



KHOA HỌC DỮ LIỆU

ĐỒ ÁN CUỐI KÌ

DỰ ĐOÁN GIÁ ĐIỆN THOẠI

Nhóm 14:

Lê Hoàng Phương Nhi – 18120496

Lê Thị Như Quỳnh – 18120530



NỘI DUNG

- Giới thiệu đồ án
- Thu thập dữ liệu
- Khám phá dữ liệu
- Tiền xử lý dữ liệu

- Mô hình hóa
- Đánh giá kết quả
- Nhìn lại quá trình
- Tài liệu tham khảo

Giới thiệu đề án

- Câu hỏi: Dự đoán khoảng giá (thấp, trung bình, cao) của điện thoại dựa trên các thuộc tính đặc trưng của điện thoại?
 - ✓ Input: Các đặc trưng của điện thoại
 - ✓ Output: Khoảng giá của điện thoại
- Ý nghĩa thực tế:
 - ✓ Đối với người tiêu dùng: Đem lại các thông tin cần thiết cho người muốn mua điện thoại (ví dụ: khi người tiêu dùng muốn mua điện thoại có cấu hình nào đó thì mức giá sẽ rơi vào khoảng bao nhiêu)
 - ✓ Đối với nhà sản xuất: Cân nhắc để đưa ra được mức giá phù hợp cho điện thoại chuẩn bị tung ra thị trường
- Nguồn cảm hứng: Nhóm em tự nghĩ

Thu thập dữ liệu

Dữ liệu thu thập trên trang: <https://phonesdata.com/en/>

Dữ liệu thu thập trên trang là hợp pháp:

```
rp = urllib.robotparser.RobotFileParser()
rp.set_url('https://phonesdata.com/robots.txt')
rp.read()
rp.can_fetch('*', 'https://phonesdata.com/en/smartphones/')
True
```

POPULAR SMARTPHONES



NOKIA X6
(2018)



APPLE
IPHONE XS



HUAWEI P20
LITE



LENOVO Z5
PRO



ALCATEL
IDOL 5

Khám phá dữ liệu

Dữ liệu bao gồm 5677 dòng, 30 cột (chưa tiền xử lý)

- 'brand': hãng điện thoại
- 'technology': công nghệ màn hình
- 'touch_screen': loại màn hình
- 'display_colors': số màu của màn hình
- 'screen_size': kích thước màn hình (inch)

Khám phá dữ liệu

Dữ liệu bao gồm 5677 dòng, 30 cột (chưa tiền xử lý)

- 'screen_area': diện tích màn hình
- 'screen_to_body_ratio': tỉ lệ màn hình hiển thị
- 'screen_resolution': độ phân giải màn hình
- 'rear_camera': thông tin về camera sau
- 'front_camera': thông tin về camera trước

Khám phá dữ liệu

Dữ liệu bao gồm 5677 dòng, 30 cột (chưa tiền xử lý)

- 'os': hệ điều hành của điện thoại
- 'gpu': thông tin về chip xử lý đồ họa
- 'external_memory': thông tin về bộ nhớ ngoài
- 'internal_memory': thông tin về bộ nhớ trong (Ram, rom)
- 'cpu': thông tin về cpu

Khám phá dữ liệu

Dữ liệu bao gồm 5677 dòng, 30 cột (chưa tiền xử lý)

- 'dimensions': thông tin về chiều dài, chiều rộng, độ dày của điện thoại
- 'weight': trọng lượng điện thoại
- 'battery': thông tin về pin
- 'approximate_price': giá của điện thoại
- 'net_work': thông tin về mạng của điện thoại (2g, 3g, 4g, 5g)

Khám phá dữ liệu

Dữ liệu bao gồm 5677 dòng, 30 cột (chưa tiền xử lý)

- 'speed': tốc độ mạng điện thoại
- 'gprs': điện thoại có hỗ trợ gprs không (giao thức kết nối internet)
- 'gps': điện thoại có hỗ trợ gps không (định vị vị trí)
- 'wifi': điện thoại có hỗ trợ kết nối wifi không
- 'nfc': điện thoại có hỗ trợ kết nối nfc không

Khám phá dữ liệu

Dữ liệu bao gồm 5677 dòng, 30 cột (chưa tiền xử lý)

