Nhập môn khoa học dữ liệu

Báo cáo đồ án cuối kỳ

Nhóm 36:

1712284 - Hoàng Gia Bảo

18120164 - Lê Minh Đức

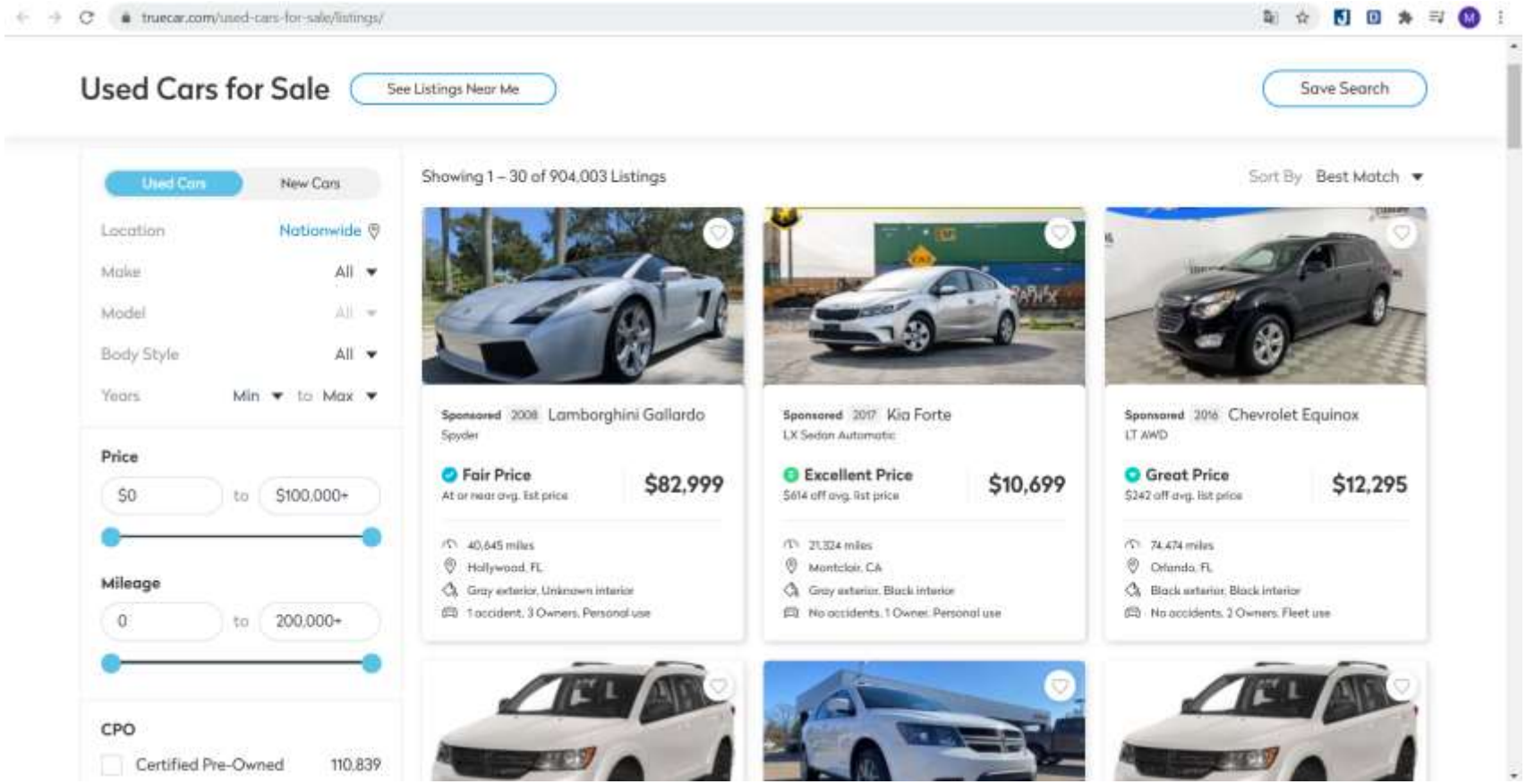
GV: Thầy Trần Trung Kiên

Nội dung

- Thu thập dữ liệu
- Khám phá dữ liệu
- Đặt câu hỏi
- Tiền xử lí dữ liệu
- Mô hình hóa
- Nhìn lại quá trình làm đồ án
- Tài liệu tham khảo

Thu thập dữ liệu

- Thu thập dữ liệu thông tin về xe hơi đã qua sử dụng trên trang web https://www.truecar.com/used-cars-for-sale/listings/



Thu thập dữ liệu

- Đầu tiên, kiểm tra xem liệu trang web có cho phép mình thu thập dữ liệu?

