

# Chủ đề

- > Dự đoán giá tiền laptop dựa trên thuộc tính có sẵn.
- > Lợi ích của chủ đề:
  - \* Chọn trước các mẫu phù hợp túi tiền rồi lấy cấu hình tính ra giá dự đoán và so sánh với giá niệm yết của cửa hàng.
  - ❖ Đưa ra những cấu hình mong muốn để dự đoán giá của máy tính rồi từ khoảng giá tiền đó tìm những mẫu máy tính có những thuộc tính gần với mong muôn.
- > Nguồn cảm hứng của chủ đề:
  - \* Việc chọn cách định giá một sản phẩm nào đó lấy cảm hứng từ các bài làm khác đã thấy qua.
  - \* Cảm hứng từ chính bản thân khi đã từng phân vân trong việc chọn mẫu laptop để mua và chưa có những cơ sở đủ mạnh để đưa ra quyết định.

- > Hình thức thu thập dữ liệu: web api.
- Link api: <a href="https://noteb.com/api/webservice.php">https://noteb.com/api/webservice.php</a>.
- Link api document: <a href="https://noteb.com/res/doc/noteb\_API\_documentation.pdf">https://noteb.com/res/doc/noteb\_API\_documentation.pdf</a>.
- Lý do chọn lựa web api: Cung cấp nhiều data laptop với nhiều loại thuộc tính khác nhau đa dạng.
- ➤ Hạn chế: Số lượt request trong 1 ngày là 1000 lần => Tốn thời gian thu thập trong nhiều ngày.

#### Method *get\_model\_info*

```
"code": 26,
"message": "Valid method.",
"result": {
 "0": {
   "model info": [
       "id": 4899,
       "noteb name": "Asus ROG Strix Scar 17 G732 LWS (US)",
       "name": "Asus ROG Strix Scar 17 G732",
       "extra_name": "",
       "submodel_info": ["LWS", "(US)"]
   "config id": "9621119275876599808",
   "model resources": {
     "thumbnail": "https://noteb.com/res/img/models/thumb/t_2630_1.jpg",
     "image 1": "https://noteb.com/res/img/models/2630 1.jpg",
     "image 2": "https://noteb.com/res/img/models/2630_2.jpg",
     "image_3": "https://noteb.com/res/img/models/2630_3.jpg",
     "image_4": "https://noteb.com/res/img/models/2630_4.jpg",
     "official link": "https://www.asus.com/Laptops/ROG-Strix-SCAR-15-17/",
     "official_link2": null,
     "launch_date": "2020-04-06",
     "primary model": "4471"
   "cpu": {
     "prod": "Intel",
     "model": "i7-10875H",
     "lithography": "14",
     "cache": "16",
     "base speed": "2.30",
     "boost_speed": "5.10",
     "cores": "8",
     "tdp": "45",
     "other info": "SSE4.2,AVX2.0,64-bit,HT,VT-d,VT-x,TBT 3.0",
     "rating": "71.1",
     "integrated video id": "447",
     "integrated_video": "Unknown"
```

```
"display": {
 "size": "17.3",
 "horizontal resolution": "1920",
 "vertical_resolution": "1080",
 "type": "LED IPS",
 "sRGB": "100",
 "touch": "no",
 "other info": "3ms response time"
"memory": { "size": "32", "speed": "3200", "type": "DDR4" },
"primary_storage": {
 "model": "SSD M.2 PCIe",
 "cap": "8192",
 "rpm": null,
 "read speed": "2040"
"secondary storage": {
 "model": "N/A",
 "cap": "0",
 "rpm": null,
 "read_speed": null
"gpu": {
 "prod": "Nvidia",
 "model": "GeForce RTX 2070 SUPER",
 "architecture": "Turing",
 "lithography": "12",
 "shaders": "2560",
 "base speed": "1140",
 "boost_speed": "1380",
 "shader speed": "1140",
 "memory speed": "1750",
 "memory bandwidth": "256",
 "memory size": "8192",
 "memory_type": "GDDR6",
 "tdp": "115",
 "other info": "Nvidia PhysX, Nvidia CUDA, Nvidia BatteryBoost, Optimus, Nvidia MFAA,
 "rating": "79.9"
```