- 'usb': điện thoại có hỗ trợ ô cắm usb không
- 'bluetooth': thông tin về bluetooth của điện thoại
- 'radio': điện thoại có hỗ trợ đài radio không
- 'headphone_jack': điện thoại có hỗ trợ jack cắm tai nghe không
- 'sim_card': thông tin về sim

Khám phá dữ liệu

Dữ liệu có những vấn đề sau:

- Dữ liệu còn có các dòng bị lặp
- Dữ liệu còn chứa các giá trị thiếu
- Một số cột dữ liệu lộn xộn, khó khai thác
- Một số cột chưa đúng định dạng kiểu dữ liệu

Tiền xử lý dữ liệu

- Có những điện thoại không có thông tin về giá, xóa những dòng đó:

```
data_phone = data_phone[data_phone.approximate_price != 'None']
```

- Xóa các dòng bị trùng lặp:

```
phones_df = phones_df.drop_duplicates()
```

Tiền xử lý dữ liệu – Những thuộc tính chưa đúng định dạng

- `display_colors`: có nhiều loại đơn vị → tách số và chuyển về cùng đơn vị là M (triệu màu)
- `screen_size`, `screen_area`, `screen_to_body_ratio`, `rear_camera`, `front_camera`, `weight`: tách lấy số
- `screen_resolution`: tách lấy 2 số và nhân lại với nhau
- `os`: tách lấy tên hệ điều hành và phiên bản hệ điều hành
- `gpu`: tách lấy tên chip xử lý đồ họa
- `external_memory`: tách làm 2 cột
 - ✓ `external_memory`: điện thoại có hỗ trợ bộ nhớ ngoài không
 - ✓ `external_memory_size`: dung lượng bộ nhớ ngoài (lấy số, chuẩn hóa về cùng đơn vị là GB)

Tiền xử lý dữ liệu – Những thuộc tính chưa đúng định dạng

- `internal_memory`: tách lấy số dung lượng của RAM (chuẩn hóa về cùng đơn vị là GB)
- `cpu`: tách lấy số (tốc độ xử lý của cpu) (chuẩn hóa về cùng đơn vị là GHz)
- `dimensions`: tách lấy 3 số tương ứng với 3 cột
 - ✓ `length`: chiều dài
 - ✓ `width`: chiều rộng
 - ✓ `high`: độ dày
- `battery`: tách làm 3 cột
 - ✓ `battery_mah`: tách lấy số, thể hiện dung lượng pin
 - ✓ `battery_removable`: pin rời hay pin liền
 - ✓ `battery_type`: loại pin

Tiền xử lý dữ liệu – Những thuộc tính chưa đúng định dạng

- `approximate_price`: tách lấy số (chuẩn hóa về cùng đơn vị là Euro), sau đó chia khoảng cho giá
- `net_work`: tách ra thành 4 cột (2g, 3g, 4g, 5g)
- `radio, gprs, gps, wifi, usb, bluetooth`: dòng nào 'No' → 0, còn lại → 1 (không xét các dòng thiếu giá trị)
- `headphone_jack`: dòng nào 'No, included adaptor for 3.5mm', 'No' → 0, còn lại → 1 (không xét các dòng thiếu giá trị)

Tiền xử lý dữ liệu

- Xóa các cột:
 - ✓ 'sim_card': dữ liệu lộn xộn khó xử lý
 - ✓ 'dimensions', 'battery': đã tách ra các cột khác nên xóa cột gốc
 - ✓ 'approximate_price', 'price', 'EU', 'USD', 'INR', 'BTC', '€', '₹': các cột phát sinh khi xử lý giá tiền
 - ✓ 'gprs', 'gps', 'wifi', 'usb', 'bluetooth', '_2G', '_3G', '_5G', 'touch_screen': chênh lệch giữa các giá trị thuộc tính lớn
 - ✓ 'speed': chỉ cần net_work là đủ rồi
 - ✓ 'nfc': thiếu quá nhiều dữ liệu
 - ✓ 'external_memory_size': thiếu quá nhiều dữ liệu