Dữ liệu thu thập hợp pháp (check trước khi crawl)

```
rp = urllib.robotparser.RobotFileParser()
rp.set_url('https://www.cars-data.com/robots.txt')
rp.read()
rp.can_fetch('*', 'https://www.truecar.com/used-cars-for-sale/listings/')
```

```
True
```

Kiểm tra trên 1 ô tô cần crawl

```
rp.can_fetch('*', 'https://www.truecar.com/used-cars-for-sale/listing/1FTNE1EL8CDA68267/2012-ford-econoline-cargo-van/')
```

```
True
```

- Việc thu thập là hoàn toàn hợp pháp

Thu thập dữ liệu

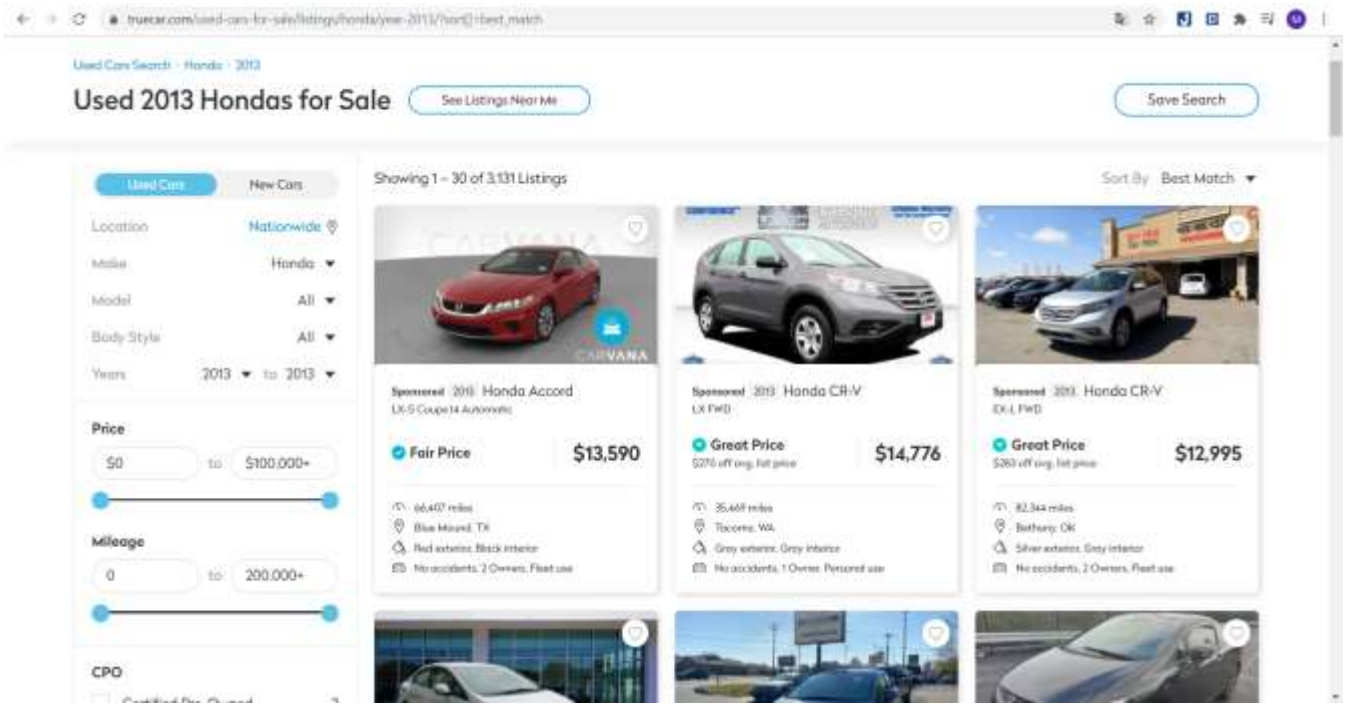
- Sử dụng thư viện selenium để chỉnh cột bên trái chọn hãng (Honda, Ford, Huyndai, Mercedes-Benz, Toyota, Mazda), Years: 2013-2020

- VD: năm 2013: lấy thông tin 100 xe Honda, 100 xe Ford, 100 xe Huyndai, 100 xe Mercedes-Benz, 100 xe Toyota, 100 xe Mazda

