PROJECT 3

BÁO CÁO ĐỒ ÁN VỀ PHÂN TÍCH DỮ LIỆU WEB

Giảng viên hướng dẫn: Lê Ngọc Thành

Danh sách thành viên:

- 20424030 - Đặng Trung Hiếu
- 20424051 - Nguyễn Thành Long

Mục lục

I. Môi trường, thư viện

A. Môi trường thực hiện code:

- Windows / Ubuntu
- Ngôn ngữ lập trình Python 3
- Jupyter Notebook / Visual Studio Code

B. Thư viện:

- Pandas : Làm việc với dạng bảng
- Numpy : Làm việc với dạng ma trận
- Matplotlib : Trực quan dữ liệu
- Sklearn: Sử dụng hàm máy học
- Một số hàm hỗ trợ viết trong file source

II. Công việc từng thành viên

Nằm trong file "**ChiTietPhanCong.docx**"

III. Mức độ hoàn thành tổng thể và đánh giá của nhóm

STT	Tiêu chí	Tỉ lệ	Đánh giá
1	Đặt ra các vấn đề cần giải quyết	10%	10%
2	Mô tả dữ liệu liên quan	5%	5%
3	Chọn lựa, giải thích tính phù hợp của các mô hình học máy trên dữ liệu và bài toán nêu ra	10%	10%
4	Thực hiện huấn luyện và mô tả chi tiết các thuật toán được triển khai để huấn luyện	25%	20%
5	Mô tả cách phân chia dữ liệu huấn luyện và kiểm thử, lý giải cách phân chia và chứng tỏ kết quả không quá phụ thuộc vào cách phân chia đó.	10%	8%
6	Giải thích các độ đo để đánh giá mô hình	5%	5%

7	Phân tích và trực quan hóa kết quả thu được, lý giải các điểm quan trọng	25%	20%
8	Kết luận về vấn đề nêu ra ban đầu	10%	10%
	Tổng cộng	100%	88%

IV. Tiền xử lý

a) Bổ sung thêm field “numOfServices”:

Mục đích để tái sử dụng, ý nghĩa của field là số lượng dịch vụ mà quán hỗ trợ.

b) Xem qua dữ liệu cùng với các thống kê cơ bản:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5801 entries, 0 to 5800
Data columns (total 74 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   website_id                              5801 non-null   object
 1   website                                 5801 non-null   object
 2   url                                     5801 non-null   object
 3   full_name                               5801 non-null   object
 4   phone                                   2435 non-null   object
 5   district                                5801 non-null   object
 6   rate                                    5801 non-null   float64
 7   rate_count                              5801 non-null   float64
 8   favorite                                3927 non-null   float64
 9   active_time                             5801 non-null   object
 10  price_from                              5801 non-null   float64
 11  price_to                                5801 non-null   float64
 12  other_service                           4901 non-null   object
 13  monday_open                             5783 non-null   float64
 14  monday_close                            5783 non-null   float64
 15  tuesday_open                            5787 non-null   float64
 16  tuesday_close                           5787 non-null   float64
 17  wednesday_open                          5787 non-null   float64
 18  wednesday_close                         5787 non-null   float64
 19  thursday_open                           5785 non-null   float64
 20  thursday_close                          5785 non-null   float64
 21  friday_open                             5786 non-null   float64
 22  friday_close                            5786 non-null   float64
 23  saturday_open                           5783 non-null   float64
 24  saturday_close                          5783 non-null   float64
 25  sunday_open                             5761 non-null   float64
 26  sunday_close                            5761 non-null   float64
 27  s_cho mua về                            5801 non-null   bool
 28  s_có bàn ngoài trời                     5801 non-null   bool
 29  s_có chiếu bóng đá                      5801 non-null   bool
 30  s có chỗ chơi cho trẻ em                5801 non-null   bool
```

```
 30  s_có chỗ chơi cho trẻ em                5801 non-null   bool
 31  s_có chỗ đậu ôtô                        5801 non-null   bool
 32  s_có giao hàng                          5801 non-null   bool
 33  s_có giao hàng
:
> 100k & quận 1       5801 non-null   bool
 34  s_có hồ bơi                             5801 non-null   bool
 35  s_có hỗ trợ hội thảo                    5801 non-null   bool
 36  s_có hỗ trợ người khuyết tật            5801 non-null   bool
 37  s_có karaoke                            5801 non-null   bool
 38  s_có khu vực hút thuốc                  5801 non-null   bool
 39  s_có lò sưởi                            5801 non-null   bool
 40  s_có máy lạnh & điều hòa                5801 non-null   bool
 41  s_có nhạc sống                          5801 non-null   bool
 42  s_có phòng riêng                        5801 non-null   bool
 43  s_có thẻ thành viên                     5801 non-null   bool
 44  s_có thẻ thành viên
:
có bán vouchers  5801 non-null   bool
 45  s_có wifi                               5801 non-null   bool
 46  s_có xuất hóa đơn đỏ                    5801 non-null   bool
 47  s_có xuất hóa đơn đỏ
:
thẻ giảm giá    5801 non-null   bool
 48  s_dịch vụ tại chỗ                       5801 non-null   bool
 49  s_giao hàng                             5801 non-null   bool
 50  s_giao hàng gián tiếp                   5801 non-null   bool
 51  s_giữ xe máy miễn phí                   5801 non-null   bool
 52  s_giữ xe máy miễn phí
:
2 tiếng        5801 non-null   bool
 53  s_không có đồ ăn mang đi                5801 non-null   bool
 54  s_không giao hàng                       5801 non-null   bool
 55  s_không ăn tại chỗ                      5801 non-null   bool
 56  s_mua hàng ngay trên xe                 5801 non-null   bool
 57  s_mua sắm tại cửa hàng                  5801 non-null   bool
 58  s_nhà hàng                              5801 non-null   bool
 59  s_nhận hàng ở lề đường                  5801 non-null   bool
 60  s_nhận tại cửa hàng                     5801 non-null   bool
 61  s_nên đặt trước                         5801 non-null   bool
 62  s_nên đặt trước
:
bắt buộc             5801 non-null   bool
 63  s_phòng gym                             5801 non-null   bool
 64  s_spa & massage                         5801 non-null   bool
```

