

NHẬP MÔN KHOA HỌC

DỮ LIỆU

VẤN ĐÁP ĐỒ ÁN CUỐI KỲ

Nhóm 02
Merlin
Khóa 2022



GIẢNG VIÊN



Thầy Lê Ngọc Thành



Thầy Lê Nhựt Nam

Introduction to Data Science - CQ2022/21



THÀNH VIÊN NHÓM



Bùi Đình Bảo
21120201



Nguyễn Hữu Bền
22120029



Lê Nguyễn Gia Bảo
22120023



Trương Tiến Anh
22120017

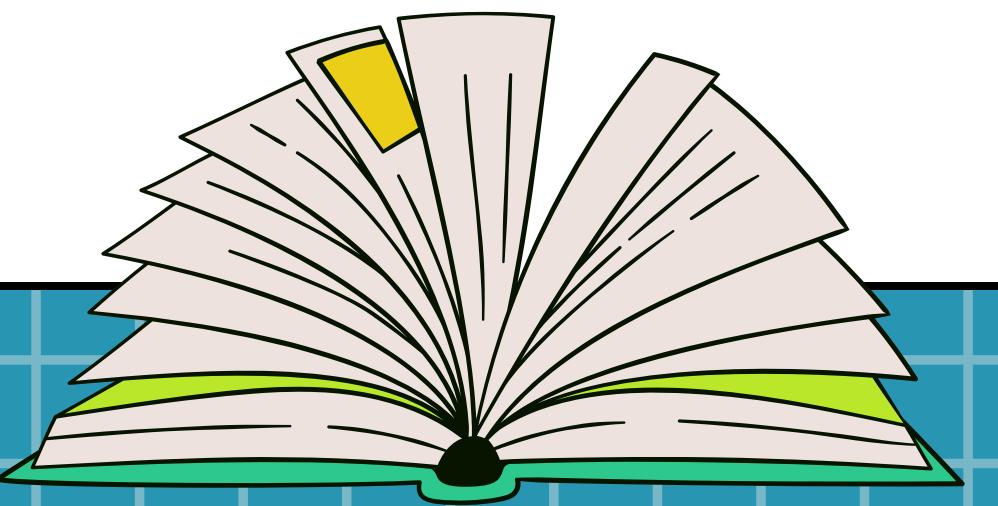


Đoàn Minh Cường
22120043



NỘI DUNG

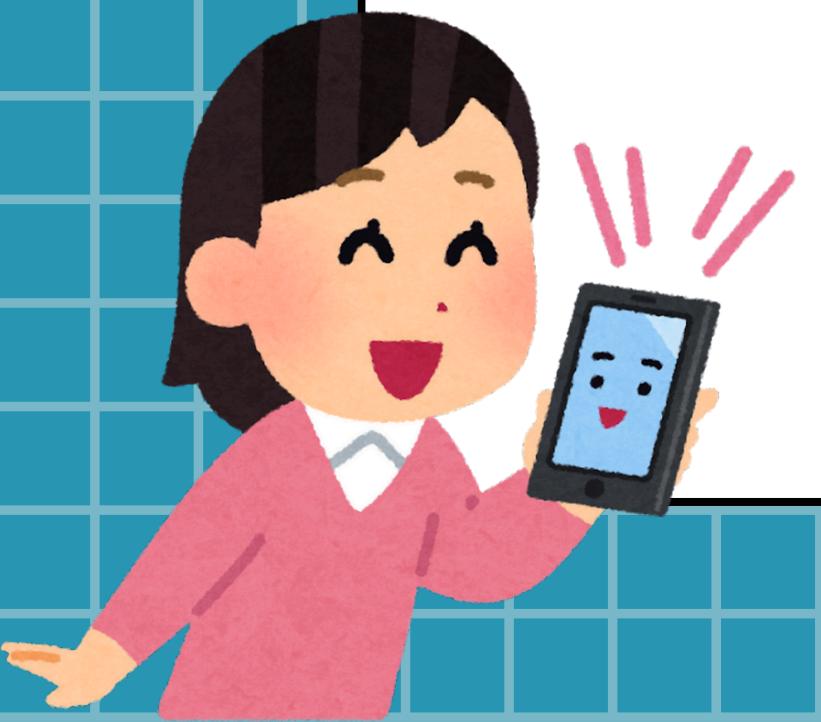
- **Tổng quan đề tài**
- **Thu thập dữ liệu**
- **Tiền xử lý dữ liệu**
- **Khám phá dữ liệu**
- **Phân tích và trực quan hóa**
- **Mô hình hóa dữ liệu**
- **Đánh giá**



TỔNG QUAN ĐỒ ÁN

Đề tài
Phân tích thị trường **điện thoại di động**
dựa trên dữ liệu thu thập từ cửa hàng
Mobile City.

