Nhập môn Khoa học dữ liệu

Đồ án cuối kì

Dự đoán giá điện thoại

Giáo viên hướng dẫn: Trần Trung Kiên
Nhóm 13: Thái Hoàng Lâm 18120434
Nguyễn Văn Minh 18120464

Nội dung

- Thu thập dữ liệu
- Khám phá dữ liệu
- Tiền xử lý
- Đưa ra câu hỏi
- Mô hình hóa
- Nhìn lại quá trình

Thu thập dữ liệu

- Hình thức thu thập: Parse HTML
- Dữ liệu được thu thập trên trang https://www.dienmayxanh.com/dien-thoai
- Nội dung trang web có **JavaScript** → Sử dụng thư viện Selenium
- Check file robots.txt => được cho phép.

Thu thập dữ liệu

- Parse trang web để lấy link vào trang thông tin chi tiết từng sản phẩm
- Sau khi đã lấy được tất cả các phone link và lưu vào file phone_links.txt
- Số link phone ta có là 179
- Parse mỗi trang web tương ứng với mỗi link để lấy dữ liệu điện thoại

```
Entrée [6]: print('Tổng số phone link ta có: ',len(links))

            Tổng số phone link ta có:  179
```

Khám phá dữ liệu

- Hình ảnh dữ liệu thu thập được

	Phone_Name	Brand	Screen	OS	Main_Camera	Front_Camera	CPU	RAM	Storage	Battery	Price
0	oppo-reno5	oppo	6.43	Android 11	Chính 64	44	Snapdragon	8.0	128.0	4310	8.690.000
1	oppo-a92	oppo	6.50	Android 10	Chính 48	16	Snapdragon	8.0	128.0	5000	6.490.000₫ -4%
2	iphone-12-mini	iphone	5.40	iOS 14	2 camera 12	12	Apple	4.0	64.0	2227	21.990.000
3	xiaomi-mi-10t-pro	xiaomi	6.67	Android 10	Chính 108	20	Snapdragon	8.0	256.0	5000	12.990.000₫ -5%
4	samsung-galaxy-a12-6gb	samsung	6.50	Android 10	Chính 48	8	MediaTek	6.0	128.0	5000	4.690.000₫ -2%
5	samsung-galaxy-m51	samsung	6.70	Android 10	Chính 64	32	Snapdragon	8.0	128.0	7000	9.490.000₫ -1%
6	realme-c15	realme	6.50	Android 10	Chính 13	8	Snapdragon	4.0	64.0	6000	4.190.000₫ -4%
7	vivo-y51-2020	vivo	6.58	Android 11	Chính 48	16	Snapdragon	8.0	128.0	5000	6.290.000₫ -4%
8	samsung-galaxy-z-fold-2	samsung	7.59	Android 10	3 camera 12	10	Snapdragon	12.0	256.0	4500	50.000.000
9	samsung-galaxy-z-fold2-5g-dac-biet	samsung	7.59	Android 10	3 camera 12	10	Snapdragon	12.0	256.0	4500	50.000.000

Khám phá dữ liệu

- Dữ liệu bao gồm: 175 dòng và 11 cột
- Ý nghĩa mỗi côt:
- Phone_Name: tên của điện thoại
- Brand: tên của thương hiệu
- Screen: thông tin màn hình
- OS: Hệ điều hanh
- Main_Camera: Độ phân giải của camera trước
- Front_Camera: Độ phân giải của camera sau
- CPU: bộ xử lý trung tâm
- Ram: bộ nhớ dữ liệu tạm thời
- Storage: Kích thước bộ nhớ trong
- Battery: Dung lượng pin
- Price: Giá

Khám phá dữ liệu

- Dữ liệu gặp những vấn đề sau:
- Dữ liệu chứa các giá trị thiếu (NA)
- Một số cột dữ liệu chưa đúng định dạng kiểu dữ liệu

```
df.isna().sum()
```

```
Brand            0
Screen           0
OS              38
Main_Camera     20
Front_Camera    40
CPU             38
RAM             41
Storage         37
Battery          0
Price            1
dtype: int64
```

```
df.dtypes
```

```
Brand           object
Screen         float64
OS              object
Main_Camera     object
Front_Camera    object
CPU             object
RAM            float64
Storage        float64
Battery          int64
Price           object
dtype: object
```