```
 65  s_sân tennis                                5801 non-null   bool
 66  s_tip cho nhân viên                         5801 non-null   bool
 67  s_trả bằng thẻ                              5801 non-null   bool
 68  s_trả bằng thẻ
:
visa/master            5801 non-null   bool
 69  s_ăn tại chỗ                                5801 non-null   bool
 70  s_đặt lịch hẹn trực tuyến                   5801 non-null   bool
 71  s_đồ ăn mang đi                             5801 non-null   bool
 72  s_đồ ăn tận phòng                           5801 non-null   bool
 73  numOfServices                               4901 non-null   object
dtypes: bool(46), float64(19), object(9)
memory usage: 1.5+ MB
```

Thống kê:

	rate	rate_count	favorite	price_from	price_to	monday_open	monday_close	tuesday_open	tuesday_close	wednesday_open	v
count	5801.000000	5801.000000	3927.000000	5.801000e+03	5.801000e+03	5783.000000	5783.000000	5787.000000	5787.000000	5787.000000	
mean	5.819203	62.664388	41.939649	5.825490e+04	4.733293e+05	8.453139	20.179587	8.464677	20.185260	8.474662	
std	3.676943	153.641717	163.079957	1.455547e+05	2.248216e+06	4.418259	5.221462	4.414098	5.207387	4.411955	
min	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	6.500000	20.000000	7.000000	20.000000	7.000000	
50%	7.586000	3.000000	2.000000	2.000000e+04	1.000000e+05	9.000000	22.000000	9.000000	22.000000	9.000000	
75%	8.600000	32.000000	17.000000	5.500000e+04	3.000000e+05	10.000000	23.000000	10.000000	23.000000	10.000000	
max	10.000000	1000.000000	4746.000000	4.000000e+06	1.000000e+08	22.000000	23.983333	22.000000	23.983333	22.000000	

wednesday_close	thursday_open	thursday_close	friday_open	friday_close	saturday_open	saturday_close	sunday_open	sunday_close
5787.000000	5785.000000	5785.000000	5786.000000	5786.000000	5783.000000	5783.000000	5761.000000	5761.000000
20.175917	8.474898	20.177148	8.468839	20.172834	8.438723	20.182227	8.405164	20.201892