Nguồn: **MOBILE CITY** (mobilecity.vn)



TỔNG QUAN ĐỒ ÁN

Lý do chọn
đề tài:

Mục tiêu
nghiên cứu:

- Điện thoại thông minh là công cụ không thể thiếu trong cuộc sống hiện đại, do tính năng hữu ích mà nó mang lại.
- Cung cấp cái nhìn sâu sắc và giá trị cho: nhà sản xuất, chủ cửa hàng, người tiêu dùng.
- Phân tích các xu hướng hiện tại của các mẫu điện thoại.
- Cung cấp thông tin giá trị để hỗ trợ nghiên cứu thị trường.
- Phát triển các chiến lược marketing hiệu quả cho các thương hiệu điện thoại.
- Đề xuất phương pháp định giá cho các mẫu điện thoại mới.

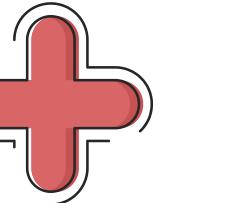


THU THẬP DỮ LIỆU

- Công cụ sử dụng:



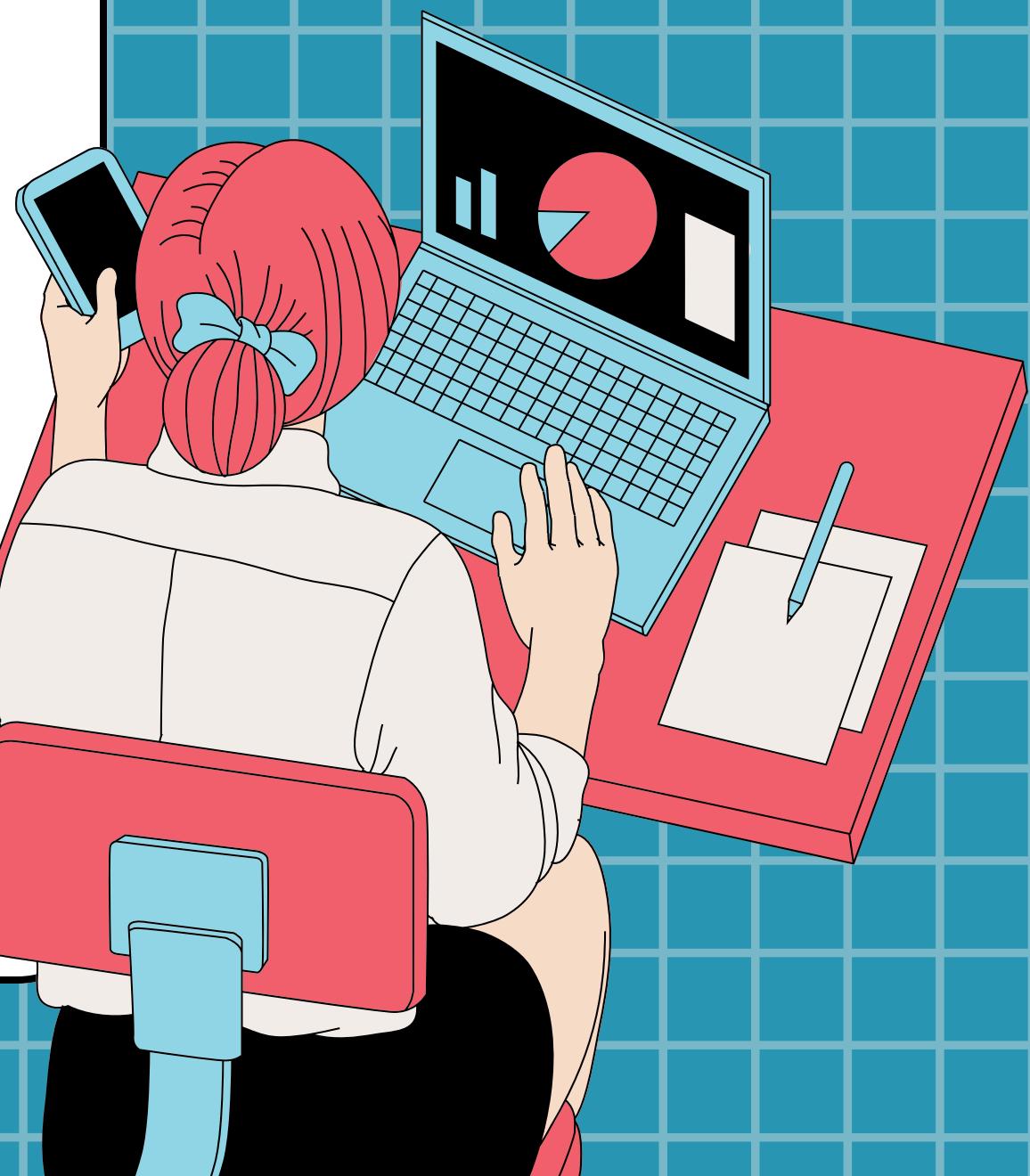
Selenium



BeautifulSoup

- Trang web thu thập: MobileCity

<https://mobilecity.vn/dien-thoai>



THU THẬP DỮ LIỆU

- Quy trình thực hiện:
 1. Thu thập link đến từng trang điện thoại chi tiết
 2. Lưu link của từng điện thoại vào file *links.csv*
 3. Lấy link đến từng trang điện thoại từ file *links.csv*
 4. Thu thập các thuộc tính của điện thoại ứng với mỗi link
 5. Lưu dữ liệu của mỗi điện thoại vào file *raw_data.csv*



THU THẬP DỮ LIỆU

- Về dữ liệu thu thập được:

File *links.csv*

- + Số lượng dòng: 1444
- + Số lượng cột: 2
(Index + Link)

File *raw_data.csv*

- + Số lượng dòng: 1444
- + Số lượng cột: 9



THU THẬP DỮ LIỆU

- Các thuộc tính trong **raw_data.csv**:
 - Index
 - Đánh giá
 - Loại điện thoại
 - Tên sản phẩm
 - Màu sắc - Phiên bản bộ nhớ - giá tương ứng
 - Thời gian bảo hành
 - Số lượt đánh giá và hỏi đáp
 - Thông số kỹ thuật
 - Đường dẫn



TIỀN XỬ LÝ DỮ LIỆU

- Quy trình thực hiện:
 1. Tải dữ liệu từ file *raw_data.csv*
 2. Xử lý dữ liệu cơ bản:
 - a. Xóa các dòng dữ liệu bị trùng và bị sai
 - b. Đặt lại tên thuộc tính
 - c. Phân tách thành các mẫu điện thoại cho cùng một tên điện thoại
 3. Trích xuất thuộc tính cần thiết đồng thời chuyển đổi định dạng dữ liệu
 4. Lưu kết quả vào file *processed_data.csv*