Tiền xử lý

- Cột Price (dự định sẽ là cột output): loại bỏ ký tự khác số, chuyển về dạng int, đồng thời loại bỏ dòng missing price

```
df.drop(df[df['Price'].isna()].index,inplace =True)
df['Price']= df['Price'].str.extract(r'(\d+\.\d+[.]?[\d]*)', expand=False)
df['Price']= df['Price'].str.replace(r'[.]','',regex=True).astype(dtype = 'int64',errors = 'ignore')
```

- Cột Main_Camera và Front_Camera: Lấy chuỗi có dạng số float or int Chuyển về dạng float.

```
df.Main_Camera = pd.to_numeric(df.Main_Camera.str.extract(r'([\d.]+)', expand=False), errors='coerce')
```

```
df.Front_Camera = pd.to_numeric(df.Front_Camera.str.extract(r'([\d.]+)', expand=False), errors='coerce')
```

Tiền xử lý

- Xem xét cột Price

```
df['Price'].value_counts()
```

```
23990000    6
6990000     6
3990000     5
4990000     5
12990000    4
           ..
890000      1
2290000     1
18990000    1
1250000     1
36000000    1
Name: Price, Length: 99, dtype: int64
```

Nhận xét:

Ta thấy rằng cột Price hiện có đến 99 giá trị khác nhau

⇒ Khó để phân lớp

⇒ Ta làm mịn dữ liệu cột price bằng cách chia bin thành 5 khoảng và gán nhãn class từ 1->5

Tiền xử lý

- Chúng ta sẽ chia bin giá của điện thoại theo luật sau:
- Phân khúc điện thoại cơ bản (label = 1): Price <= 1,000,000
- Phân khúc điện thoại phổ thông (label = 2): 1,000,000 < Price <=6,000,000
- Phân khúc điện thoại trung cấp (label = 3): 6,000,000 < Price <= 10,000,000
- Phân khúc điện thoại cận cao cấp (label = 4): 10,000,000 < Price <= 18,000,000
- Phân khúc điện thoại cao cấp (label = 5): Price > 18,000,000

- Thêm cột Class chứa các bin này
- Cột Price không còn ý nghĩa → Xóa đi

Tiền xử lý

- Dữ liệu sau khi được chia bin

	Brand	Screen	OS	Main_Camera	Front_Camera	CPU	RAM	Storage	Battery	Class
13	samsung	6.90	Android 10	108.0	10.0	Exynos	12.0	256.0	4500	5
77	xiaomi	6.53	Android 10	48.0	8.0	Snapdragon	6.0	128.0	6000	2
168	masstel	1.77	NaN	NaN	NaN	NaN	NaN	NaN	800	1
15	samsung	6.90	Android 10	108.0	10.0	Exynos	8.0	256.0	4500	5
140	itel	2.40	KaiOS	0.3	NaN	Spreadtrum	NaN	4.0	1900	1
152	itel	2.40	NaN	0.3	NaN	NaN	NaN	NaN	1200	1
129	nokia	5.70	NaN	5.0	5.0	Spreadtrum	1.0	16.0	2800	2
122	xiaomi	6.53	Android 10	13.0	5.0	MediaTek	2.0	32.0	5000	2
29	vivo	6.56	Android 10	48.0	32.0	Snapdragon	8.0	256.0	4315	5
154	mobell	2.40	NaN	NaN	NaN	NaN	NaN	NaN	1200	1

Đưa ra câu hỏi:

Điện thoại với những tính năng gì sẽ có tầm giá thế nào?