Method *list\_models*: lấy ra tất cả mẫu laptop với param cho trước.

```
"code": 26,
"message": "Valid method.",
"result": {
   "0": {
        "model info": [
                "noteb_name": "Dell XPS 15 9560 UHD (US)",
                "name": "Dell XPS 15 9560",
                "extra_name": "",
                "submodel info": [
                   " UHD",
        "model resources": {
           "thumbnail": "https:\/\/noteb.com\/res\/img\/models\/thumb\/t_508_1.jpg",
           "image_1": "https:\/\/noteb.com\/res\/img\/models\/508_1.jpg",
           "image_2": "https:\/\/noteb.com\/res\/img\/models\/508_2.jpg",
           "image 3": "https:\/\/noteb.com\/res\/img\/models\/508 3.jpg",
           "image_4": "https:\/\/noteb.com\/res\/img\/models\/508_4.jpg",
           "official link": "http:\/\/www.dell.com\/en-us\/shop\/dell-laptops\/xps-15\/spd\/xps-15-9560-laptop",
           "official_link2": null,
           "launch_date": "2017-02-13",
           "primary model": "798"
   "1": {
        "model_info": [
               "noteb_name": "Dell Inspiron 13 5379 2-in-1 (US)",
               "name": "Dell Inspiron 13 5379 2-in-1",
               "extra_name": "",
                "submodel info": [
        "model resources": {
           "thumbnail": "https:\/\/noteb.com\/res\/img\/models\/thumb\/t_971_1.jpg",
            "image 1": "https:\/\/noteb.com\/res\/img\/models\/971 1.jpg",
            "image_2": "https:\/\/noteb.com\/res\/img\/models\/971_2.jpg",
            "image 3": "https:\/\/noteb.com\/res\/img\/models\/971 3.jpg",
```

15/1/2021 Nhóm 16

Method get\_model\_info\_all: từ 1 tên laptop lấy hiển thị nhiều cấu hình khác nhau.

```
"result": {
           "model info": [
                  "id": 838, -> noteb model id for this model
                  "noteb name": "Apple MacBook Pro 13 (H12017) Classi.",
              ... -> same as get_model_info, refer to it for more information
           "config id": "12244054780192000000",
           "model resources": {
              "image 1": "http://86.123.134.36/notebro/res/img/models/499 1.jpg",
              ... -> same as get_model_info, refer to it for more information
           "cpu": {
              "374": { -> noteb id for this cpu
                  "prod": "Intel",
                  "model": "i5-7360U",
                  "lithography": "14",
              ... -> same as cpu section from get model info, refer to it for more information
                  "integrated video id": 0, -> indicates the element number in the gpu object for
the integrated video solution associated with this processor
                  "integrated video": "Intel Iris Plus Graphics 640"
              "375": { -> noteb id for this cpu
                  "prod": "Intel",
                  "model": "i7-7660U",
              ... -> same as cpu section from get_model_info, refer to it for more information
                  "integrated video id": 1, -> indicates the element number in the gpu object for
the integrated video solution associated with this processor
                  "integrated video": "Intel Iris Plus Graphics 640"
              "selected": 374 -> indicates the element in the returned cpu object that is selected in
the retrieved configuration
           "display": {
              "27": { -> noteb id for this display
```

Method get\_conf\_info: lấy ra tất cả mẫu laptop với param cho trước.

#### Request example

```
["apikey"]=> "112233aabbcc",
["method"]=> "get conf info"
["param"]=>
       ["model id"]=> "838",
       ["cpu id"]=> "375",
       ["display id"]=> "27",
       ["memory id"]=> "18",
       ["primary storage id"]=> "29",
       ["secondary storage id"]=> "0",
       ["gpu id"]=> "500",
       ["wireless id"]=> "54",
       ["optical drive id"]=> "0",
       ["motherboard id"]=> "491",
       ["chassis id"]=> "497",
       ["battery id"]=> "219",
       ["warranty id"]=> "0",
```

#### Response example

```
"code": 29,
   "message": "Valid method. No valid configuration id provided, falling
back to component search.",
      "result": { ... -> this part is the same as the configuration part from the get_model_info, refer
to it for more information
      "model id": "1046",
      "config score": "45.76",
      "config price": "923",
      "config price min": "894",
      "config price max": "953",
      "battery life raw": "3",
      "battery life hours": "3:00",
      "total storage capacity": "128"
   "daily hits left": "470"
```

- Dữ liệu: 800 dòng, 23 cột.
- Các cột và ý nghĩa:

- 1. producer
- 2. processor prod
- 3. processor model
- 4. cores
- 5. core base speed (GHz)
- 6. core boost speed (GHz)
- 7. ram type
- 8. ram cap (GB)
- 9. ssd (GB)
- 10. hdd (GB)
- 11. gpu prod
- 12. gpu size (MB)
- 13. gpu base speed (GHz)
- 14. gpu boost speed (GHz)
- 15. screen type
- 16. screen size (inch)
- 17. creen horizontal resolution
- 18. screen vertical resolution
- 19. sRGB (%)
- 20. weight (kg)
- 21. os
- 22. battery capacity (Whr)
- 23. price(USD)

tên nhà sản xuất laptop

nhà sản xuất CPU

dòng CPU

số nhân của CPU

tốc độ xung nhịp cơ bản của CPU

tốc độ xung nhịp tối đa của CPU

loại RAM

dung lượng RAM

dung lượng SSD

dung lượng HDD

nhà sản xuất card màn hình

dung lượng card màn hình

tốc độ xung nhịp cơ bản của GPU

tốc độ xung nhịp tối đa của GPU

loai màn hình

kích thước màn hình

độ phân giải màn hình theo chiều ngang

độ phân giải màn hình theo chiều dọc

độ phủ màu của màn hình dựa trên tiêu chuẩn sRGB

trọng lượng máy

hệ điều hành

dung lượng pin

giá

- ➤ Output y: cột *price(USD)*.
- > Input vector X: các cột còn lại.
- Tách tập dữ liệu (thành 2 phần):
  - Tập Test: 50%.
  - ➤ Tập Train+Validation: 50%, chia tập này thành tập Train và Validation theo tỷ lệ 70% và 30%.
- Các kiểu dữ liệu của vector X:

producer	object
processor prod	object
processor model	object
cores	int64
core base speed (GHz)	float64
core boost speed (GHz)	float64
ram type	object
ram cap (GB)	int64
ssd (GB)	int64
hdd (GB)	int64
gpu prod	object
gpu size (MB)	int64
gpu base speed (GHz)	int64
gpu boost speed (GHz)	int64
screen type	object
screen size (inch)	float64
screen horizontal resolution	int64
screen vertical resolution	int64
sRGB (%)	int64
weight (kg)	float64
os	object
battery capacity (WHr)	float64

Các cột dữ liệu là số: tiến hành quan sát các giá trị missing\_ratio, lower\_quartile, median, upper\_quartile.

	cores	core base speed (GHz)	core boost speed (GHz)	ram cap (GB)	ssd (GB)	hdd (GB)	gpu size (MB)	gpu base speed (GHz)	gpu boost speed (GHz)	screen size (inch)	screen horizontal resolution	screen vertical resolution	sRGB (%)	weight (kg)	battery capacity (WHr)
missing_ratio	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00	0.0
min	2.0	1.0	3.0	4.0	0.0	0.0	128.0	200.0	500.0	12.3	1366.0	768.0	0.0	0.85	30.0
lower_quartile	4.0	1.6	3.9	8.0	0.0	256.0	1700.0	300.0	1100.0	14.0	1920.0	1080.0	60.0	1.40	45.0
median	4.0	2.1	4.2	8.0	0.0	256.0	2048.0	400.0	1200.0	15.6	1920.0	1080.0	90.0	1.80	52.5
upper_quartile	6.0	2.6	4.5	16.0	0.0	512.0	6144.0	1140.0	1468.0	15.6	1920.0	1080.0	100.0	2.10	66.2
max	8.0	3.0	5.1	128.0	2000.0	6144.0	16384.0	1531.0	1725.0	17.3	3840.0	2400.0	100.0	4.70	100.0

Các cột dữ liệu là chuỗi: tiến hành quan sát các giá trị missing\_ratio, num\_values, values\_ratio.