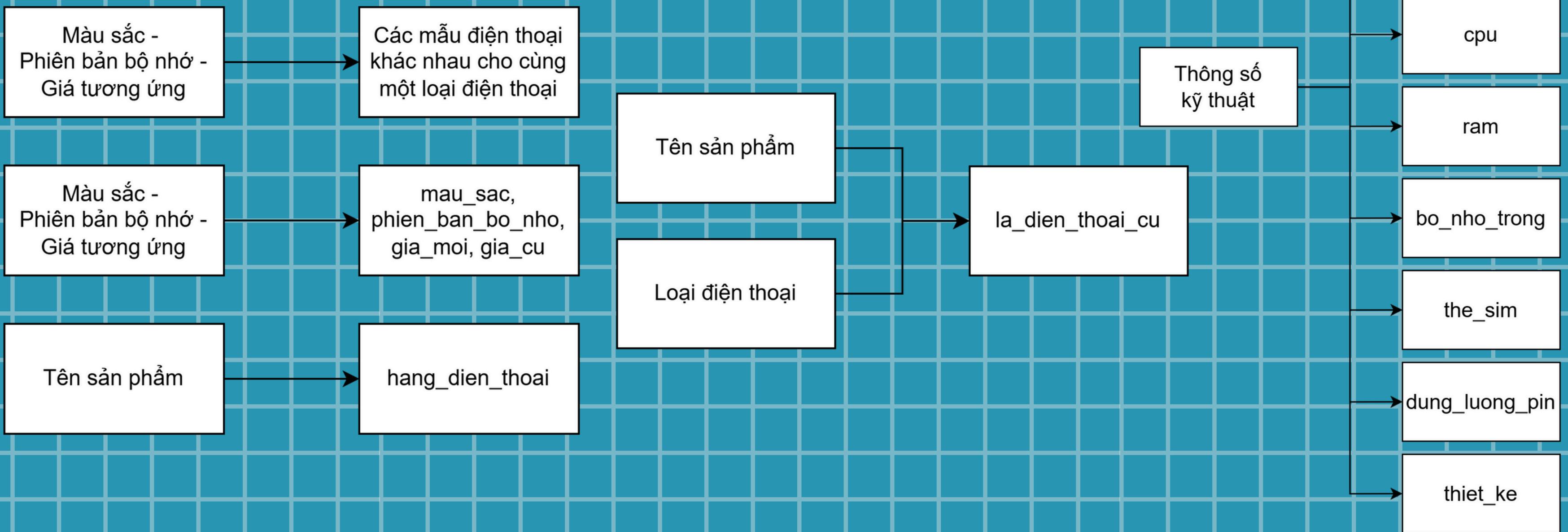
TIỀN XỬ LÝ DỮ LIỆU

- Thuộc tính được trích xuất như thế nào?
 1. Define một **custom aggregation function** cho