- Input: Thương hiệu và các cột tính năng của điện thoại bao gồm: Brand, Screen, OS, Main_Camera, Front_Camera, CPU RAM,Storage, Battery
- Output: cột Class
- → Chúng ta sẽ dự đoán phân khúc tầm giá của một sản phẩm điện thoại mới thông qua thương hiệu và tính năng của nó. Chúng ta thấy rằng, khi ra mắt một sản phẩm smart_phone mới, các nhà sản xuất thường đưa ra tính năng dự kiến trước rồi mới chính thức đưa ra giá phù hợp với từng thị trường.

Nguồn cảm hứng câu hỏi:

- Chúng em tự nghĩ ra từ việc yêu thích công nghệ, cập nhật các sản phẩm chuẩn bị ra mắt cũng như dự đoán giá điện thoại để tiết kiệm tiền mua.

Trả lời câu hỏi được lợi ích gì?

- **Đối với nhà sản xuất:** tính toán hợp lý tính năng của điện thoại để phù hợp với giá tiền khi phân phối tại thị trường Việt Nam.

=> **Từ đó, họ sẽ tối ưu hoá được lợi nhuận, thu hút được người dùng. Chẳng hạn như hãng SamSung: họ sản xuất nhiều dòng điện thoại từ cơ bản đến cao cấp với tính năng và giá tiền phù hợp với mọi đối tượng người tiêu dùng ở Việt Nam.**

- **Đối với người tiêu dùng:** Biết được điện thoại mình muốn mua với những tính năng mình cần sẽ ở tầm giá nào.

=> **Tránh được việc mua điện thoại ở những cửa hàng không uy tín như mua lầm hàng nhái (tính năng cao nhưng giá thấp). Hoặc trong trường hợp như là sinh viên, có thể lên kế hoạch tiết kiệm tiền để mua.**

Khám phá dữ liệu (để tách các tập)

- Cột output có kiểu dữ liệu gì? Int32
- Cột ouput có giá trị thiếu không? 0
- Tỉ lệ các lớp trong Ouput?

```
2    36.206897
1    23.563218
3    16.666667
5    14.367816
4     9.195402
Name: Class, dtype: float64
```

Tiền xử lý

- Tách dữ liệu thành 3 tập train, validation, test với tỉ lệ lần lượt là 60%, 20%, 20%
- Vector input X: tất cả các cột trừ cột Class
- Output y: Cột class

Khám phá dữ liệu (tập huấn luyện)

- Các cột đang có kiểu dữ liệu phù hợp Bao gồm 3 cột category : Brand, OS
- **Với cột input dạng số, chúng phân bố như thế nào?**¶

	Screen	Main_Camera	Front_Camera	RAM	Storage	Battery
missing_ratio	0.00	11.70	21.6	22.5	21.6	0.0
min	1.77	0.08	2.0	1.0	4.0	800.0
lower_quartile	4.70	8.00	8.0	3.0	64.0	2113.5
median	6.40	13.00	10.0	4.0	128.0	4000.0
upper_quartile	6.50	48.00	16.0	8.0	128.0	5000.0
max	7.59	108.00	44.0	12.0	512.0	6000.0

- Tỉ lệ missing value của các cột dạng số < 30%

→ có thể xử lý và không cần bỏ cột nào.

Khám phá dữ liệu (tập huấn luyện)

- **Với các cột input không thể dạng số, các giá trị được phân bố như thế nào?**

	OS	Brand	CPU
missing_ratio	22.5	0	21.6
num_values	9	13	5
value_ratios	{'Android 10': 62.8, 'Android 9 ': 16.3, 'iOS 14': 12.8, 'Android 8 ': 2.3, 'Android 11': 1.2, 'KaiOS': 1.2, 'Android 10 ': 1.2, 'iOS 12': 1.2, 'EMUI 10 ': 1.2}	{'samsung': 15.3, 'nokia': 12.6, 'iphone': 10.8, 'oppo': 9.0, 'vivo': 9.0, 'xiaomi': 9.0, 'masstel': 8.1, 'vsmart': 8.1, 'realme': 8.1, 'mobell': 3.6, 'itel': 2.7, 'energizer': 2.7, 'huawei': 0.9}	{'Snapdragon': 43.7, 'MediaTek': 28.7, 'Apple': 13.8, 'Exynos': 11.5, 'Spreadtrum': 2.3}