	producer	gpu prod	os	screen type	ram type	processor prod	processor model
missing_ratio	0	0	0	0	0	0	0
num_values	6	3	6	4	2	2	6
value_ratios	{'lenovo': 29.6, 'dell': 20.8, 'hp': 19.6, 'asus': 19.6, 'acer': 8.8, 'apple': 1.7}	{'intel': 52.1, 'nvidia': 34.6, 'amd': 13.3}	{'windows home 10.00': 60.8, 'windows pro 10.00': 33.3, 'linux ubuntu 0.00': 2.5, 'macos 11.00': 1.7, 'no os 0.00': 1.2, 'chrome os 0.00': 0.4}	{'led ips': 87.9, 'led tn': 10.0, 'oled': 1.7, 'led tn wva': 0.4}	{'ddr4': 96.7, 'ddr3': 3.3}	{'intel': 86.7, 'amd': 13.3}	{'i7': 43.3, 'i5': 36.7, 'ryzen5': 9.6, 'i3': 6.7, 'ryzen3': 2.1, 'ryzen7': 1.7}

- Tách <u>processor model</u> ra thành 2 cột: 1 cột gồm các <u>processor</u> của intel, 1 cột gồm các <u>processor</u> của amd; do cách tách này nên cột <u>processor prod</u> không còn cần thiết nữa.
- → Xây dựng function **split\_processor\_model** để tách process model thành <u>intel\_processor\_model</u>, <u>amd\_processor\_model</u>.
- ➤ 2 cột <u>producer</u> và <u>os</u> có nhiều giá trị khác nhau nên sẽ chọn các giá trị xuất hiện nhiều nhất theo <u>num\_top\_producer</u> và <u>num\_top\_os</u> các giá trị khác được thay bằng giá trị "others".
- → Xây dựng lớp **ColAdderDropper** để xử lý cho <u>num\_top\_producer</u> và <u>num\_top\_os.</u>

#### Hàm split\_processor\_model

```
# Hàm định nghĩa transformer tách giá trị của thuộc tính processor_model thành 2 cột:
# - 1 côt gồm các processor của intel
# - 1 cột gồm các processor của amd
# Và chuyển về dạng số bằng phương pháp ranking
# Đồng thời xóa 2 cột "processor prod" và "processor model"
def split_processor_model(X):
    intel_processor_rank = {'i3': 1,
                            'i5': 2,
                            'i7': 3,
                            'ryzen3': 0,
                            'ryzen5': 0,
                            'rvzen7': 0}
    intel processor model = X['processor model'].replace(intel processor rank)
    amd_processor_rank = {'i3': 0,
                          'i5': 0,
                          'i7': 0,
                          'ryzen3': 1,
                          'ryzen5': 2,
                          'ryzen7': 3}
    amd processor_model = X['processor model'].replace(amd_processor_rank)
    return X.assign(intel_processor_model=intel_processor_model, amd_processor_model=amd_processor_model).drop(["processor prod"
```

#### Lóp ColAdderDropper

```
# Hàm chọn các giá trị xuất hiện nhiều nhất theo "num top producer" đối với cột "producer" và "num top os" đối với cột "os"
# và các giá trị khác được thay bằng giá trị "others"
class ColAdderDropper(BaseEstimator, TransformerMixin):
    def __init__(self, num_top_producers=1, num_top_os=1):
       self.num top producers = num top producers
       self.num_top_os = num_top_os
   def fit(self, X_df, y=None):
        producer_col = X_df.producer
       self.producer_counts_ = producer_col.value_counts()
       producers = list(self.producer_counts_.index)
       self.top_producers_ = producers[:max(1, min(self.num_top_producers, len(producers)))]
       os col = X df.os
       self.os counts = os col.value counts()
       os = list(self.os_counts_.index)
       self.top os = os[:max(1, min(self.num top os, len(os)))]
        return self
   def transform(self, X_df, y=None):
       X = X df.copy()
       X.loc[:, "producer"].replace(list(set(X.producer.unique())-set(self.top producers )), 'others', inplace=True)
       X.loc[:, "os"].replace(list(set(X.os.unique())-set(self.top_os_)), 'others', inplace=True)
        return X
```

### Xây dựng pipeline

- > FunctionTransformer cho hàm split\_processor\_model.
- > ColAdderDropper.
- SimpleImputer cho các thuộc tính định danh không có thứ tự với chiến thuật *most\_frequent* để điền giá trị thiếu, sau đó dùng **OneHotEncoder**.
- > SimpleImputer cho các thuộc tính số với chiến thuật *mean* để điền giá trị thiếu.
- ➤ **SimpleImputer** cho các thuộc tính số có thứ tự với chiến thuật *most\_frequent* để điền giá trị thiếu.
- > StandardScaler để chuẩn hóa các giá trị về khoảng chuẩn.
- > Sử dụng **SGDRegressor** để train model.