- Tiếp tục làm vậy đến năm 2020

→có tổng cộng (100*6)*8 = 4800 thông tin xe

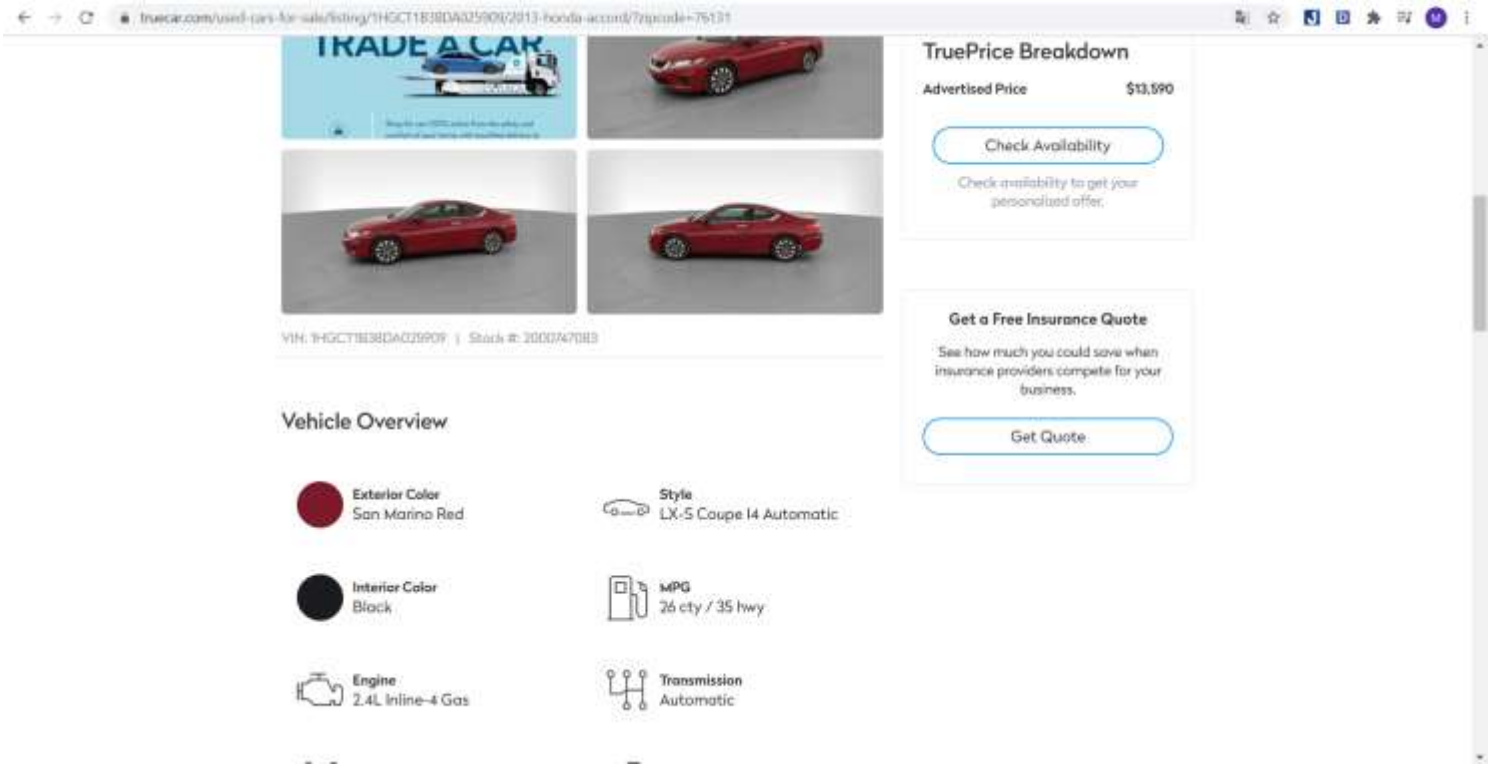
- Ở mỗi trang như vậy sẽ có khoảng 30 xe và ta sẽ parse HTML để lấy đường link của từng xe và đi vào để lấy dữ liệu chi tiết của nó



Thu thập dữ liệu

- Đây là trang chi tiết của một ô tô và ta sẽ thu thập những thông tin sau: Brand/make, Model, Body Style, Year, Mileage, Engine, Fuel type, Number of owners, Price

- Sau khi thu thập hết dữ liệu của 1 trang ban đầu ta sẽ click sang trang tiếp theo cho tới khi đủ 100 dữ liệu và tiếp tục như vậy với các năm khác, các hãng khác tới khi đủ 4800 dữ liệu



Khám phá dữ liệu

- Sơ lược qua về dữ liệu
- Có tổng cộng 4800 dòng và 9 cột
- Ý nghĩa của mỗi cột như sau

	make	model	body style	year	mileage	engine	fuel type	number of owners	price
0	Honda	Odyssey	Touring Elite	2013	122,986	3.5L V-6 Gas	Gas	2.0	12,895
1	Honda	Pilot	EX FWD	2013	80,532	3.5L V-6 Gas	Gas	3.0	11,694
2	Honda	CR-V	LX FWD	2013	79,796	2.4L Inline-4 Gas	Gas	1.0	13,590
3	Honda	Civic	LX Sedan Automatic	2013	76,882	1.8L Inline-4 Gas	Gas	1.0	7,986
4	Honda	Civic	LX Sedan Automatic	2013	113,673	1.8L Inline-4 Gas	Gas	1.0	6,990

- **Cột "make"**: tên hãng xe
- **Cột "model"**: tên dòng xe
- **Cột "body style"**: dáng xe (kết cấu thân xe)
- **Cột "year"**: năm sản xuất chiếc xe
- **Cột "mileage"**: tổng số dặm xe đã đi
- **Cột "engine"**: thông số động cơ xe
- **Cột "fuel type"**: xe đi loại nhiên liệu gì?
- **Cột "number of owners"**: xe đã qua tay bao nhiêu đời chủ rồi?
- **Cột "price"**: giá thành xe hiện được bán (USD)

