BÁO CÁO ĐỒ ÁN CUỐI KỲ

Môn: Nhập môn Khoa học dữ liệu

Nhóm 26
Nguyễn Đức Trực - 18120621
Nguyễn Trần Trung - 18120625

GV hướng dẫn: Trần Trung Kiên

Mục lục

1. Chủ đề

- Câu hỏi: Sử dụng các mô hình máy học để dự đoán giá cả laptop trên thị trường theo các thông tin cấu hình từ máy.
- Ý nghĩa:
  - Tham khảo giá máy tính mới chưa ra thị trường.
  - Giúp người mua xem xét giá tiền có đúng với cấu hình không.
- Nguồn cảm hứng: Dựa trên chính nhu cầu của hầu hết sinh viên khi lên đại học.

2. Thu thập dữ liệu

- Dữ liệu được thu thập từ: Phong Vũ (phongvu.vn)
- Hình thức thu thập: parse html.
- Kết quả: được 794 dòng dữ liệu. (Tương ứng với thông tin của laptop)

2. Thu thập dữ liệu

Mỗi dòng dữ liệu bao gồm:

1. Thương hiệu
2. Bảo hành
3. Màu sắc
4. Series laptop
5. Part-number
6. Thế hệ CPU
7. CPU
8. Chip đồ họa
9. RAM
10. Màn hình
11. Lưu trữ
12. Số cổng lưu trữ tối đa
13. Kiểu khe M.2 hỗ trợ
14. Cổng xuất hình
15. Cổng kết nối
16. Kết nối không dây
17. Bàn phím
18. Hệ điều hành
19. Kích thước
20. Pin
21. Khối lượng
22. Đèn LED trên máy
23. Phụ kiện đi kèm
24. Bảo mật
25. Ổ đĩa quang
26. Tính năng
27. Tên sản phẩm
28. Mã SKU
29. Giá

3. Khám phá dữ liệu

```
: train_X_df.dtypes

: Title           object
  Brand           object
  Warranty         int64
  Color           object
  SeriesLaptop    object
  PartNum         object
  CPUgen          object
  CPU             object
  GraphicChip     object
  RAM             object
  Screen          object
  Storage         object
  MaxStoPortNum   object
```

CPUgen	CPU	GraphicChip	...	K
Core i3 , Intel Core thế hệ thứ 10	Intel Core i3-1005G1 (1.2 GHz - 3.4 GHz / 4MB...	Intel UHD Graphics	...	s
Core i5 , Intel Celeron 1000	Intel Core i5-10300H (2.5 GHz - 4.5 GHz / 8MB...	NVIDIA GeForce GTX 1650Ti 4GB GDDR6 / Intel ...	...	

- Nhận xét về tập dữ liệu
  - Dữ liệu có 28 thuộc tính.
  - Một số thuộc tính có kiểu dữ liệu chưa phù hợp.
- VD: RAM, Storage,Screen,...
  - Cần trích xuất các dữ liệu quan trọng như CPUgen, GraphicChip,…

4. Tiền xử lý dữ liệu

- Đầu tiên ta sẽ thực hiện bước tiền xử lý là tách tập kiểm tra, validation và tập test ra theo tỉ lệ: 70%:15%:15%.
- Loại bỏ nhiều cột có nhiều giá trị khác nhau hoặc ít ảnh hưởng đến giá thành.
- Rút trích các dữ liệu chính từ các cột và thay thế chúng và chuyển các cột thành các kiểu dữ liệu phù hợp.

4. Tiền xử lý dữ liệu

Brand	GraphicChip	RAM	Screen	Pin	Weight	Security	CPUs	chipCPU	gen	SSD
ASUS	Intel	8	13.3	3.0	1.3	Yes	Core i5	U	10.0	512
ACER	AMD	8	15.6	2.0	1.7	No	Ryzen 3	U	3.0	256
Dell	Intel	4	14.0	4.0	2.0	No	Core i3	U	8.0	1000
ASUS	NVIDIA	8	15.6	3.0	2.2	No	Core i5	H	8.0	1000
HP	NVIDIA	8	15.6	3.0	2.2	No	Core i7	H	9.0	512

```
Brand          object
GraphicChip    object
RAM             int64
Screen        float64
Pin           float64
Weight        float64
Security       object
CPUs           object
chipCPU        object
gen            float64
SSD             int64
dtype: object
```

