Nhập môn khoa học dữ liệu

Đồ án cuối kỳ

Nhóm 16

Thành viên:

18120058 - Phạm Công Minh

18120090 - Phạm Nguyên Minh Thy

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 niêm 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 muốn.
- *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.

Thu thập dữ liệu

- *Hình thức thu thập dữ liệu*: web api.
- *Link api*: https://noteb.com/api/webservice.php.
- *Link api document*: https://noteb.com/res/doc/noteb_API_documentation.pdf.
- *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.

Thu thập dữ liệu

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,N
        "rating": "79.9"
      },
```

Thu thập dữ liệu

Method *list_models*: lấy ra tất cả mẫu laptop với param cho trước.

```
r = requests.post('https://noteb.com/api/webservice.php',
                  data={'apikey': "112233aabbcc",
                        'method': "list_models",
                        'param[model_id]': "",
                        'param[model_name]': "Dell"})
```


```
{
    "code": 26,
    "message": "Valid method.",
    "result": {
        "0": {
            "model_info": [
                {
                    "id": 798,
                    "noteb_name": "Dell XPS 15 9560  UHD (US)",
                    "name": "Dell XPS 15 9560",
                    "extra_name": "",
                    "submodel_info": [
                        " UHD",
                        "(US)"
                    ]
                }
            ],
            "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": [
                {
                    "id": 1178,
                    "noteb_name": "Dell Inspiron 13 5379 2-in-1 (US)",
                    "name": "Dell Inspiron 13 5379 2-in-1",
                    "extra_name": "",
                    "submodel_info": [
                        "(US)"
                    ]
                }
            ],
            "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",
```

Thu thập dữ liệu

Method *get_model_info_all*: từ 1 tên laptop lấy hiển thị nhiều cấu hình khác nhau.

```
"result": {
   "0": {
      "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
```

Thu thập dữ liệu

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"
}
```

Tiền xử lý dữ liệu

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

1. producer	tên nhà sản xuất laptop
2. processor prod	nhà sản xuất CPU
3. processor model	dòng CPU
4. cores	số nhân của CPU
5. core base speed (GHz)	tốc độ xung nhịp cơ bản của CPU
6. core boost speed (GHz)	tốc độ xung nhịp tối đa của CPU
7. ram type	loại RAM
8. ram cap (GB)	dung lượng RAM
9. ssd (GB)	dung lượng SSD
10. hdd (GB)	dung lượng HDD
11. gpu prod	nhà sản xuất card màn hình
12. gpu size (MB)	dung lượng card màn hình
13. gpu base speed (GHz)	tốc độ xung nhịp cơ bản của GPU
14. gpu boost speed (GHz)	tốc độ xung nhịp tối đa của GPU
15. screen type	loại màn hình
16. screen size (inch)	kích thước màn hình
17. creen horizontal resolution	độ phân giải màn hình theo chiều ngang
18. screen vertical resolution	độ phân giải màn hình theo chiều dọc
19. sRGB (%)	độ phủ màu của màn hình dựa trên tiêu chuẩn sRGB
20. weight (kg)	trọng lượng máy
21. os	hệ điều hành
22. battery capacity (Whr)	dung lượng pin
23. price(USD)	giá

Tiền xử lý dữ liệu

	producer	processor prod	processor model	cores	core base speed (GHz)	core boost speed (GHz)	ram type	ram cap (GB)	ssd (GB)	hdd (GB)	...	gpu boost speed (GHz)	screen type	screen size (inch)	screen horizontal resolution	screen vertical resolution	sRGB (%)	weight (kg)	os	battery capacity (WHr)	price(USD)
299	dell	intel	i5	4	1.6	4.2	ddr3	8	0	256	...	1100	led ips	13.3	1920	1080	90	1.24	windows home 10.00	45.0	723.0
500	hp	intel	i7	4	1.8	4.9	ddr4	8	0	512	...	1150	led ips	15.6	1920	1080	60	2.05	windows home 10.00	53.2	885.5
303	dell	intel	i5	4	2.4	4.2	ddr4	8	0	256	...	1300	led ips	14.5	2560	1600	100	1.26	windows home 10.00	52.0	730.0
40	acer	intel	i7	4	1.8	4.6	ddr3	8	0	256	...	1468	led ips	14.0	1920	1080	100	1.35	windows home 10.00	48.0	949.0
495	hp	intel	i7	4	1.8	4.6	ddr4	8	0	256	...	1150	led tn	17.3	1600	900	80	2.45	windows home	41.0	830.0

Tiền xử lý dữ liệu

- 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
```

Tiền xử lý dữ liệu

- 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

Tiền xử lý dữ liệu

- 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}

Tiền xử lý dữ liệu

- Tách *processor model* ra thành 2 cột: 1 cột gồm các *processor* của intel, 1 cột gồm các *processor* của amd; do cách tách này nên cột *processor prod* không còn cần thiết nữa.

→ Xây dựng function **split_processor_model** để tách process model thành *intel_processor_model*, *amd_processor_model*.

- 2 cột *producer* và *os* có nhiều giá trị khác nhau nên sẽ chọn các giá trị xuất hiện nhiều nhất theo *num_top_producer* và *num_top_os* các giá trị khác được thay bằng giá trị *"others"*.

→ Xây dựng lớp **ColAdderDropper** để xử lý cho *num_top_producer* và *num_top_os.*

Tiền xử lý dữ liệu

Hàm split_processor_model

```
# Hàm định nghĩa transformer tách giá trị của thuộc tính processor_model thành 2 cột:
# - 1 cột 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,
                            'ryzen7': 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",
```

Tiền xử lý dữ liệu

Lớp 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='l1' 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 = 4.
  - best_num_top_os = 5.
  - Độ lỗi trên val: 30.12%.
  - Độ lỗi trên test: 32.98%.

```
# Hàm tính độ đo r^2
def compute_mse(y, preds):
    return ((y - preds) ** 2).mean()
def compute_rr(y, preds, baseline_preds):
    return 1 - compute_mse(y, preds) / compute_mse(y, baseline_preds)
baseline_preds = train_y_sr.mean()
```

```
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))

train_errs = []
val_errs = []
alphas = [0.01, 0.1, 1, 10, 100]
num_top_producers_s = range(1, 8)
num_top_os_s = range(1, 7)
best_val_err = float('inf'); best_num_top_os = None; best_num_top_producers = None;
for alpha in alphas:
    for num_top_producers in num_top_producers_s:
        for num_top_os in num_top_os_s:
            full_pipeline.set_params(coladderdropper__num_top_producers=num_top_prod
            full_pipeline.fit(train_X_df, train_y_sr)
            train_errs.append(round(100 - compute_rr(train_y_sr, full_pipeline.predi
            val_errs.append(round(100 - compute_rr(val_y_sr, full_pipeline.predict(v
            if val_errs[-1] < best_val_err:
                best_val_err = val_errs[-1]
                best_alpha = alpha
                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