### Mô hình SGDRegressor

- > Điểm mạnh: hội tụ nhanh.
- Thuật toán lấy khoảng 10 epochs đi qua N điểm dữ liệu rồi dùng quy tắc cập nhật weight cho mỗi điểm. Sau mỗi epoch, data được shuffle để đảm bảo tính ngẫu nhiên.
- ➤ Mô hình dùng penalty='11' và tiến hành thử nghiệm giá trị alpha tốt nhất cho mô hình với các giá trị: 0.01, 0.1, 1, 10, 100.

### Mô hình SGDRegressor

- > Kết quả tốt nhất thu được:
  - alpha = 10.
  - best\_num\_top\_producers = 3.
  - best\_num\_top\_os = 2.
  - Độ lỗi trên val: 28.48%.
  - Độ lỗi trên test: 32.99%.

```
full pipeline = make pipeline(FunctionTransformer(split processor model),
                                  ColAdderDropper(num top producers=4, num top os=4),
                                  column transformer, StandardScaler(),
                                  SGDRegressor(penalty='l1', random state=0))
 6 train_errs = []
 7 | val errs = []
 8 alphas = [0.01, 0.1, 1, 10, 100]
 9 num top producers s = range(1, 8)
10 num top os s = range(1, 7)
best_val_err = float('inf'); best_num_top_os = None; best_num_top_producers = None;
12 for alpha in alphas:
        for num top producers in num top producers s:
13
14
           for num top os in num top os s:
15
               full pipeline.set params(coladderdropper num top producers=num top prod
16
               full pipeline.fit(train X df, train y sr)
17
                train errs.append(round(100 - compute rr(train y sr, full pipeline.predi
18
                val errs.append(round(100 - compute rr(val y sr, full pipeline.predict(v
19
               if val errs[-1] < best val err:
20
                    best val err = val errs[-1]
                    best_alpha = alpha
21
22
                    best num top producers = num top producers
                    best num top os = num top os
```

# Nhìn lại quá trình làm đồ án

#### Những khó khăn đã gặp phải:

- > Trong quá trình tìm chủ đề: phải chọn được chủ đề phù hợp với khả năng.
- Trong quá trình thu thập dữ liệu: khó khăn trong việc tìm kiếm API và lấy dữ liệu tự động từ API số lượt request tối đa mỗi ngày chỉ có 1000 lần dẫn đến tốn thời gian trong việc thu thập dữ liệu.
- Trong quá trình tiền xử lý: không có khó khăn.
- Trong quá trình mô hình hóa: Khó khăn trong việc lựa chọn mô hình và các tham số cho mô hình cần phải tìm hiểu để chọn cho hợp lý.

### Nhìn lại quá trình làm đồ án

#### Những gì đã học được:

- > Hiểu rõ thêm về quy trình thu thập, xử lý và mô hình hóa dữ liệu.
- > Học được cách làm việc nhóm trên Github.
- > Cách code để thu thập dữ liệu tự động từ API.
- > Hiểu thêm về các thuật toán mô hình hóa và các tham số.
- ➤ Không phải model có độ lỗi trên tập validation thấp thì sẽ tốt khi chạy trên tập kiểm tra.

# Nhìn lại quá trình làm đồ án

### Những cải thiện nếu có thêm thời gian:

- Cải thiện mô hình:
  - ❖ Thu thập thêm dữ liệu.
  - ❖ Loại những mẫu dữ liệu nhiễu.
  - Tìm hiểu thêm để thêm hoặc xóa bỏ các thuộc tính không cần thiết.
  - Tìm hiểu thêm về các thuật toán mô hình hóa để lựa chọn và đặt các tham số phù hợp với bài toán hơn.
- > Viết slide báo cáo hoàn chỉnh hơn.
- > TÌm hiểu thêm về Github để làm việc nhóm hiệu quả hơn.

### Tài liệu tham khảo

- 1. scikit-learn.org.
- 2. Slide bài giảng môn "Nhập môn Khoa học Dữ liệu" của thầy Nguyễn Trần Trung Kiên.
- 3. Các bài tập và các file Demo môn "Nhập môn Khoa học Dữ liệu" của thầy Nguyễn Trần Trung Kiên.

# Cám ơn thầy đã lắng nghe và góp ý cho tụi em