Khám phá dữ liệu – Tiền xử lý dữ liệu thô

- Ta sẽ kiểm tra các dòng trùng lặp và xóa nó
- Ta sẽ thay thế dấu ‘,’ bằng dấu ‘.’ ở cột ‘mileage’ và ‘price’ và chuyển nó về dạng số và ta thu được tập dữ liệu
- Kết quả thu được còn 4652 dòng và 9 cột

```
# bỏ dấu phẩy
used_car_df.loc[:, 'mileage'] = used_car_df.loc[:, 'mileage'].str.replace(',', '')
used_car_df.loc[:, 'price'] = used_car_df.loc[:, 'price'].str.replace(',', '')
```

```
# chuyển sang kiểu float
used_car_df['mileage'] = used_car_df['mileage'].astype(float)
used_car_df['price'] = used_car_df['price'].astype(float)
used_car_df
```

	make	model	body style	year	mileage	engine	fuel type	number of owners	price
0	Honda	Odyssey	Touring Elite	2013	122986.0	3.5L V-6 Gas	Gas	2.0	12895.0
1	Honda	Pilot	EX FWD	2013	80532.0	3.5L V-6 Gas	Gas	3.0	11694.0
2	Honda	CR-V	LX FWD	2013	79796.0	2.4L Inline-4 Gas	Gas	1.0	13590.0
3	Honda	Civic	LX Sedan Automatic	2013	76882.0	1.8L Inline-4 Gas	Gas	1.0	7986.0
4	Honda	Civic	LX Sedan Automatic	2013	113673.0	1.8L Inline-4 Gas	Gas	1.0	6990.0
...	...	...	...	...	...	...	...	...	...
4795	Mazda	CX-5	Grand Touring FWD	2020	13964.0	2.5L Inline-4 Gas	Gas	1.0	23888.0
4796	Mazda	CX-5	Sport FWD	2020	3759.0	2.5L Inline-4 Gas	Gas	1.0	22498.0
4797	Mazda	CX-5	Sport FWD	2020	9916.0	2.5L Inline-4 Gas	Gas	1.0	21990.0
4798	Mazda	CX-9	Grand Touring AWD	2020	5176.0	2.5L Inline-4 Gas Turbocharged	Gas	1.0	36992.0
4799	Mazda	CX-5	Touring AWD	2020	4902.0	2.5L Inline-4 Gas	Gas	0.0	25737.0

4652 rows × 9 columns

Đặt câu hỏi

- **Câu hỏi:** làm sao để dự đoán giá trị của một chiếc xe cũ, đã qua sử dụng dựa trên các thông số của chiếc xe đó?
- Câu hỏi này giải quyết được 2 vấn đề trong thực tế:
  - Cho các người chủ muốn bán xe:
    - Họ ước lượng trước được giá trị của chiếc xe mà mình muốn bán dựa trên giá thị trường hiện tại. Để khi đến bất kì nơi bán nào, họ có thể thương lượng giá cả tốt hơn
  - Cho các đại lý mua bán xe cũ:
    - Biết được giá trị thị trường, để có thể thương lượng giá cả với người bán
    - Nếu đã sở hữu xe cũ --> có thể đưa ra giá trị chiếc xe để bán đi
- **Nguồn gốc**: ý tưởng này sinh khi vô tình xem được một clip đánh giá siêu xe trên Youtube

Khám phá dữ liệu (để tách các tập)

- Vì đây là bài toán hồi quy, nên cần kiểm tra cột output (giá xe) đã mang giá trị số chưa
- Cần kiểm tra cột output có còn giá trị thiếu hay không? Nếu còn bỏ dòng đó khỏi dữ liệu thô hiện tại

Sau khi xử lý xong 2 điều kiện trên

→ Tiến hành tách train:val:test theo tỉ lệ 60:20:20 của dữ liệu thô

Khám phá dữ liệu (Tập huấn luyện)

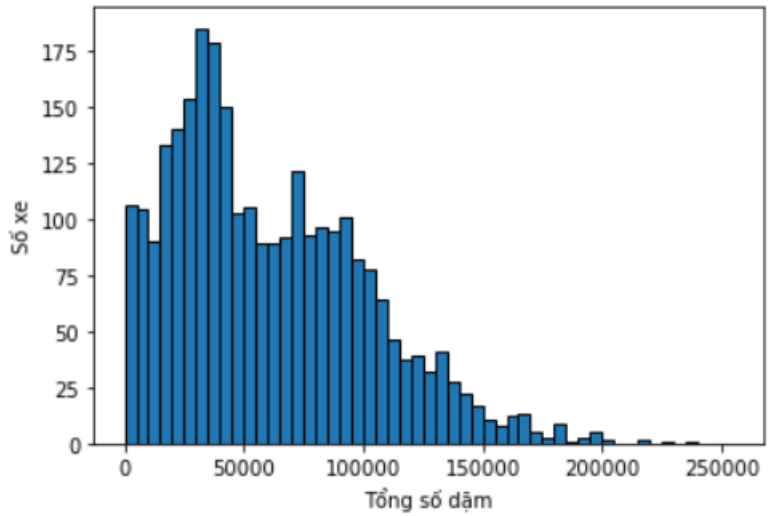
- Khám phá dữ liệu để có thể hiểu thêm dữ liệu cho việc tiền xử lí dữ liệu
- Những cột dạng số

	year	mileage	number of owners
missing_ratio	0.0	0.00	0.035868
min	2013.0	5.00	0.000000
lower_quartile	2014.0	29219.50	1.000000
median	2017.0	52173.00	1.000000
upper_quartile	2019.0	88003.75	2.000000
max	2020.0	257866.00	6.000000