Dữ liệu đã được làm sạch

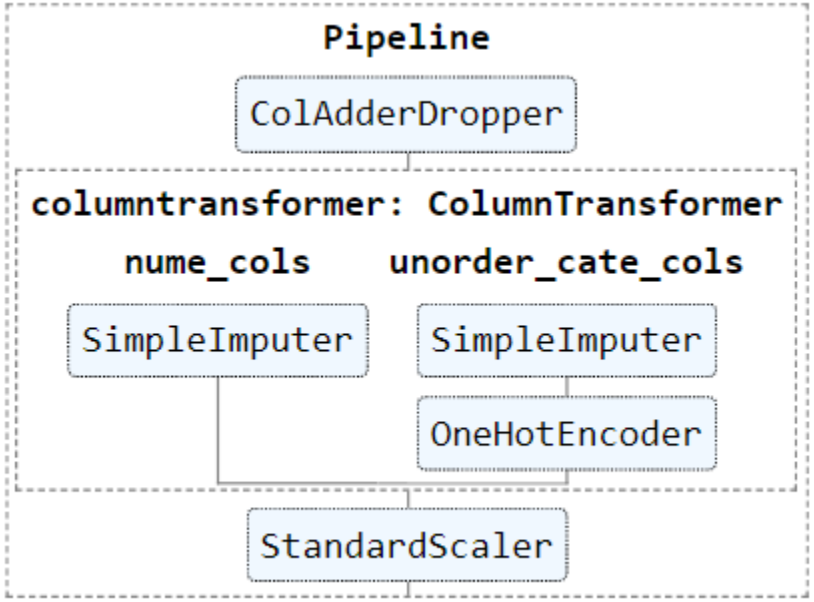
4. Tiền xử lý dữ liệu

```
---CPU
Core i5     261
Core i3     112
Core i7      97
Ryzen 5      21
Ryzen 3      17
Ryzen 7      11
Pentium       7
Celeron       6
Core i9       2
Name: 0, dtype: int64

---chipCPU
U    286
G    117
H    111
N     13
i     11
Y      1
Name: 0, dtype: int64
```

- Xây dựng class ColDropper kế thừa từ class BaseEstimator và TransformerMixin để thực hiện nhiệm vụ tiền xử lý.
- Các cột chứa nhiều giá trị ảnh hưởng đến train như CPU và chipCPU cần lấy nhiều giá trị khác nhau.
  - num_top_cpus
  - num_top_chipCPUs

4. Tiền xử lý dữ liệu



- Sử dụng các phương thức trong sklearn:
  - SimpleImputer, OneHotEncoder: Điền dữ liệu thiếu và chuyển về dạng số.
  - StandardScaler: Chuẩn hóa dữ liệu.
  - Pipeline.

5. Huấn luyện các mô hình

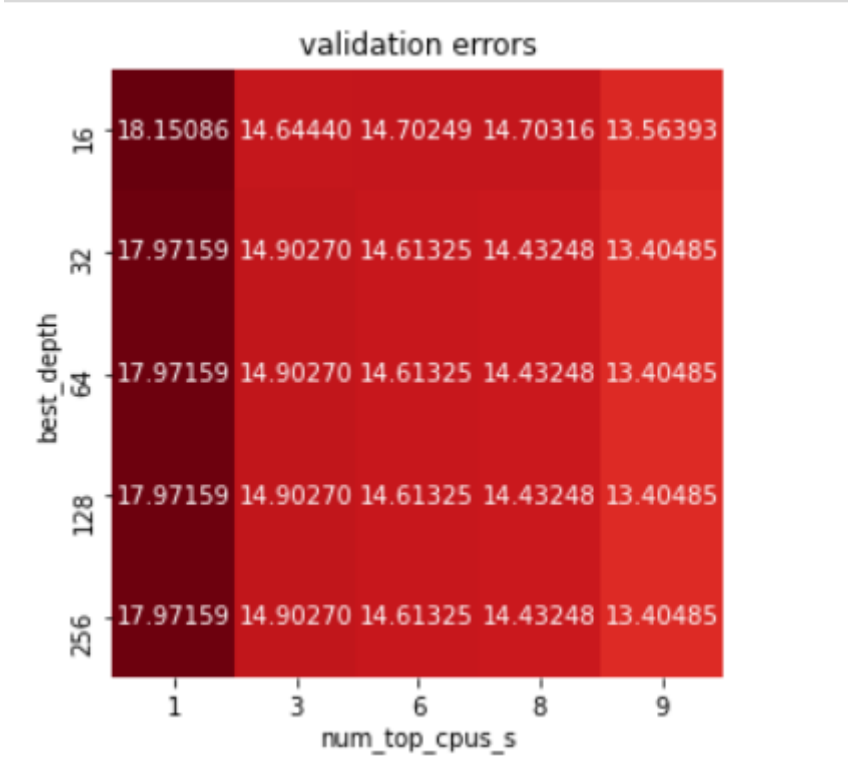
```
# Tính độ đo r^2 trên tập huấn luyện
def compute_mse(y, preds):
    return ((y - preds) ** 2).mean()
def compute_rr(y, preds, baseline_preds):
    return 1 - compute_mse(y, preds) / compute_mse(y, bas
baseline_preds = train_y_sr.mean()
```