Tỉ lệ missing value < 30%

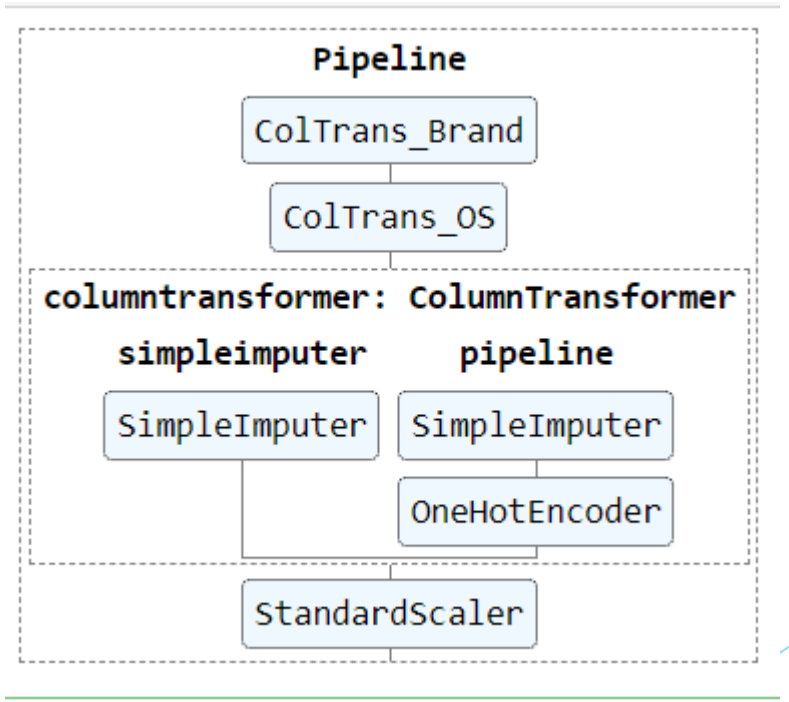
Nhận xét :cột Brand, OS có nhiều giá trị khác nhau.

Tiền xử lý
(tập huấn luyện)

- **1.Biến đổi giá trị các cột cho việc huấn luyện hiệu quả hơn:**
- Cột Brand: Lấy top_brand (ví dụ lấy 5 brand phổ biến nhất), những giá trị khác biến thành Others
- Cột OS: trước hết ta điền missing value trong cột với 'No Support'. Tiếp đến,ta nhận thấy có 2 giá trị khác nhau 'Android 10' và 'Android 10 ' do một giá trị bị dư khoảng trắng phía sau số 10, ta tiến hành chuyển 'Android 10 ' thành Android 10.
- Rồi lấy top_Operations như cột Brand.
- **2.Xử lý giá trị thiếu và chuyển cột không phải dạng số về dạng số:**
- Đối với những cột là dạng số:
  - Cột Screen & Battery không có giá trị thiếu.
  - Các cột 'Main_Camera','Front_Camera','RAM','Storage': các cột này bị thiếu chủ yếu do các mẫu điện thoại tương ứng ở phân khúc thấp không hỗ trợ tính năng đó. Vì vậy ta sẽ cho giá trị bằng 0.
- Đối với những cột không phải dạng số:
  - Cột Brand không có giá trị thiếu.
  - Cột CPU : với những giá trị thiếu ta sẽ cho là 'No Support' (Cột OS đã điền giá trị thiếu ở trên bước 1)

Tiền xử lý
(tập huấn luyện)

- Xây dựng Class ColTrans_Brand và class ColTrans_OS để thực hiện nhiệm vụ tiền xử lý các cột ở trên
- Xây dựng preprocess_pipeline