Biểu đồ histogram biểu thị mối liên hệ giữa: tổng số dặm và số lượng xe

```
1 train_X_df['mileage'].plot.hist(bins = range(5, 260000, 5000), edgecolor = 'black')
2 plt.xlabel('Tổng số dặm')
3 plt.ylabel('Số xe')
```

```
Text(0, 0.5, 'Số xe')
```



Khám phá dữ liệu (Tập huấn luyện)

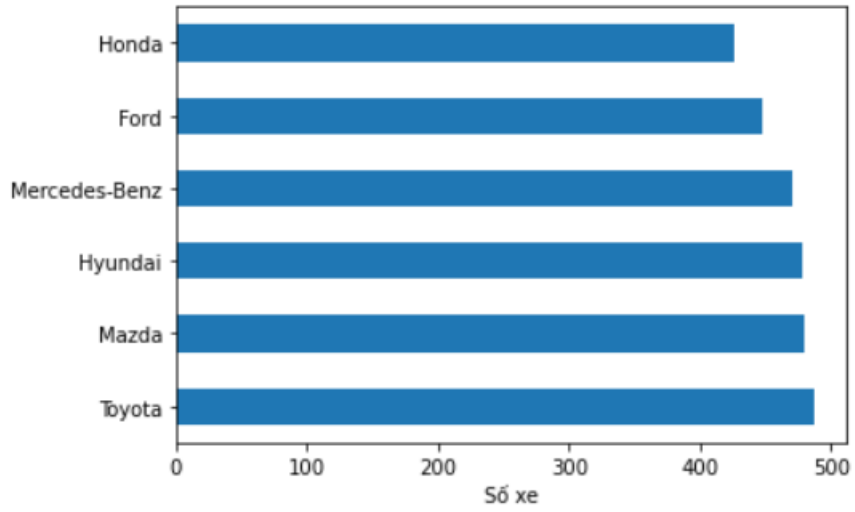
- Với những cột không phải dạng số

	make	model	body style	engine	fuel type
missing_ratio	0	0	0	0	0
nunique	6	99	619	53	7
value_counts	{'Toyota': 487, 'Mazda': 479, 'Hyundai': 478, ...	{'CX-5': 220, 'Elantra': 193, 'Civic': 152, 'C...	{'LE CVT': 75, 'SE FWD': 65, 'LX Sedan CVT': 5...	{'2.5L Inline-4 Gas': 490, '2.0L Inline-4 Gas'...	{'Gas': 2639, 'Hybrid': 109, 'Diesel': 22, 'Pl...

Biểu đồ bar chart thể hiện số lượng xe của từng hãng

```
1 train_X_df['make'].value_counts().plot.barh()
2 plt.xlabel('Số xe')
```

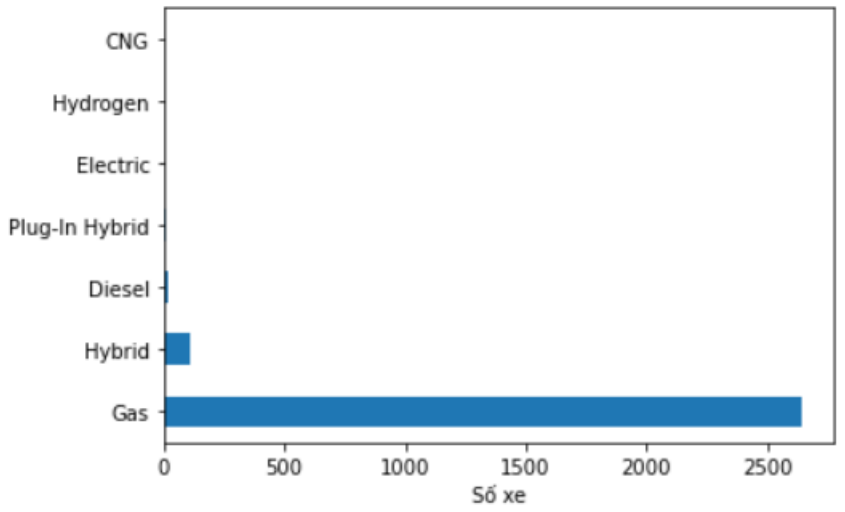
```
Text(0.5, 0, 'Số xe')
```



Biểu đồ bar chart thể hiện mối quan hệ giữa: loại nhiên liệu và số lượng xe

```
1 train_X_df['fuel type'].value_counts().plot.barh()
2 plt.xlabel('Số xe')
```

```
Text(0.5, 0, 'Số xe')
```



Tiền xử lí (tập huấn luyện)

