12328467394420659}
```

- Giá trị gs_NB.best_score_:

```
0.8115614459743662
```

Xác thực siêu tham số của mô hình và báo cáo quá trình tinh chỉnh



Độ chính xác của mô hình : `0.8116375344986201`

So sánh kết quả mô hình học máy

	KNN	Naive Bayes
Ưu điểm	- Hoạt động tốt trong các trường hợp dữ liệu độc lập có điều kiện hoặc có zero probability problem. - Không yêu cầu quá trình training.	- Hoạt động tốt với các bộ dữ liệu lớn, nhiều đặc trưng. - Không bị ảnh hưởng bởi curse of dimensionality.
Khuyết điểm	- Hoạt động không tốt nếu bộ dữ liệu lớn, nhiều đặc trưng hoặc bị ảnh hưởng bởi curse of dimensionality.	- Chỉ hoạt động nếu **decision boundary** là tuyến tính, elliptic hoặc parabolic. - Cần quá trình training.

So sánh kết quả mô hình học máy

	KNN	Naive Bayes
Độ chính xác	0.6223116733755032	0.8116375344986201
Lý do	Bộ dữ liệu lớn và có nhiều đặc trưng.	

06 THE END/