5.215470	4.410622	5.215819	4.412386	5.225363	4.436966	5.233537	4.475502	5.247239
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
20.000000	7.000000	20.000000	7.000000	20.000000	6.500000	20.000000	6.500000	20.000000
22.000000	9.000000	22.000000	9.000000	22.000000	9.000000	22.000000	9.000000	22.000000
23.000000	10.000000	23.000000	10.000000	23.000000	10.000000	23.000000	10.000000	23.000000
23.983333	22.000000	23.983333	22.000000	23.983333	22.000000	23.983333	22.000000	23.983333

V. Bài toán và cách giải quyết

a) Long section

1) **Bài toán số 1**: Giả sử chúng ta cần tìm hiểu xem liệu rằng những quán ăn được mọi người đánh giá nhiều thì số lượng yêu thích của quán sẽ ra sao, liệu có theo 1 giả định nào đó không?

Với giả định trên ta hiểu được rằng cần tìm ra mối liên hệ giữa sự yêu thích của mọi người đối với quán ăn và số lượng người đã đánh giá cho quán ăn

đó. Sau khi trực quan dữ liệu ở project 2 thì chúng ta cũng đã phần nào hiểu được chúng có một quan hệ tương quan.

Các dữ liệu liên quan cho bài học máy này là: **rate_count** và **favorite**, trong đó output của chúng ta là **favorite.**

Qua bảng tóm tắt dữ liệu ở bước tiền xử lý ta nhận ra **rate_count** và **favorite** đều là những field có giá trị là kiểu số (định lượng) và liên tục nên phương pháp mô hình phù hợp nhất cho bài toán này là “linear regression”.

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model.

Bước chuẩn bị dữ liệu: một bài toán máy học ta cần chia ra làm hai phần trong đó 1 phần để training và một phần để kiểm tra testing. Tỉ lệ 2 phần này em sử dụng theo 80:20, với cách chia này khá phù hợp với dữ liệu có data > 1000 dòng.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số nếu có
- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

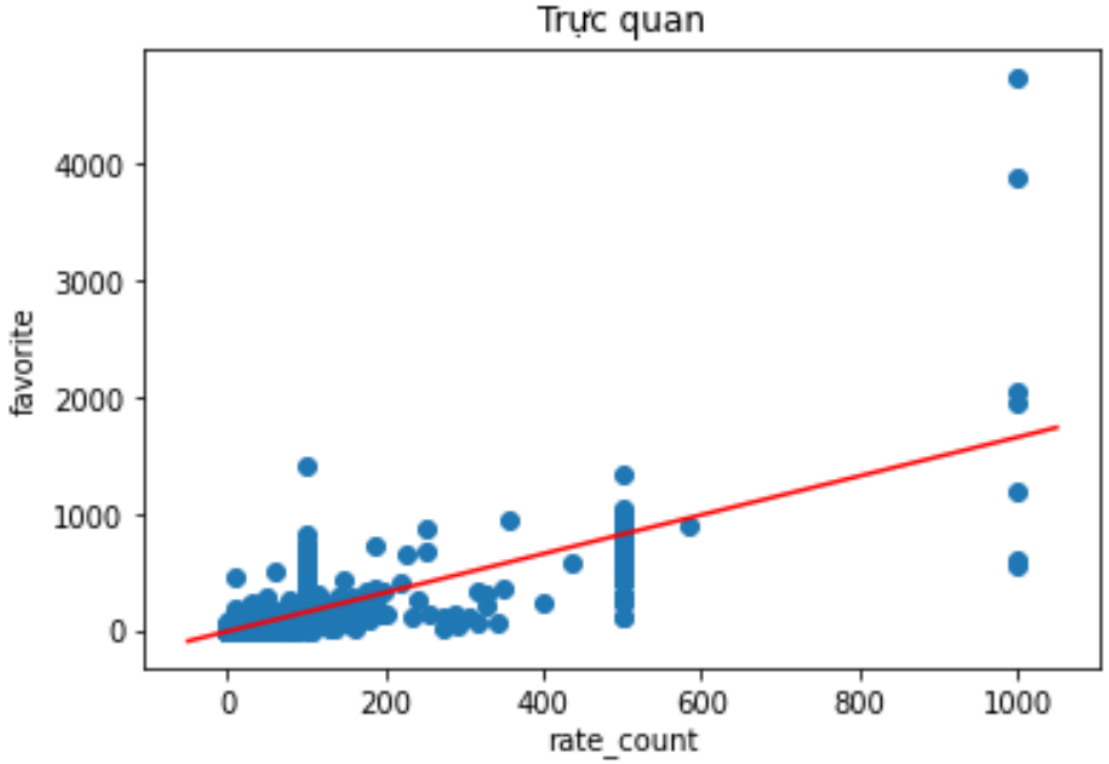
Bước kiểm tra:

- Xem qua tham số mô hình
- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

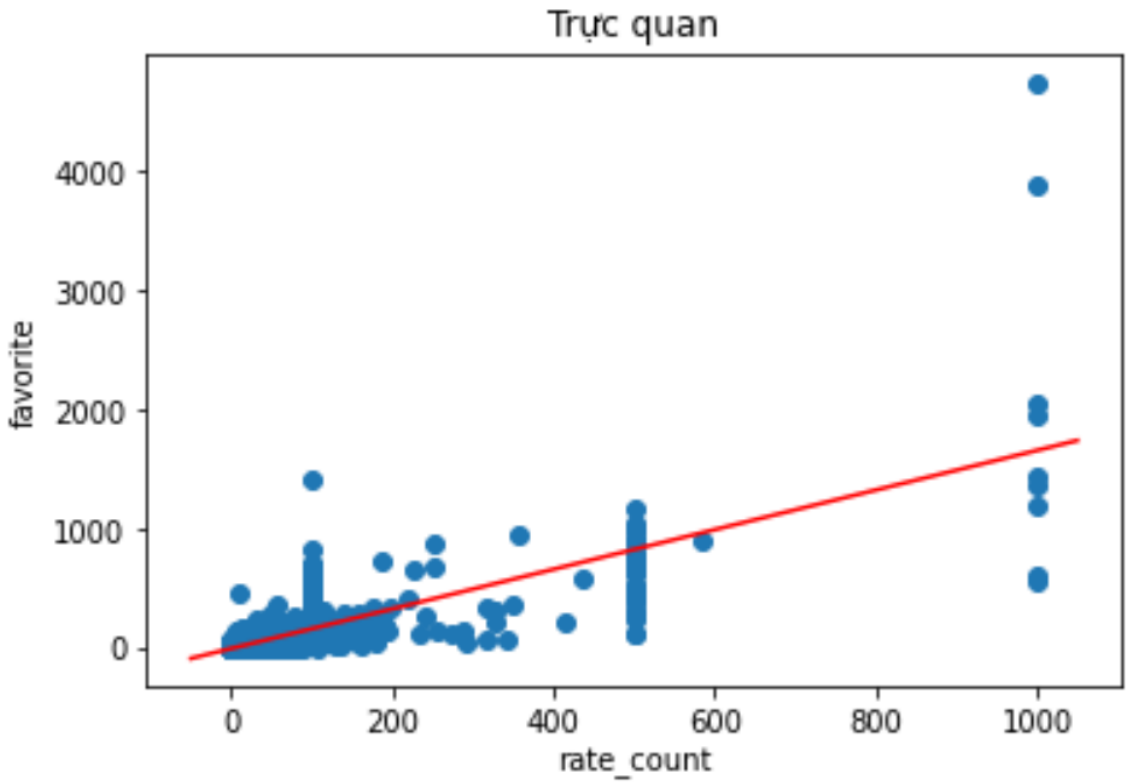
```
Tham số mô hình:
[1.66050653]
3.351534588749466