- Sử dụng độ do **R-Squared** cho mô hình hồi quy.
- Đánh giá lần lượt các mô hình:
  - Mô hình SGDRegressor
  - Mô hình RandomforestRegressor

Mô hình SGDRegressor

- Tham số: random_state=0.
- Siêu tham số alpha với 5 giá trị khác nhau:
  [0.01, 0.1, 1, 10, 100]
- Tham số num_top_cpus với 6 giá trị khác nhau:
  [1, 3, 5, 6, 8, 9]
- Độ lỗi nhỏ nhất trên tập validation là 15,2%.

Mô hình RandomforestRegressor



- Tham số: random_state=0.
- Tham số num_top_cpus với 6 giá trị khác nhau: [1, 3, 5, 6, 8, 9]
- Độ lỗi nhỏ nhất trên tập valida tion là 13,4%.

6. Đánh giá và chạy mô hình trên tập test

- Cả mô hình Randomforest Regression và SGD Regression cho kết quả khả quan trên tập validation, tuy nhiên kết quả vẫn còn khá chủ quan vì việc lựa chọn các siêu tham số đều được làm bằng tay.
- Cả 2 mô hình đều chạy khá nhanh, nhóm em chọn mô hình Randomforest Regression vì kết quả có vẻ tốt hơn.
- Độ chính xác trên tập test: `0.7844`.

7. Nhìn lại quá trình làm đồ án

- Khó khăn:
  - Khó khăn trong việc lấy dữ liệu từ các trang web bán laptop.
  - Dữ liệu parse được không nhiều.
  - Khó khăn trong việc tìm hiểu các siêu tham số cho mô hình.
  - Khó khăn trong việc chọn các thuộc tính, đặc trưng phù hợp để train.
  - Thời gian hạn chế vì đồ án diễn ra trong thời gian thi cử.

7. Nhìn lại quá trình làm đồ án

- Những điều hữu ích học được:
  - Kĩ năng làm việc nhóm.
  - Kĩ năng sử dụng các công cụ hỗ trợ làm đồ án (như github, trello,...).
  - Các kĩ năng khám phá dữ liệu.
  - Tìm hiểu được thêm nhiều mô hình máy học hay.
  - Hiểu sâu sắc hơn quy trình Khoa học dữ liệu qua việc tự tìm hiểu và làm đồ án.

7. Nhìn lại quá trình làm đồ án

- Những dự định nếu có thời gian thêm:
  - Thêm phần phân tích tương quan dữ liệu để chọn thuộc tính, đặc trưng phù hợp hơn.
  - Tìm hiểu kỹ hơn các mô hình hiện tại cũng như tìm hiểu thêm các mô hình khác để đưa ra các siêu tham số tối ưu hơn.
  - Tiền xử lý dữ liệu sạch hơn.
  - Chuẩn bị slide báo cáo hoàn chỉnh hơn.

8. Tài liệu tham khảo

- Laptop Prices Prediction | Kaggle
- A Complete Tutorial which teaches Data Exploration in detail (analyticsvidhya.com)
- sklearn.linear_model.SGDRegressor — scikit-learn 0.24.0 documentation (scikit-learn.org)
- sklearn.ensemble.RandomForestRegressor — scikit-learn 0.24.0 documentation (scikit-learn.org)

THANKS FOR WATCHING