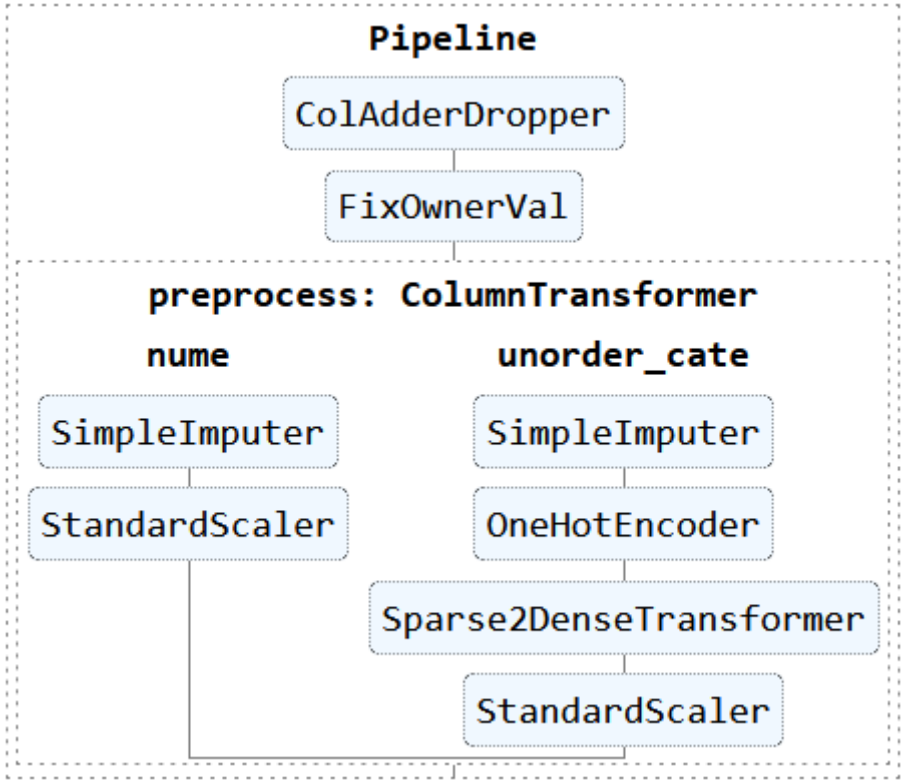
- Sau khi khám phá tập huấn luyện ta cần xử lý:
  - Cột “body-style” có quá nhiều giá trị → Xóa cột này
  - Đơn giản hóa cột “engine” bằng cách chỉ lấy thông số thể tích động cơ ở dạng số
  - Cột “fuel types”: chủ yếu các xe chạy bằng gas, nên ta sẽ giữ nguyên giá trị gas, các giá trị khác chuyển thành “Others”
  - Cột “model” giữ lại các giá trị có số lượng nhiều nhất, các giá trị còn lại chuyển thành “Others”

→ Dùng class **ColAdderDropper** để thực hiện các việc này

Tiền xử lí (tập huấn luyện)

- Với các cột dạng số, ta sẽ điền giá trị thiếu bằng giá trị median của cột
- Lưu ý với cột "number of owners", xử lý 1 bước trước khi xử lý giá trị thiếu, đó là: nếu giá trị trong cột "number of owners" = 0 thì ta chuyển nó = 1. Vì tiêu chí của ta là các xe đã qua sử dụng, mà đã qua sử dụng thì phải có >= 1 người chủ (class **FixOwnerVal**)
- Với các cột không phải dạng số và không có thứ tự:
  - Ta sẽ điền giá trị thiếu bằng giá trị mode (giá trị xuất hiện nhiều nhất) của cột
  - Chuyển ma trận thưa sang ma trận dày đặc bằng class **Sparse2Dense**
  - Sau đó, ta sẽ chuyển sang dạng số bằng phương pháp mã hóa one-hot

Tiền xử lí (tập huấn luyện)



Cấu trúc của preprocessing_pipeline

Mô hình hóa

- Thử nghiệm với 2 mô hình:
  - Neural Network (MLPRegressor)
  - Linear Regression