Độ lỗi trên tập train:
Độ lỗi MSE: 10842.265197649605
Độ lỗi R^2: 0.6108484194870373
Độ lỗi trên tập test:
Độ lỗi MSE: 4620.229083019934
Độ lỗi R^2: 0.7851198506885622
```



Chúng ta sẽ kiểm thử lại để xem việc chia 80:20 có ảnh hưởng nhiều đến kết quả hay không:

```
Độ lỗi trên tập train:
Độ lỗi MSE: 10717.638457176581
Độ lỗi R^2: 0.6177538186008218
Độ lỗi trên tập test:
Độ lỗi MSE: 5118.26037033007
Độ lỗi R^2: 0.7538502765821988
```



Nhận xét:

- Sau khi trải qua 2 model với cách chia 80:20 thì kết quả không thay đổi quá nhiều (qua hình ảnh trực quan của hàm tuyến tính)

- Độ lỗi của R bình phương cho thấy có giá trị có thay đổi nhỏ, không lớn lắm

Điều đó chứng tỏ rằng kết quả không phụ thuộc quá nhiều vào cách chia bộ dữ liệu training set và test set.

Kết luận:

- Có thể thấy độ lỗi MSE của cả 2 mô hình có giá trị lớn đó cũng là do 1 phần outlier gây ra , việc thu thập dữ liệu không thể tránh khỏi

- Dựa vào hàm tuyến tính đã xác định được qua model thì ta có thể thấy rằng favorite tương quan với rate_count theo quan hệ đồng biến.

Vậy mô hình đã giải quyết được bài toán mà chúng ta đã đề ra ban đầu.

2) **Bài toán số 2**: Giả sử chúng ta cần tìm hiểu xem liệu rằng những quán ăn có hỗ trợ dịch vụ: "nên đặt trước" thì giá cả sẽ ra sao, liệu có theo 1 giả định nào đó không?

Với giả định trên ta hiểu được rằng cần tìm ra mối liên hệ giữa việc quán ăn có hỗ trợ dịch vụ “nên đặt trước” (hay dễ hiểu hơn là khách hàng nên đặt món trước khi tới quán ăn) và giá trung bình của quán đưa ra, giá này bao gồm mức giá từ và mức giá đến thể hiện khoảng giá mà các món ăn có trong cửa hàng. Vậy thì chúng có quan hệ gì thì phải thử.

Các dữ liệu liên quan cho bài học máy này là: **price_from, price_to** và **s_nên đặt trước**, trong đó output của chúng ta là **s_nên đặt trước.**

Qua bảng tóm tắt dữ liệu ở bước tiền xử lý ta nhận ra **price_from, price_to** đều là những field có giá trị là kiểu số (định lượng) và **s_nên đặt trước** là kiểu giá trị định tính phân lớp (true/false) nên phương pháp mô hình phù hợp nhất cho bài toán này là “logistic regression”.

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model. Ở đây chúng ta phải xác định được record nào của website shopeefood bởi vì chúng không có giá trị other_service, đồng nghĩa sẽ gây nhiễu rất nhiều tới thuật toán và quá trình học.

Bước chuẩn bị dữ liệu: một bài toán máy học ta cần chia ra làm hai phần trong đó 1 phần để training và một phần để kiểm tra testing. Tỉ lệ 2 phần này em sử dụng theo 80:20, với cách chia này khá phù hợp với dữ liệu có data > 1000 dòng.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số nếu có

- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

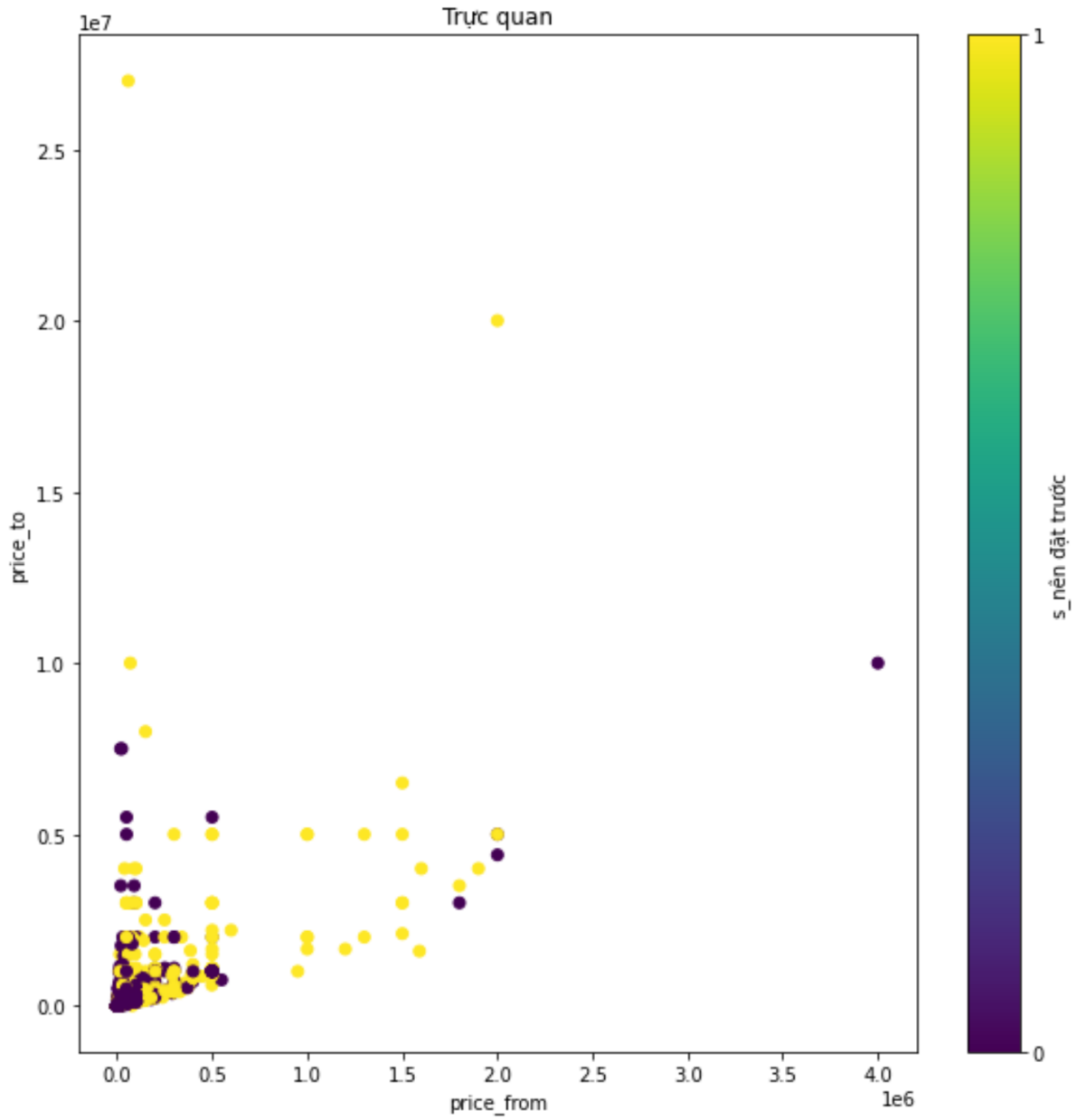
Bước kiểm tra:

- Xem qua tham số mô hình
- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Tham số mô hình:
[[-1.09441743e-06  4.98709446e-09]]
[-1.5333935e-10]

Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.15510204081632653
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.13761467889908258
```



Chúng ta sẽ kiểm thử lại để xem việc chia 80:20 có ảnh hưởng nhiều đến kết quả hay không:

```
Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.15637755102040815
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.13557594291539246
```

Nhận xét:

- Sau khi trải qua 2 model với cách chia 80:20 thì kết quả không thay đổi quá nhiều

- Độ lỗi Phần trăm phân lớp sai cho thấy có giá trị có thay đổi nhỏ, không lớn lắm

Điều đó chứng tỏ rằng kết quả không phụ thuộc quá nhiều vào cách chia bộ dữ liệu training set và test set.

Kết luận:

- Có thể thấy độ lỗi phân lớp cả 2 mô hình có giá trị khá nhỏ đó cũng là một dấu hiệu tốt cho thấy việc học máy hiệu quả với phương pháp logistic regression.

- Dựa vào tham số đã xác định được qua model thì ta có thể thấy rằng với **price_from** và **price_to** có thể xác định được phân lớp dịch vụ **nên đặt trước** (True/False).

Vậy mô hình đã giải quyết được bài toán mà chúng ta đã đề ra ban đầu.