Tiền xử lý dữ liệu

- Sau khi vẽ heatmap để xem xét sự tương quan giữa các thuộc tính thì nhóm loại tiếp các thuộc tính: 'high', 'display_colors', 'headphone_jack', 'radio'
- Class ColAdderDropper: Để xóa thuộc tính và gán lại giá trị cho thuộc tính 'brand', 'technology', 'os'
- num_mean_cols: điền giá trị thiếu bằng giá trị trung bình của thuộc tính
- num_median_cols: điền giá trị thiếu bằng giá trị trung vị của thuộc tính
- unordered_cate_cols: điền giá trị thiếu bằng giá trị phổ biến nhất của thuộc tính sau đó sử dụng phương pháp one-hot để chuyển dữ liệu thành số

Mô hình hóa

Nhóm lựa chọn 3 mô hình: RandomForestClassifier, SVC (Support vector machines), MLPClassifier

- Với RFC: chọn mô hình tốt nhất dựa trên `n_estimators` và `num_top_titles`
- Với SVC: chọn mô hình tốt nhất dựa trên `num_top_titles`
- Với MLP: chọn mô hình tốt nhất dựa trên số lớp ẩn, số neural ở lớp ẩn và `num_top_titles`

Mô hình hóa - RFC

Các thiết lập của nhóm:

- `max_features = 0.3`: theo nhóm tìm hiểu thì nên chọn từ 0.3 - 0.5
- `min_samples_leaf = 3`: theo nhóm tìm hiểu thì nên chọn từ 3 - 5
- `max_depth = 10`: theo nhóm tìm hiểu thì nên chọn từ 5 - 10

Score cao nhất của mô hình này:

```
: best_val_score
```

```
0.7142857142857143
```

Mô hình hóa - SVC

Các thiết lập của nhóm:

- `kernel = 'linear'`: vì nhóm muốn thử theo mô hình tuyến tính
- `decision_function_shape = 'ovo'`: theo nhóm tìm hiểu thì phân nhiều lớp nên chọn thiết lập này

Score cao nhất của mô hình này:

```
: best_val_score
```

```
0.6532738095238095
```

Mô hình hóa - MLP

Các thiết lập của nhóm:

- solver='adam': theo nhóm tìm hiểu thì dữ liệu hơn 1000 dòng nên dùng adam
- early_stopping=True: dừng khi kết quả không được cải thiện, để tránh overfitting

Score cao nhất của mô hình:

- Mô hình 1 hidden_layer_sizes = (128):

```
best_val_score
0.6666666666666666
```

- Mô hình 2 hidden_layer_sizes = (256, 128):

```
: best_val_score
0.6696428571428571
```

Đánh giá kết quả

Mô hình RFC cho kết quả trên tập validation cao nhất vì vậy chọn mô hình này để kiểm tra tập test

Kết quả trên tập test:

```
full_pipeline.score(test_X, test_y)
```

```
[Parallel(n_jobs=1)]: Using backend Sequen  
[Parallel(n_jobs=1)]: Done 26 out of 26
```

```
0.7416666666666667
```

Nhìn lại quá trình

- Những khó khăn gặp phải:
 - ✓ Thu thập dữ liệu: trang web đầu tốn nhiều thời gian nhưng kết quả không như mong muốn, trang web thứ hai bị block, trang web thứ ba quá ít dữ liệu ☹
 - ✓ Dữ liệu quá thô gây khó khăn trong tiền xử lý
 - ✓ Khó khăn trong việc lựa chọn mô hình
- Bài học rút ra:
 - ✓ Thực hành lại quy trình một bài toán Khoa học dữ liệu
 - ✓ Hiểu rõ hơn về quá trình tiền xử lý, khám phá dữ liệu, mô hình hóa
 - ✓ Biết thêm nhiều hàm của python
- Nếu có thêm thời gian nhóm em sẽ làm kỹ hơn

Tài liệu tham khảo

- <https://www.kaggle.com/hypopossum/gsm-arena-phone-dataset>
- <https://stackoverflow.com/questions/20463281/how-do-i-solve-overfitting-in-random-forest-of-python-sklearn>
- <https://www.kaggle.com/iabhishekofficial/mobile-price-classification?select=train.csv>
- <https://github.com/hmhuan/Data-science-project>
- <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
- Các file bài tập, demo của thầy



CẢM ƠN THẦY ĐÃ THEO DÕI