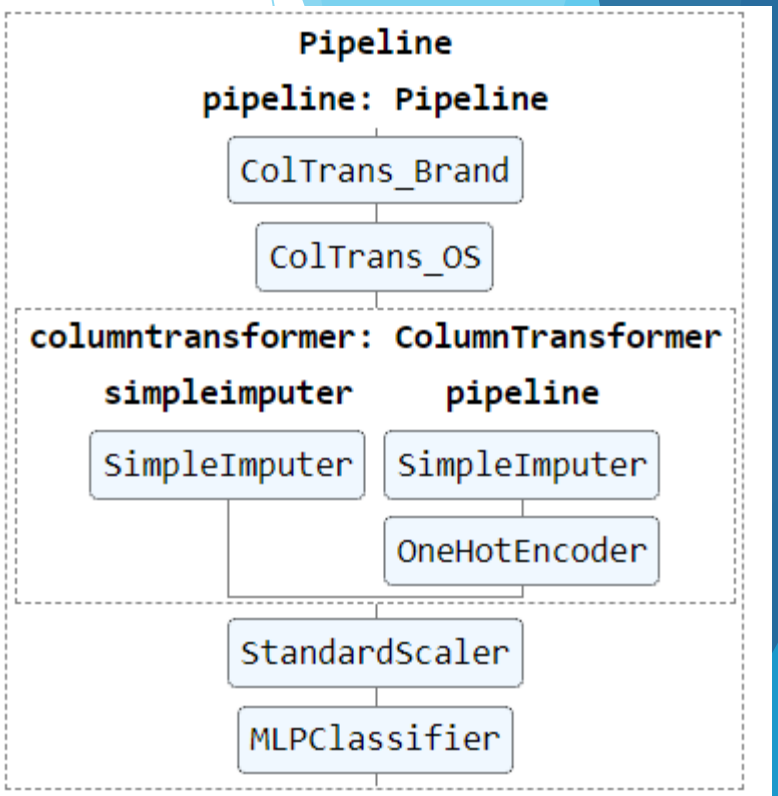
Tiền xử lý
(tập validation)

- Một khi đã có preprocess_pipeline với các giá trị (`top_titles_`, mean, mode, ...) đã được tính từ tập huấn luyện, ta có thể dễ dàng dùng phương thức transform để tiền xử lý cho các véc-tơ input mới trong tập validation và tập kiểm tra

Tiền xử lý và mô hình hóa

- Thực hiện transform trên tập **validation**
- Mô hình hóa:
  - Neural Netword
  - Multinomial Logistic Regression

Mô hình hóa
Neural Netword



- Thử nghiệm với mô hình Neural Netword với các tham số:

- hidden_layer_sizes= (20), activation='tanh',solver='lbfgs',random_state=0, max_iter=3000
- Siêu tham số của alpha với 5 giá trị khác nhau : 0.1, 1, 10, 100, 1000

  Siêu tham số num_top_s: 1,3,5,7,9,11

- Sau khi xây dựng mô hình, tạo full_pipeline chứa process_pipeline_full và mô hình hóa

0.0

Mô hình hóa
Neural Netword

- Thử nghiệm với mô hình Neural Netword
- Tìm được độ lỗi trên tập val best_val_err= 7.142857142857142
- best_alpha=0.1
- best_num_top_s = 3

Đánh giá mô hình tìm được

Độ lỗi trên tập test = 0 => Mô hình Neural NetWork tốt với độ lỗi trên tập test 0%.