*** Mở rộng bài toán qua phương pháp học máy Neural Network**

Việc áp dụng neural network vào bài toán mang cho ta khả năng tối ưu tốt hơn so với việc đi tìm các features trước khi cho vào học máy, nhưng bài toán này ta sẽ kiểm tra xem liệu rằng neural network có tối ưu cho cả trường hợp tìm feature trước hay không.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số: 1 tầng ẩn, 3 noron, hàm kích hoạt “tanh”, thuật toán cực tiểu hóa: “LBFGS”, lặp tối đa 1000 lần
- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

Bước kiểm tra:

- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.1530612244897592
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.1437308868501529
```

Thử nghiệm với trường hợp tăng layer và số noron lên. (tham số: 2 tầng ẩn, lần lượt số noron là 30 - 15 noron, hàm kích hoạt “logistic”, thuật toán cực tiểu hóa: “LBFGS”, lặp tối đa 1000 lần)

Kết quả:

```
Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.1507653061224898
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.15392456676860347
```

Nhận xét:

- Qua 2 mô hình neural network thì kết quả về độ lỗi phân lớp khá giống nhau.
- Tùy vào hidden_layer_sizes thì thời gian chạy huấn luyện sẽ khác nhau, tuy nhiên độ lỗi cũng không được cải thiện nhiều.

Kết luận chung khi sử dụng neural network với bài toán hiện tại:, neural network cũng có thể tối ưu cho cả trường hợp tìm feature trước với kết quả cũng rất khả quan.

3) **Bài toán số 3**: Giả sử chúng ta cần tìm hiểu xem liệu rằng những quán ăn có hỗ trợ nhiều dịch vụ, hay số lượng yêu thích, hay số lượng đánh giá thì ảnh hưởng tới RATE quán sẽ ra sao, liệu có theo 1 giả định nào đó không?

Với giả định trên ta hiểu được rằng cần tìm ra mối liên hệ giữa số sao đánh giá về chất lượng dịch vụ của quán ăn với số lượng người đã đánh giá cho quán ăn đó, hay là số lượng khách hàng yêu thích quán ăn đó, và cũng có thể là số lượng dịch vụ mà quán ăn có / cung cấp cho khách hàng. Sau khi trực quan dữ liệu ở project 2 của một số trường, ta đã biết được rate,rate_count,favorite chúng có một quan hệ tương quan nhưng số lượng dịch vụ thì chưa biết. Vậy thì chúng có quan hệ gì thì phải thử.

Các dữ liệu liên quan cho bài học máy này là: **numOfServices, rate_count, favorite** và **rate**, trong đó output của chúng ta là **rate.**

Để đơn giản ta phân lớp lại rate từ khoảng giá trị liên tục từ 0 tới 10 ta sẽ chia làm 3 loại:

- **BAD**: từ 0 đến 3 sao
- **NORMAL**: từ 4 đến 7 sao
- **GOOD**: từ 8 đến 10 sao

Kết hợp với bảng tóm tắt dữ liệu ở bước tiền xử lý ta nhận ra **numOfServices, rate_count, favorite** đều là những field có giá trị là kiểu số (định lượng) và phân lớp lại rate có số lớp > 2 nên phương pháp mô hình phù hợp nhất cho bài toán này là “softmax regression”.

Ta cũng sẽ xác định ra các field cho ra kết quả tốt nhất, nên ta sẽ đi từng field trước, để tìm hiểu mối quan hệ

Bước tiền xử lý chung: chúng ta xây dựng thêm 1 cột f_rate, mang ý nghĩa phân lớp lại giá trị rate theo như đã mô tả ở trên.

Bước chuẩn bị dữ liệu: một bài toán máy học ta cần chia ra làm hai phần trong đó 1 phần để training và một phần để kiểm tra testing. Tỉ lệ 2 phần này em sử dụng theo 80:20, với cách chia này khá phù hợp với dữ liệu có data > 1000 dòng.

*) Field : **rate_count**

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model. Ở đây chúng ta phải loại bỏ các record mà **rate_count** có giá trị NAN hay không có giá trị.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số (đa lớp, thuật toán cực tiểu hóa: “lbfgs”, số lần lặp tối đa 1e4)
- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

Bước kiểm tra:

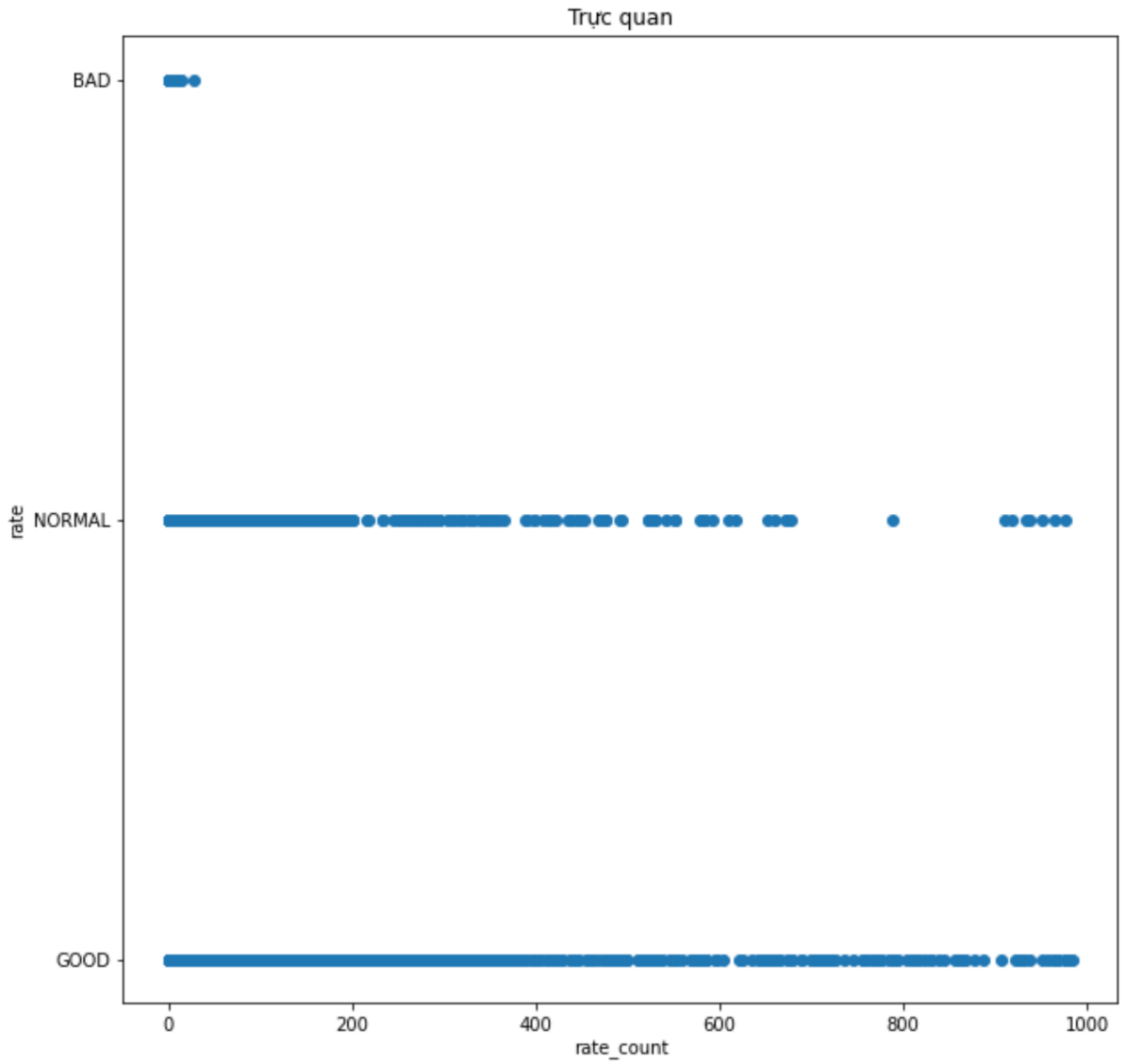
- Xem qua tham số mô hình

- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Tham số mô hình:
[[-1.00002967]
 [ 0.50147722]
 [ 0.49855245]]
[ 0.93951333 -0.53978918 -0.39972415]

Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.40714285714285714
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.417940876656473
```



