

- Nhập môn Khoa học dữ liệu-

Lớp: CQ2018/2 – Học kì I/ 2020-2021

Giảng viên: Trần Trung Kiên

Trợ giảng: Hoàng Xuân Trường

Báo cáo đồ án cuối kì

Dự đoán giá thuê khách sạn, homestay

Nhóm 46: Đặng Hữu Thắng – Nguyễn Hữu Huân

TP.Hồ Chí Minh, ngày 16/1/2021

NỘI DUNG



01

Giới thiệu đề án

Câu hỏi được đặt ra là gì?



02

Thu thập dữ liệu



03

Tiền xử lý dữ liệu

Khám phá dữ liệu và tiền xử lý



04

Xây dựng mô hình

Phân tích dữ liệu, mô hình hóa



05

Đánh giá và tổng kết



06

Tham khảo

Saturday
16.01.2021

Giới thiệu đồ án

Câu hỏi được đặt ra?

GIỚI THIỆU ĐỒ ÁN

▲ Câu hỏi?

Từ các tiện nghi/ dịch vụ mà bên cho thuê chỗ ở (như khách sạn, homestay), dự đoán giá thuê của chỗ ở này



GIỚI THIỆU ĐỒ ÁN



Input

Các tiện nghi/dịch vụ cung cấp



Output

Giá thuê



Lợi ích

- *Bên cho thuê*: đưa ra giá tiền thuê sao cho phù hợp với thị trường. Cũng như cần chuẩn bị những tiêu chí gì để "tối ưu" giá cho thuê lên.
- *Bên đi thuê*: dựa theo các tiêu chí mình cần để chuẩn bị ngân sách cho chỗ ở khi du lịch, lựa chọn nơi thuê phù hợp nhất theo ý mình



Nguồn gốc câu hỏi

Nhóm tự nghĩ ra dựa trên nhu cầu thực tế của bản thân

Saturday
16.01.2021

Thu thập dữ liệu

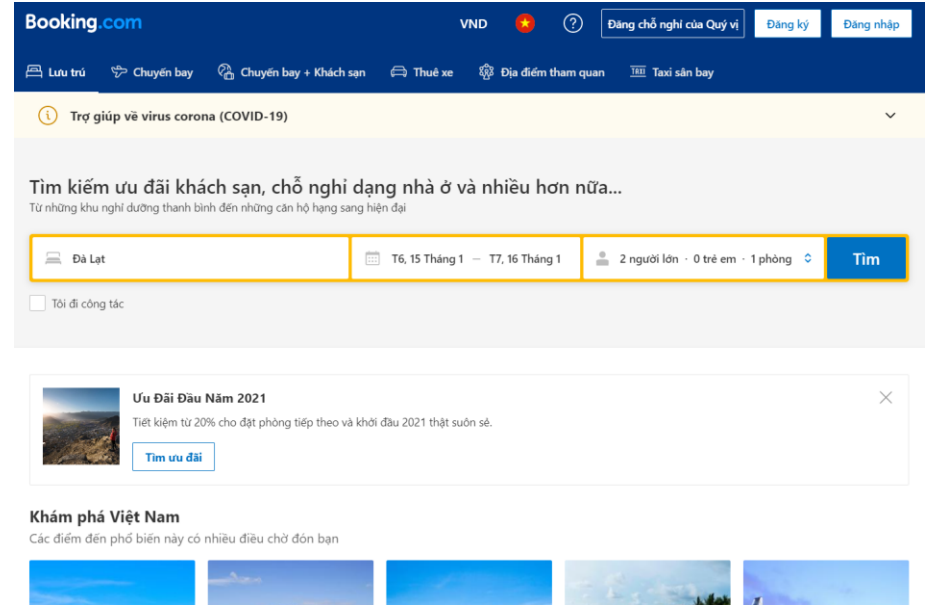
THU THẬP DỮ LIỆU

Dữ liệu được thu thập trên trang
<https://www.booking.com/>

- Dữ liệu nhóm thu thập là hợp pháp (đã check file robots.txt)

```
al = urllib.robotparser.RobotFileParser()
al.set_url('https://www.booking.com/robots.txt')
al.read()
al.can_fetch('*', 'https://www.booking.com/hotel/vn/label=gog235jc-1DCAEoggI46AdIKlgDaPQBIAEBmAEQuAEXysid=aa1ef10db21b3c99f2d46d4624c8ebf9;all_sr_blocks=408974504_130057956_253065700;highlighted_blocks=408974504_130057956_253065700;sr_order=popularity;sr_pri_blocks=408974501_130057956_253065700;sr_epoch=1610705984')
True
```