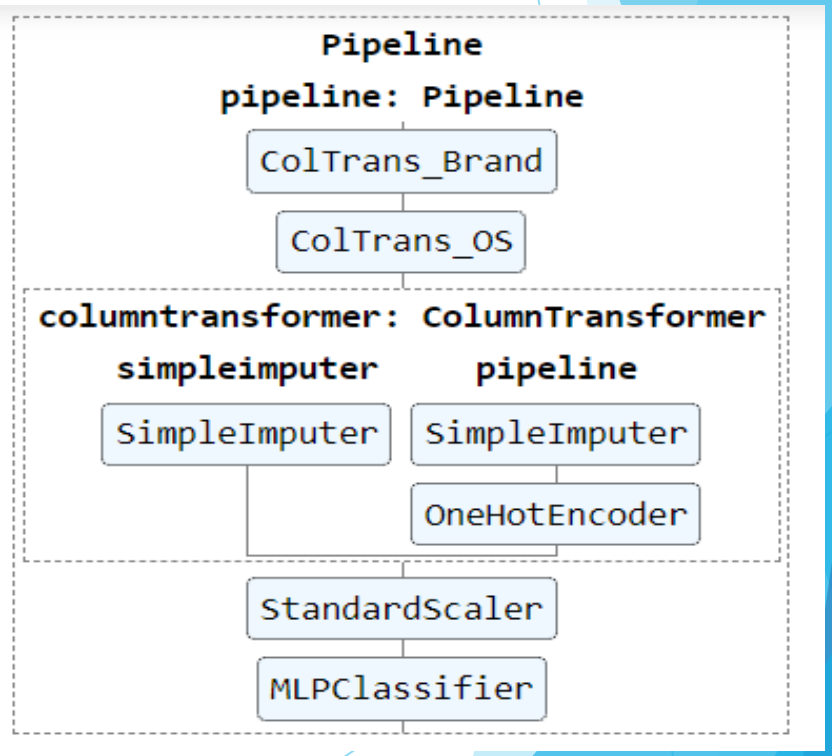
Độ lỗi trên tập test

```
((prediction != test_y_pr).mean())*100
```

```
0.0
```

Mô hình hóa
Multinomial Logistic Regression

- Thử nghiệm với mô hình với các tham số:
- multi_class='multinomial',max_iter=3000
- Siêu tham số num_top_s : 1,3,5,7,9,11
- Sau khi xây dựng mô hình, tạo full_pipeline
- chứa process_pipeline_full và mô hình hóa



Mô hình hóa
Multinomial Logistic Regression

- Thử nghiệm với mô hình :
- Tìm được độ lỗi trên tập validation là 14.285714285714285
- best_num_top_s_2=1

  Đánh giá mô hình tìm được:

  Độ lỗi trên tập test 2.85714285714857

➔ Mô hình Multinomial Logistic Regression vẫn cho kết quả rất tốt với độ lỗi trên tập test chỉ xấp xỉ 3%

Mô hình Decision Tree

- Thử nghiệm với mô hình với các tham số:
- Siêu tham số num_top_s : 1,3,5,7,9,11
- Sau khi xây dựng mô hình, tạo full_pipeline chứa process_pipeline_full và mô hình hóa
- Tìm được độ lỗi trên tập validation là 14.285714285714285
- Best_num_top= 1
- Đánh giá mô hình:
- Độ lỗi trên tập test 0.0

Đánh giá mô hình

- Độ lỗi trên tập validation :
  - Multinomial Logistic Regression=Decision Tree > Neural Network
- Độ lỗi trên tập test :
  - Multinomial Logistic Regression > Neural Network = Decision Tree

→ **Mô hình tốt nhất là Neural Network**

Nhìn lại quá trình

- Khó khăn:
- Thời gian đồ án trong thời gian thi HK1
- Khó khăn trong việc tiền xử lý dữ liệu
- Khó khăn về thu thập dữ phong phú,đa dạng , ít nhiễu, ít thô
- Hạn chế về matplotlib nên không thể trực quan hóa dữ liệu một cách sinh động.

- Những lợi ích mang lại:
- Học được cách tiền xử lý dữ liệu, khám phá dữ liệu, mô hình hóa, đánh giá mô hình hóa một cách tốt hơn
- Hiểu rõ hơn về quy trình Khoa học dữ liệu .

- Nếu có thêm thời gian nhóm em sẽ thu thập dữ liệu đa dạng phong phú hơn , thử với nhiều mô hình khác để độ lỗi trên tập test nhỏ nhất

Tài liệu tham khảo

- Các file bài tập và demo của thầy