Nhận xét:

- Độ lỗi của mô hình khá cao, nhưng trong khoảng chấp nhận được, ít ra vẫn tốt hơn là baseline
- Việc chia training và test set không ảnh hưởng quá nhiều tới kết quả (sau khi chạy lại thì kết quả độ lỗi không thay đổi nhiều)

*) Field : **favorite**

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model. Ở đây chúng ta phải loại bỏ các record mà **favorite** có giá trị NAN hay không có giá trị.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số (đa lớp, thuật toán cực tiểu hóa: “lbfgs”, số lần lặp tối đa 1e4)

- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

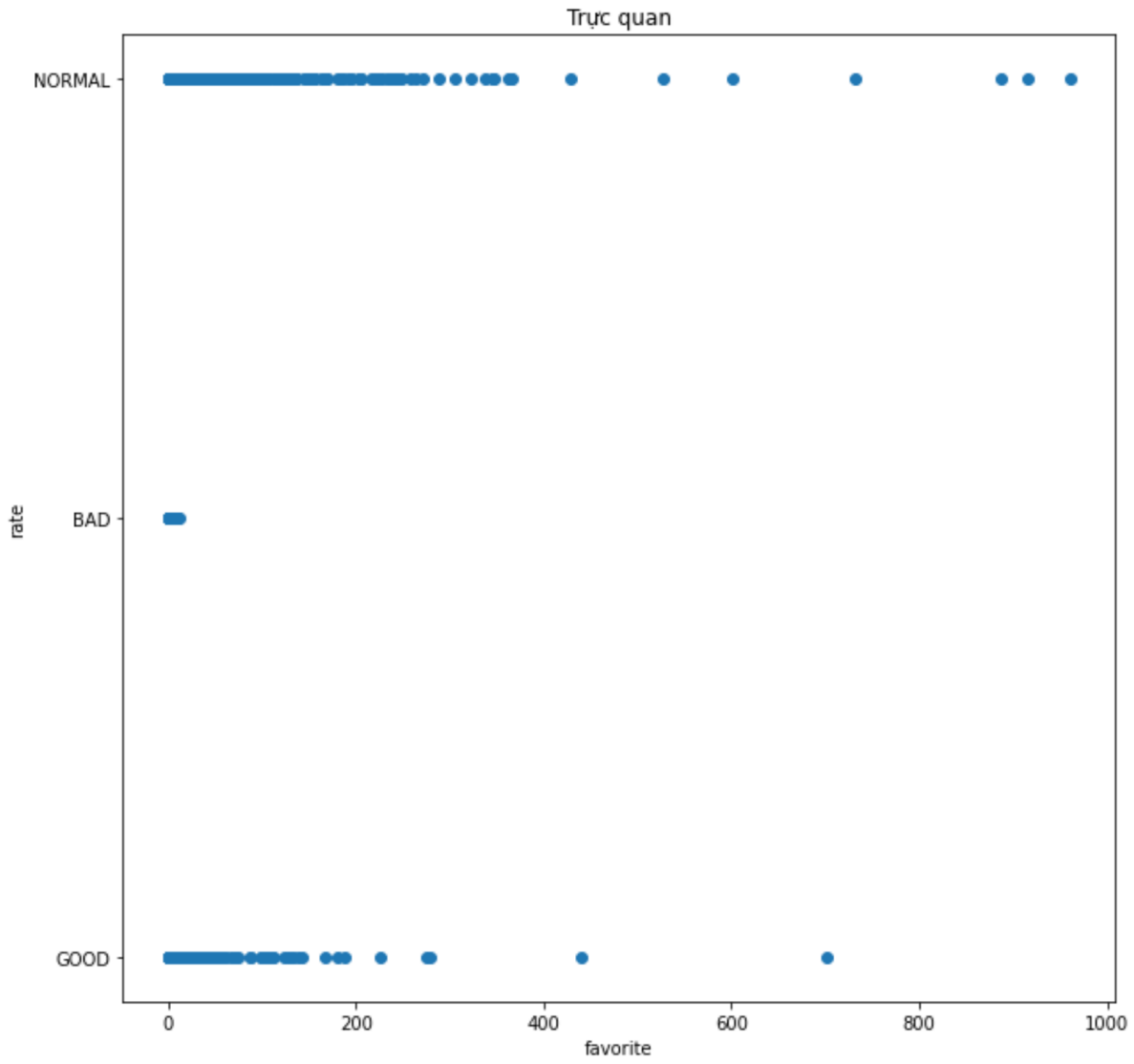
Bước kiểm tra:

- Xem qua tham số mô hình
- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Tham số mô hình:
[[-0.42919616]
 [ 0.21003039]
 [ 0.21916577]]
[ 0.97849641 -0.62528288 -0.35321353]

Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.37835605121850474
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.3811881188118812
```



Nhận xét:

- Độ lỗi của mô hình không cao lắm, tuy nhiên vẫn rất khó xác định được phân lớp của output
- Việc chia training và test set không ảnh hưởng quá nhiều tới kết quả (sau khi chạy lại thì kết quả độ lỗi không thay đổi nhiều)

*) Field : **numOfServices**

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model. Ở đây chúng ta phải loại bỏ các record mà **numOfServices** có giá trị NAN hay không có giá trị.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số (đa lớp, thuật toán cực tiểu hóa: “lbfgs”, số lần lặp tối đa 1e4)

- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

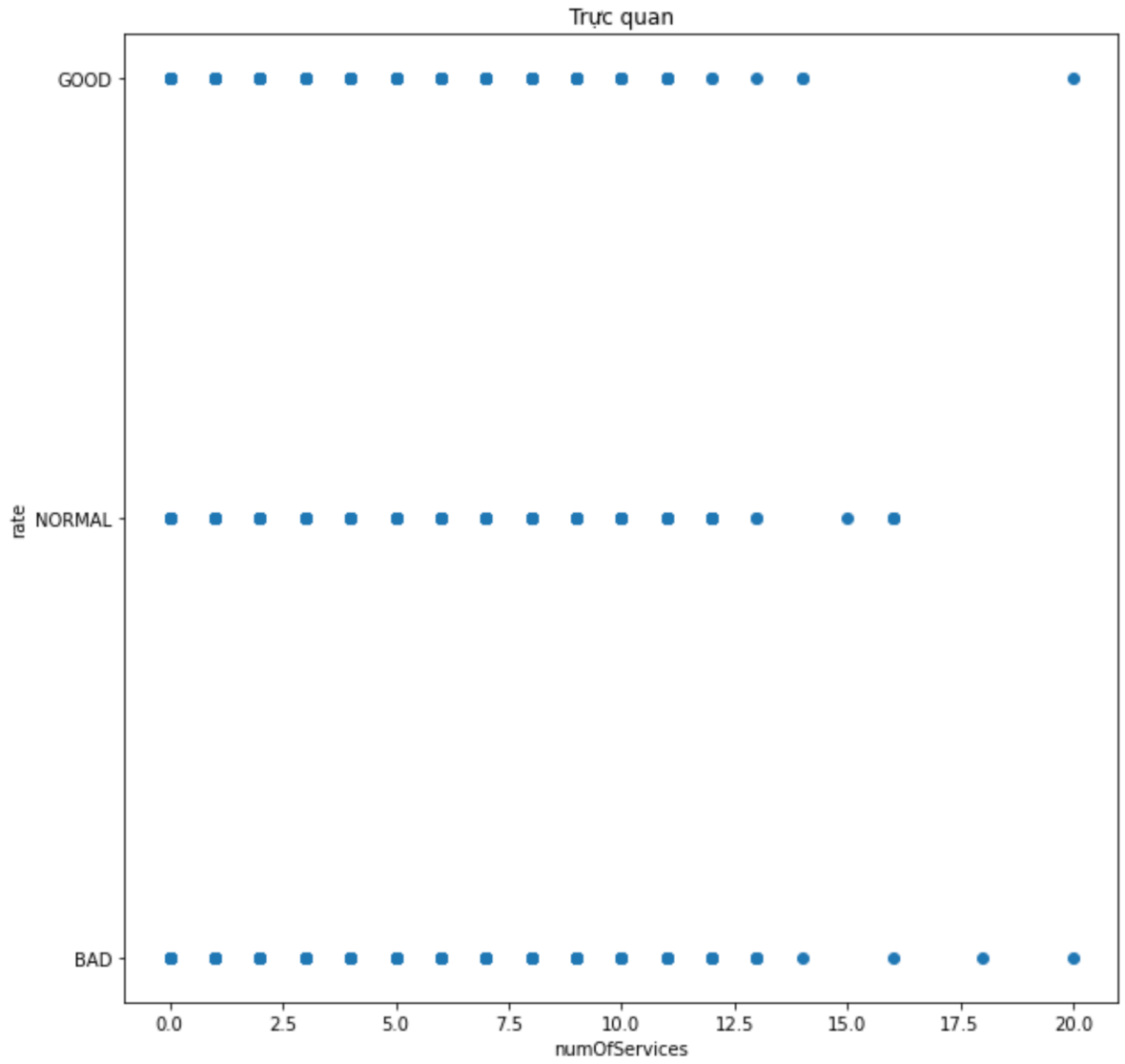
Bước kiểm tra:

- Xem qua tham số mô hình
- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Tham số mô hình:
[[ 0.03178305]
 [-0.02795663]
 [-0.00382642]]
[-0.39826035  0.27248163  0.12577872]

Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.5880102040816326
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.5749235474006116
```



Nhận xét:

- Độ lỗi của mô hình cao hơn cả baseline, không khả quan khi áp dụng.
- Việc chia training và test set không ảnh hưởng quá nhiều tới kết quả (sau khi chạy lại thì kết quả độ lỗi không thay đổi nhiều)

Với từng field riêng lẻ, ta vẫn chưa có cách nào phân tách tuyến tính output ra thành 3 lớp riêng biệt nên phải xem xét đến việc kết hợp 2 fields có chất lượng tốt lại với nhau để kiểm tra kết quả mới.