→ Để đánh giá kết quả giữa một mô hình phức tạp và một mô hình đơn giản

Mô hình hóa

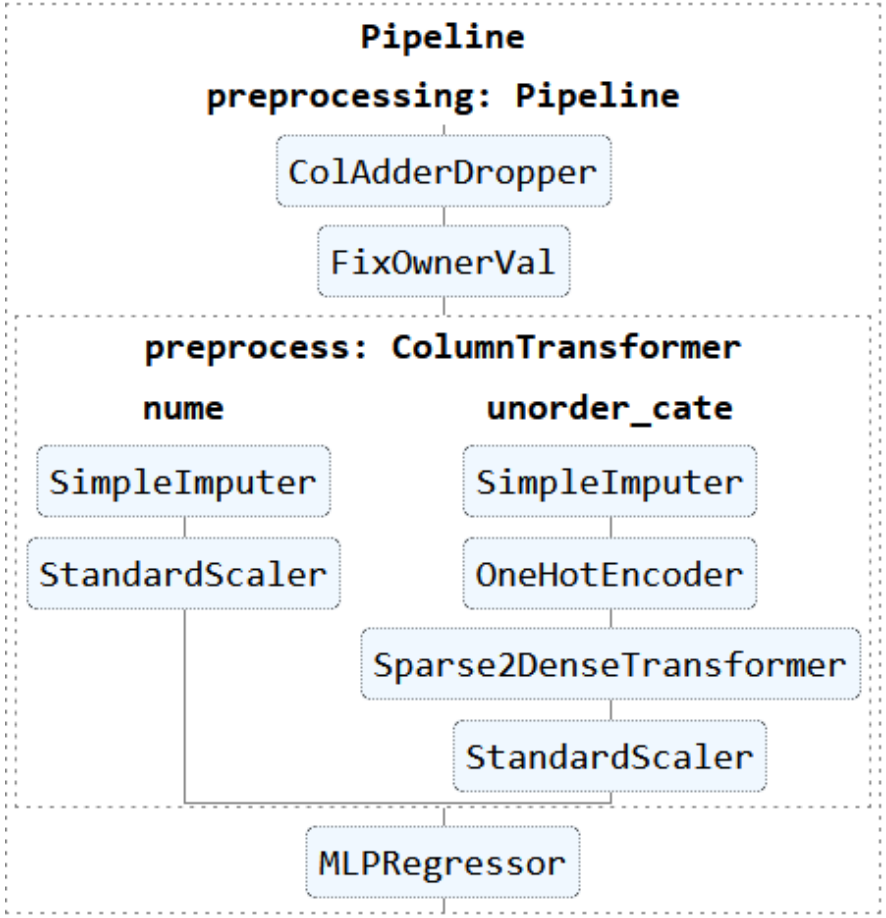
1/ Mô hình Neural Network:

```
neural_net = MLPRegressor(hidden_layer_sizes=(20),
                          activation='relu',
                          solver='adam',
                          random_state=0,
                          max_iter=2500)
```

Nhóm đã tạo mô hình mạng nơ ron cho bài toán hồi quy với các tham số:

- hidden_layer_sizes = (20): chỉ có 1 tầng ẩn, tầng này có 20 nơ ron
- activation = ‘relu’: hàm kích hoạt relu
- solver = ‘adam’: phương pháp cực tiểu hóa độ lỗi trên tập train. Document của sklearn gợi ý dùng solver này nếu dữ liệu train là hàng ngàn mẫu
- max_iter = 2500: mỗi mẫu sẽ được sử dụng trên train 2500 lần

Mô hình hóa



Full_pipeline thể hiện quá trình xử lý tập dữ liệu đến khi mô hình hóa

Mô hình hóa

- Dùng full_pipeline trên để tiến hành thử nghiệm 2 siêu tham số:
  - Siêu tham số alpha của MLPRegressor

    alpha = [0.1, 1, 10, 100, 1000]
  - Siêu tham số n_top_models của class ColAdderDropper

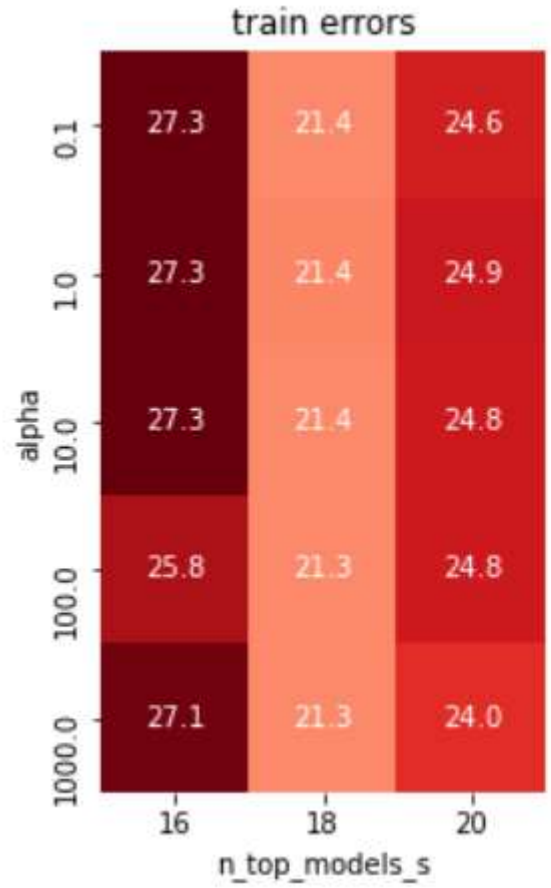
    n_top_models = [16,18, 20]