- Kiểm tra trên 1 khách sạn cụ thể



```
al.can_fetch('*', 'https://www.booking.com/hotel/vn/colline.vi.html?label=gen173nr-1DCAEoggI46AdIM1gEaPQBIAEBmAEQuAEXyAEM2AED6AEBIAIBqAIDusid=aa1ef10db21b3c99f2d46d4624c8ebf9;all_sr_blocks=408974504_130057956_253065700;highlighted_blocks=408974504_130057956_253065700;sr_order=popularity;sr_pri_blocks=408974504_130057956_253065700;sr_epoch=1610705984')
True
```

THU THẬP DỮ LIỆU

- Ở mỗi khách sạn/ homestay nhóm sẽ tiến hành lấy diện tích phòng, các tiện nghi/dịch vụ, và giá cho thuê

 29 m ²	 Nhìn ra thành phố	✓ Đồ vệ sinh cá nhân miễn phí	✓ Điện thoại	✓ Két an toàn cỡ laptop	VND 4.289.250 bao gồm thuế và phí
 Điều hòa không khí		✓ Vòi sen ✓ Áo choàng tắm	✓ Có phòng thông nhau qua cửa nối	✓ Các tầng trên đi lên bằng thang máy	
 Phòng tắm riêng		✓ Két an toàn ✓ Nhà vệ sinh	✓ Máy sấy tóc	✓ Giá treo quần áo	
 TV màn hình phẳng		✓ Ghế sofa ✓ Sàn lát gỗ	✓ Khăn tắm/Bộ khăn trải giường (có thu phí)	✓ Giấy vệ sinh	
 Hệ thống cách âm	 Minibar	✓ Khăn tắm	✓ Ấm đun nước điện	✓ Nắp che ổ cắm điện an toàn	
 WiFi miễn phí		✓ Ổ điện gần giường	✓ Truyền hình cáp		
		✓ Không gây dị ứng	✓ Dịch vụ báo thức		
		✓ Bàn làm việc ✓ TV ✓ Dép			

- Nội dung sơ lược của dữ liệu (dữ liệu thô chưa được tiền xử lí): 14316 dòng và 113 cột, trong đó
 - 1 cột là địa điểm
 - 1 cột là diện tích phòng
 - 1 cột là giá thuê phòng
 - 110 cột còn lại là tiện nghi/dịch vụ của phòng đó

Saturday
16.01.2021

Tiền xử lý dữ liệu

TIỀN XỬ LÝ DỮ LIỆU

```
df = pd.read_csv('full_data.csv')
df
```

	City	Diện tích	Đồ vệ sinh cá nhân miễn phí	Vòi sen	Áo choàng tắm	Kết an toàn	Nhà vệ sinh	Khăn tắm	Bàn làm việc	Khu vực tiếp khách	...	Đài radio	Khu vực ăn uống ngoài trời	Phòng xông hơi	Toilet chung	Bàn ghế ngoài trời	Bể sục	Hồ bơi riêng	Sân trong	Hướng nhìn sân trong	price
0	TP.Ho Chi Minh	29 m²	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	2.405.000
1	TP.Ho Chi Minh	29 m²	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	2.405.000
2	TP.Ho Chi Minh	56 m²	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	3.055.000
3	TP.Ho Chi Minh	61 m²	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	3.380.000
4	TP.Ho Chi Minh	18 m²	1	1	1	1	1	1	1	0	...	0	0	0	0	0	0	0	0	0	846.000
...
14311	Da Lat	34 m²	1	0	0	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	522.340
14312	Da Lat	36 m²	1	0	0	1	1	1	1	0	...	0	0	0	0	0	0	0	0	0	598.682
14313	Da Lat	TV màn hình phẳng	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	900.000
14314	Da Lat	30 m²	1	0	0	0	1	1	0	1	...	1	0	0	1	0	0	1	0	0	212.511
14315	Da Lat	55 m²	1	0	0	0	1	1	0	1	...	1	0	0	1	0	0	1	0	0	425.023

14316 rows x 113 columns

TIỀN XỬ LÝ DỮ LIỆU

Vấn đề của dữ liệu:

- Có quá nhiều cột
- Một số cột không được định dạng đúng kiểu dữ liệu
- Có nhiều thuộc tính (cột) mang ý nghĩa như nhau nhưng khác tên

Tiến hành xử lý thô trước để tách tập sớm:

- Cột "Diện tích": xóa các dòng không chứa đúng kiểu dữ liệu diện tích (xx m2), sau đó tiến hành xóa đuôi m2 và chuyển về dạng số
- Cột "price": xóa các dấu ".", sau đó chuyển về dạng số
- Xóa các dòng bị lặp và các cột không cần thiết ("Ổ cắm cho iPod", "Đầu đĩa CD", ... đã ghi chú trong file notebook)

TIỀN XỬ LÝ DỮ LIỆU

Tiến hành tách dữ liệu theo tỉ lệ train – validation – test: **64 – 16 – 20**

Khám phá dữ liệu trên tập train:

- Tiến hành gom các cột có điểm chung, các cột tiện nghi/dịch vụ cùng 1 loại lại bằng class `ColAdderDropper()`

VD: Các cột “Sàn trải thảm”, “Sàn lát gạch/đá cẩm thạch”, “Sàn lát gỗ” được gom lại thành 1 cột “lot_san”, với:

- “Sàn trải thảm” = 1
- “Sàn lát gạch/đá cẩm thạch” = 2
- “Sàn lát gỗ” = 3

- Có khá nhiều cột mang ý nghĩa như nhau -> ta sẽ tiến hành gộp 2 cột này lại thành 1 cột
VD: Cột “Máy pha trà/cà phê” và cột “Máy pha Cà phê”

Tiến hành xây preprocess_pipeline: bao gồm `ColAdderDropper()` ở trên và thêm các bước sau:

- Numerical columns: điền giá trị thiếu bằng giá trị trung bình của thuộc tính (mean)
- Catogerial columns: điền giá trị thiếu bằng giá trị phổ biến nhất (most frequent) và sử dụng OneHotEncoder
- Feature columns (kiểu numerical): điền giá trị thiếu bằng giá trị phổ biến nhất (most frequent)
- Cuối cùng, chuẩn hóa bằng StandardScaler()

Saturday
16.01.2021

Xây dựng mô hình

XÂY DỰNG MÔ HÌNH

Nhóm em chọn 2 mô hình regressor có trong sklearn:

- MLPRegressor
- RandomForestRegressor
- Và thêm 1 bước chuẩn hóa bằng TransformedTargetRegressor()

Nhóm sử dụng độ lỗi Mean Absolute Error (MAE) để đánh giá mô hình

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}$$

XÂY DỰNG MÔ HÌNH

MLPRegressor trong Neural Network:

- Nhóm đã thử nghiệm 1 vài mô hình với thiết lập hidden_layers và chọn ra được mô hình tối ưu nhất như sau:

```
# Neural Net regressor
mlp_regressor = MLPRegressor(hidden_layer_sizes=(100, 300, 300, 100), solver='adam', learning_rate='adaptive',
                             random_state=0, max_iter=400, early_stopping=True, verbose=0)

model = TransformedTargetRegressor(regressor=mlp_regressor,
                                   func=np.log, inverse_func=np.exp)

# Full pipeline = preprocess_pipeline + model
full_pipeline = Pipeline(steps = [
    ('preprocessor', preprocess_pipeline),
    ('model', model)
])
```

Note: solver bọn em dùng "adam" vì theo bọn em tìm hiểu thì "adam" dùng khi lượng dữ liệu lớn, còn dữ liệu nhỏ thì dùng "lbfgs"

XÂY DỰNG MÔ HÌNH

MLPRegressor trong Neural Network:

- Sau đó nhóm em tiến hành tuning để tìm ra learning_rate (alpha) tốt nhất: best_alpha = 1

```
val_MAEs = []
train_MAEs = []
alphas = [0.1, 1, 10, 100, 1000]
best_val_MAE = float('inf'); best_alpha = None;
for alpha in alphas:
    full_pipeline.set_params(model__regressor__alpha=alpha) # nested pipeline
    full_pipeline.fit(train_X_df, train_y_sr)
    preds_y = full_pipeline.predict(val_X_df)
    val_mae = mean_absolute_error(preds_y, val_y_sr)
    print("MAE =", val_mae)
    val_MAEs.append(val_mae)
    preds_train_y = full_pipeline.predict(train_X_df)
    train_MAEs.append(mean_absolute_error(preds_train_y, train_y_sr))

    if best_val_MAE > val_mae:
        best_val_MAE = val_mae
        best_alpha = alpha

'Finish!'
```


XÂY DỰNG MÔ HÌNH

RandomForestRegressor:

- Nhóm em tiến hành thiết lập ban đầu như sau:

```
# Random forest regressor
rf_regressor = RandomForestRegressor(n_estimators=800, random_state=0, verbose=0)

model = TransformedTargetRegressor(regressor=rf_regressor,
                                   func=np.log, inverse_func=np.exp)

# Full pipeline = preprocess_pipeline + model
full_pipeline = Pipeline(steps = [
    ('preprocessor', preprocess_pipeline),
    ('model', model)
])
```

XÂY DỰNG MÔ HÌNH

RandomForestRegressor:

- Sau đó nhóm em tiến hành tuning để tìm ra n_estimators (number of tree) tốt nhất: best_estimator = 400

```
val_MAEs = []
train_MAEs = []
my_n_estimators = [100, 200, 400, 600, 800, 1000, 1500]
best_val_MAE = float('inf'); best_estimator = None;
for estimator in my_n_estimators:
    full_pipeline.set_params(model__regressor__n_estimators=estimator) # nested pipeline
    full_pipeline.fit(train_X_df, train_y_sr)
    preds_y = full_pipeline.predict(val_X_df)
    val_mae = mean_absolute_error(preds_y, val_y_sr)
    print("MAE =", val_mae)
    val_MAEs.append(val_mae)
    preds_train_y = full_pipeline.predict(train_X_df)
    train_MAEs.append(mean_absolute_error(preds_train_y, train_y_sr))

    if best_val_MAE > val_mae:
        best_val_MAE = val_mae
        best_estimator = estimator
'Finish!'
```

Saturday
16.01.2021

Đánh giá và tổng kết

ĐÁNH GIÁ VÀ TỔNG KẾT

Dựa trên kết quả ở phần xây mô hình, bọn em chọn mô hình **RandomForestRegressor** vì mô hình này cho kết quả tốt hơn

```
MAE = 774003.2266868228  
MAE = 777301.3279403553  
MAE = 900038.2867390646  
MAE = 1030269.1673290639  
MAE = 1179610.7436706966
```

Độ lỗi của MLPRegressor

```
MAE = 697962.2556765627  
MAE = 695554.9814056441  
MAE = 693861.2915576712  
MAE = 695033.2259001777  
MAE = 694798.7667132171  
MAE = 694596.1129243408  
MAE = 694042.9709465638
```

Độ lỗi của RandomForestRegressor

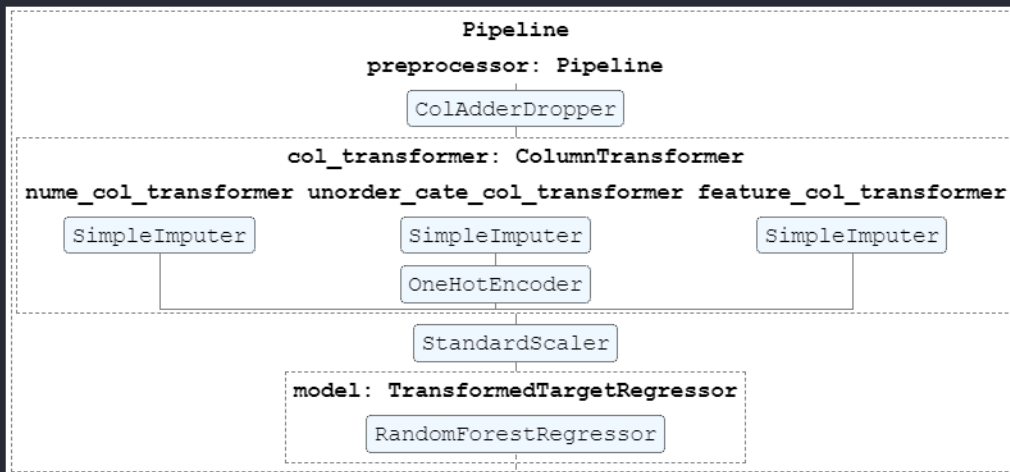
Cả 2 model đều cho độ lỗi tương đối cao, theo nhóm em thì có lẽ lí do là:

- Nhiều do có discount (giảm giá) trong những ngày nhóm crawl data
- Có quá nhiều cột: bọn em đã cố gắng xử lí nhưng không được tối ưu cho lắm
- Tập giá trị của các thuộc tính số quá ít nên sẽ khó dự đoán

ĐÁNH GIÁ VÀ TỔNG KẾT

Train lại model trên tập train + validation và test trên tập test

```
# fit on total (train+validation)
full_pipeline.fit(temp_X_df, temp_y_sr)
```



▶ M1

```
pred_test_y = full_pipeline.predict(test_X_df)
print("MAE =", mean_absolute_error(test_y_sr, pred_test_y))
```

MAE = 444715.220964204

NẾU CÓ THÊM THỜI GIAN THÌ SẼ LÀM GÌ?

- Cẩn thận hơn lúc crawl data: loại bỏ nhiễu - discount (giảm giá) trong những ngày nhóm crawl data
- Thử trên một vài model mạnh hơn: XGBoots, Gradient Bootings
- Mở rộng thêm data từ nhiều nguồn để có thể dự đoán cho nhiều nơi ở nhiều nước khác nhau nhằm phục vụ cho nhiều đối tượng hơn

Saturday
16.01.2021

Tham khảo

THAM KHẢO

Các nguồn tài liệu đã tham khảo:

[1] File notebook BT03 của thầy Kiên.

[2] Slide môn học và đặc biệt là slide DoAnCK.pdf

[3][Scikit-learn documentation](<https://scikit-learn.org/0.21/documentation.html>)

[4][pandas documentation](<https://pandas.pydata.org/docs/>)

[5] https://www.answers.com/Q/When_regression_is_not_applicable

[6] <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/discussion/103975>

Saturday
16.01.2021

Cảm ơn thầy đã theo dõi!