*) Field : **rate_count, favorite**

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model. Ở đây chúng ta phải loại bỏ các record mà **rate_count** hoặc **favorite** có giá trị NAN hay không có giá trị.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số (đa lớp, thuật toán cực tiểu hóa: “lbfgs”, số lần lặp tối đa 1e4)
- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

Bước kiểm tra:

- Xem qua tham số mô hình
- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Tham số mô hình:
[[-1.36986532 -0.13450414]
 [ 0.67726014  0.06722196]
 [ 0.69260517  0.06728218]]
[ 1.74316217 -0.9822492  -0.76091297]

Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.31268071045022716
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.3151815181518151 7
```

Nhận xét:

- Độ lỗi của mô hình có kết quả xấp xỉ với cả mô hình chỉ sử dụng 1 field, vẫn chưa thực sự tốt.
- Việc chia training và test set không ảnh hưởng quá nhiều tới kết quả (sau khi chạy lại thì kết quả độ lỗi không thay đổi nhiều)

Với kết quả chưa thực sự tốt, ta thử kết hợp cả 3 fields lại với nhau để kiểm tra kết quả mới có cải thiện được gì không?

*) Field : **rate_count, favorite, numOfServices**

Bước tiền xử lý: để chắc chắn, chúng ta cần phải làm sạch dữ liệu NAN trước khi đưa vào model. Ở đây chúng ta phải loại bỏ các record mà **rate_count** hoặc **favorite** hoặc **numOfServices** có giá trị NAN hay không có giá trị.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số (đa lớp, thuật toán cực tiểu hóa: “lbfgs”, số lần lặp tối đa 1e4)
- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

Bước kiểm tra:

- Xem qua tham số mô hình
- Trực quan mô hình nếu có thể
- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Tham số mô hình:
[[-1.21933151 -0.13057601  0.0764095 ]
 [ 0.60187481  0.06468    -0.01356279]
 [ 0.6174567   0.065896   -0.06284671]]
[ 1.49068526 -0.91545231 -0.57523295]

Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.3106154481619165 5
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.297029702970297
```

Nhận xét:

- Độ lỗi của mô hình 3 fields có kết quả tốt hơn một chút với mô hình chỉ sử dụng 1 field và 2 field
- Việc chia training và test set không ảnh hưởng quá nhiều tới kết quả (sau khi chạy lại thì kết quả độ lỗi không thay đổi nhiều)

*** Mở rộng bài toán qua phương pháp học máy Neural Network**

Việc áp dụng neural network vào bài toán mang cho ta khả năng tối ưu tốt hơn so với việc đi tìm các features trước khi cho vào học máy, nhưng bài toán

này ta sẽ kiểm tra xem liệu rằng neural network có tối ưu cho cả trường hợp tìm feature trước hay không.

Bước huấn luyện:

- Khởi tạo mô hình thông qua hàm thư viện cung cấp kèm các tham số: 2 tầng ẩn, lần lượt số noron là 25 - 9 noron, hàm kích hoạt “tanh”, thuật toán cực tiểu hóa: “LBFGS”, lặp tối đa 1000 lần
- Chia tập dữ liệu 80:20
- Fit dữ liệu vào mô hình máy học

Bước kiểm tra:

- Kiểm tra độ lỗi trên tập training và tập testing

Kết quả:

```
Độ lỗi trên tập train:
Phần trăm phân lớp sai: 0.20693928128872366
Độ lỗi trên tập test:
Phần trăm phân lớp sai: 0.2722772277227723
```

```
C:\Users\thafnhlong\AppData\Local\Programs\Python\Python39\lib\site-packages
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
```

Ngay cả học neural với 1000 lần lặp nhưng dữ liệu vẫn chưa được fit hoàn toàn nhưng kết quả độ lỗi lại có tiềm năng lớn.

Nhận xét:

- Độ lỗi của mô hình neural network có kết quả tốt nhất, nhưng dường như với cấu hình tham số cho mô hình đã gây ra overfitting khi trên tập test thì phân lớp lại sai nhiều hơn.
- Việc chia training và test set không ảnh hưởng quá nhiều tới kết quả (sau khi chạy lại thì kết quả độ lỗi không thay đổi nhiều)

Kết luận chung:

- Khi sử dụng neural network với bài toán hiện tại:, neural network cũng có thể tối ưu cho cả trường hợp tìm feature trước với kết quả cũng rất khả quan.

- Có thể thấy độ lỗi phân lớp cả 2 phương pháp (softmax,neural) cho cả 3 fields có giá trị khá nhỏ đó cũng là một dấu hiệu tốt cho thấy việc học máy hiệu quả.
- Dựa vào tham số đã xác định được qua model thì ta có thể thấy rằng với **rate_count**, **favorite**, **numOfServices** có thể xác đinh được phân lớp đánh giá **rate** ("BAD","NORMAL","GOOD").

Vậy mô hình đã giải quyết được bài toán mà chúng ta đã đề ra ban đầu.

b) Hieu section

Hạn mức khoảng giá tiền ở các quán ăn là số liệu quan trọng trong việc mô hình hóa đặc trưng thành hệ, có ảnh hưởng lớn đến tỉ lệ và số người dùng đến quán và tham gia review trên các trang web về review ẩm thực.

Đặt ra bài toán bài toán giải quyết vấn đề giá cả ở các khu vực khác nhau là quận bằng cách thực hiện bài toán ước lượng dự đoán các giá cả tại khu vực các quán ăn và đánh giá số điểm review để xem việc giá cả tại các quán ăn có tác động đến các điểm review hay không

Các dữ liệu liên quan:

District: Bảng dữ liệu về các quận trọng HCM
Rate: Số điểm đánh giá của quán ăn
Rate_count: số lượt đánh giá của quán ăn
Price_from: Khoảng giá tiền của quán bắt đầu từ
Price_to: Khoảng giá tiền của quán tối đa.

Từ bài toán trên có thể sử dụng mô hình :

Học có giám sát : phân loại(Classification)
Neural-network: nếu nâng cao

Lý do chọn mô hình học trên: Đã có cái dữ liệu input từ bên liên quan từ các dữ liệu input layer ta có thể có chuẩn bị sẵn đầu ra output layer với 3 dạng (“có tác động mạnh”,”bình thường”,”không ảnh hưởng”). Thứ 2 đây là dữ liệu cơ bản không phức tạp nên chỉ cần áp dụng mô hình học giám sát để giải quyết bài toán

VI. Thuật toán

Chi tiết ở các phần trước.

VII. Quản lý dự án:

https://github.com/thafnhlong/HCMUS-UNIV-WebScience-MachineLearning

VIII. Tài liệu tham khảo

https://scikit-learn.org/stable/

https://v1study.com/python-tham-khao-python-ai-cach-xay-dung-mang-noron-than-kinh-va-dua-ra-du-doan.html

https://machinelearningcoban.com/2016/12/27/categories/
https://paperswithcode.com/task/semi-supervised-image-classification

SLIDE lý thuyết môn học