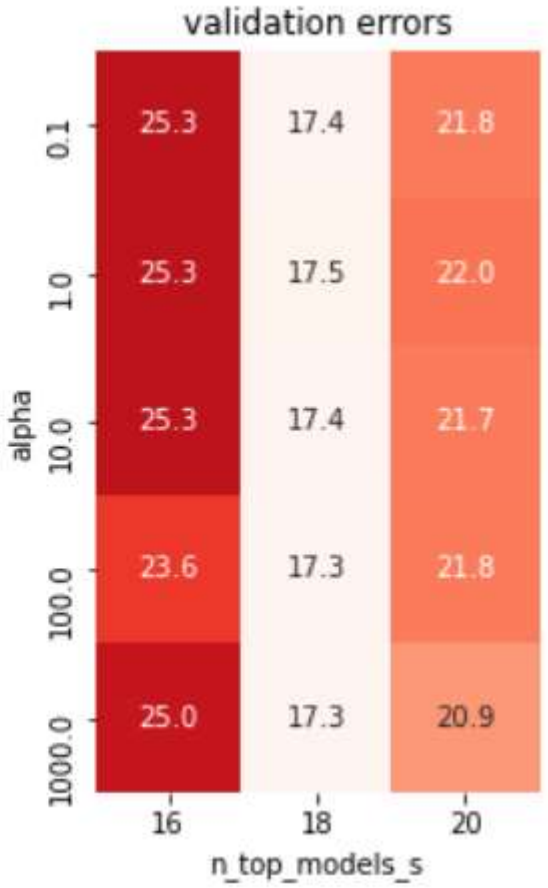
Tiến hành thử nghiệm, ta chọn ra được 2 siêu tham số làm độ lỗi trên tập val thấp nhất ~ 17% đó là:

- alpha = 100
- n_top_models = 18

Mô hình hóa

Trực quan hóa độ lỗi trên tập train và tập val dựa vào 2 siêu tham số





Mô hình hóa

- Tạo mô hình tốt nhất dựa trên 2 siêu tham số đã tìm được

```
full_pipeline.set_params(preprocessing__add_drop__n_top_models=best_n_top_models,
                         neural_net__alpha=best_alpha)
```

- Cuối cùng, ta sẽ huấn luyện lại full_pipeline trên X_df và y_sr (tập huấn luyện + tập validation) với best_alpha và best_n_top_models tìm được ở trên để ra được mô hình cụ thể cuối cùng

```
X_df = pd.concat([train_X_df,val_X_df])
y_sr = pd.concat([train_y_sr,val_y_sr])
full_pipeline.fit(X_df, y_sr)
```

Mô hình hóa

- Đánh giá mô hình neural network tìm được trên tập test

```
# Tính độ đo r^2 trên tập test
baseline_preds = y_sr.mean()
(compute_rr(test_y_sr, full_pipeline.predict(test_X_df), baseline_preds))*100
```

```
85.28598446263757
```

Độ chính xác trên tập test đạt ~ 85%

Mô hình hóa

2/ Thử nghiệm với mô hình Linear Regression:

Làm tương tự các bước như thử nghiệm với mạng nơ ron, chỉ thay mô hình dự đoán thành LinearRegression

Huấn luyện lại full_pipeline trên X_df và y_sr (tập huấn luyện + tập validation)

Đánh giá mô hình trên tập test, ta thấy độ chính xác ~ 79%

Mô hình hóa

3/ Nhận xét giữa 2 mô hình

- Mô hình Neural Network (MLPRegressor):
  - Độ chính xác trên tập test khoảng 85%
  - Về mặt thời gian thì mô hình Neural Network train khá lâu, do ta phải tìm ra siêu tham số tốt nhất cho mô hình
- Mô hình Linear Regression:
  - Tuy đây là mô hình khá đơn giản nhưng độ chính xác cũng không quá tệ (khoảng 79%)
  - Về mặt thời gian thì rõ ràng nhanh hơn rất nhiều so với mô hình Neural Network

Nhìn lại quá trình làm đồ án

1/ Những khó khăn

- Về quá trình thu thập dữ liệu: tốn khá nhiều thời gian, và cấu trúc của các chi tiết của mỗi ô tô cũng không giống nhau nên ban đầu, parse HTML xảy ra nhiều lỗi
- Ở bước tiền xử lý dữ liệu: do đây là lần đầu tiên áp dụng kỹ thuật pipeline nên đã xảy ra rất nhiều lỗi kỹ thuật khó khăn trong suốt quá trình làm preprocessing_pipeline
- Ở bước mô hình hóa dữ liệu: do thời gian khá lâu nên việc tìm ra các siêu tham số thích hợp khá tốn thời gian

Nhìn lại quá trình làm đồ án

2/ Những kiến thức bổ ích đã học được

- Quy trình Khoa học dữ liệu cơ bản
- Kỹ thuật dùng pipeline để đơn giản hóa code, và tránh các lỗi sai không mong muốn
- Kỹ thuật dùng thông tin tập train để biến đổi cho tập valid và tập test, chứ không phải tính trên từng tập, hay tính trên toàn tập rồi mới chia nhỏ như trước đây hay làm → nhận ra lỗi sai ngay đây
- Được áp dụng quy trình Khoa học dữ liệu để làm một bài toán thực tế, được trải nghiệm quá trình làm luôn
- Cách sử dụng git/github để quản lý project của nhóm, ngoài ra còn rèn luyện khả năng làm việc nhóm

Nhìn lại quá trình làm đồ án

3/ Nếu còn thời gian, sẽ làm những gì?

Nếu có thời gian, nhóm sẽ thu thập và dự đoán đa dạng các hãng xe hơn nữa, ngoài ra sẽ tìm hiểu nhiều kỹ thuật để có thể tạo ra một mô hình dự đoán giá xe cũ tốt nhất có thể

Tài liệu tham khảo

- Bài tập 03
- Các slide bài giảng
- Các code demo
- Document về các hàm của sklearn