mỗi thuộc tính cần trích xuất
 2. Sử dụng phương thức **.apply** trong pandas để tiến hành trích xuất thuộc tính

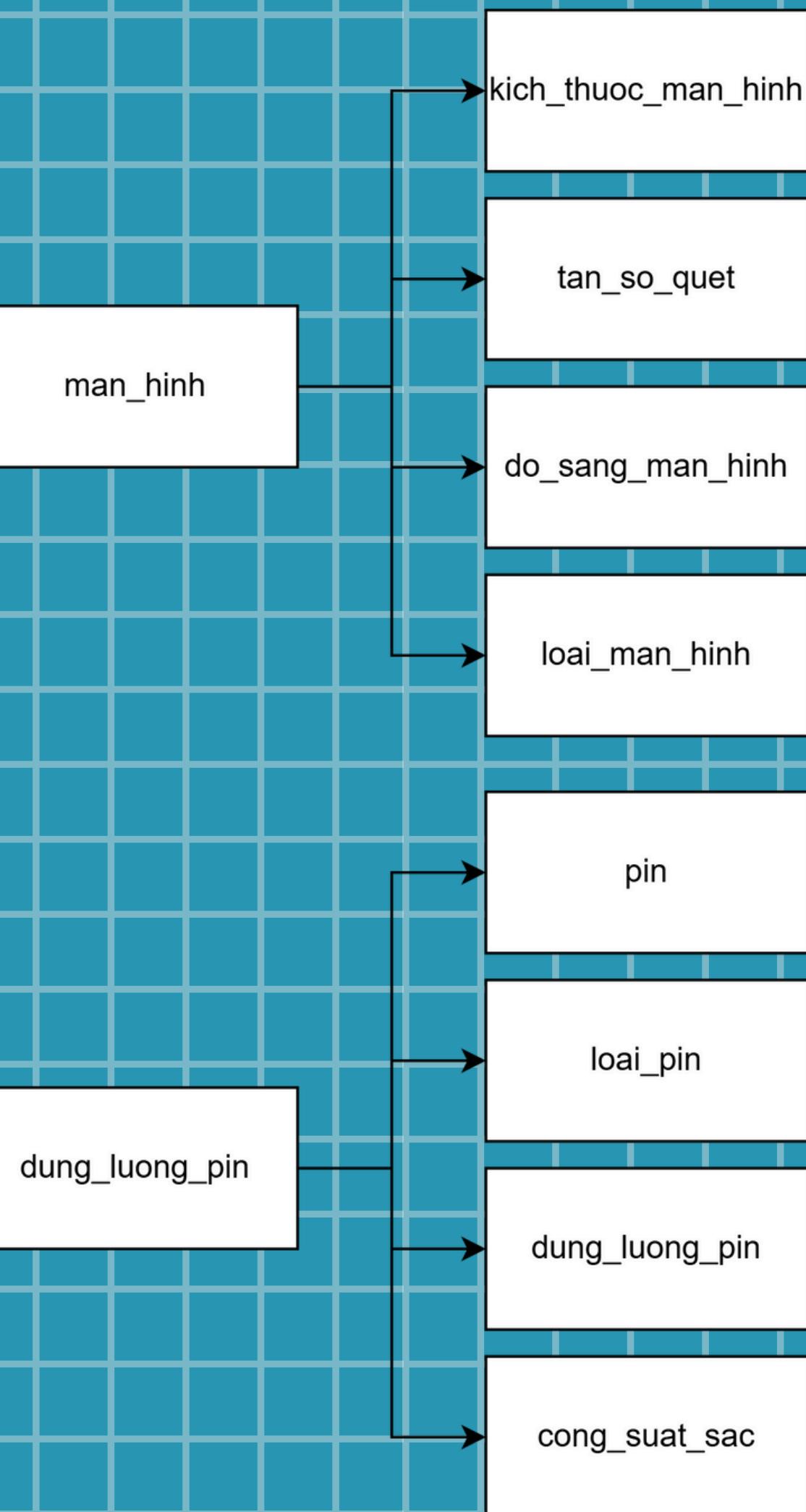
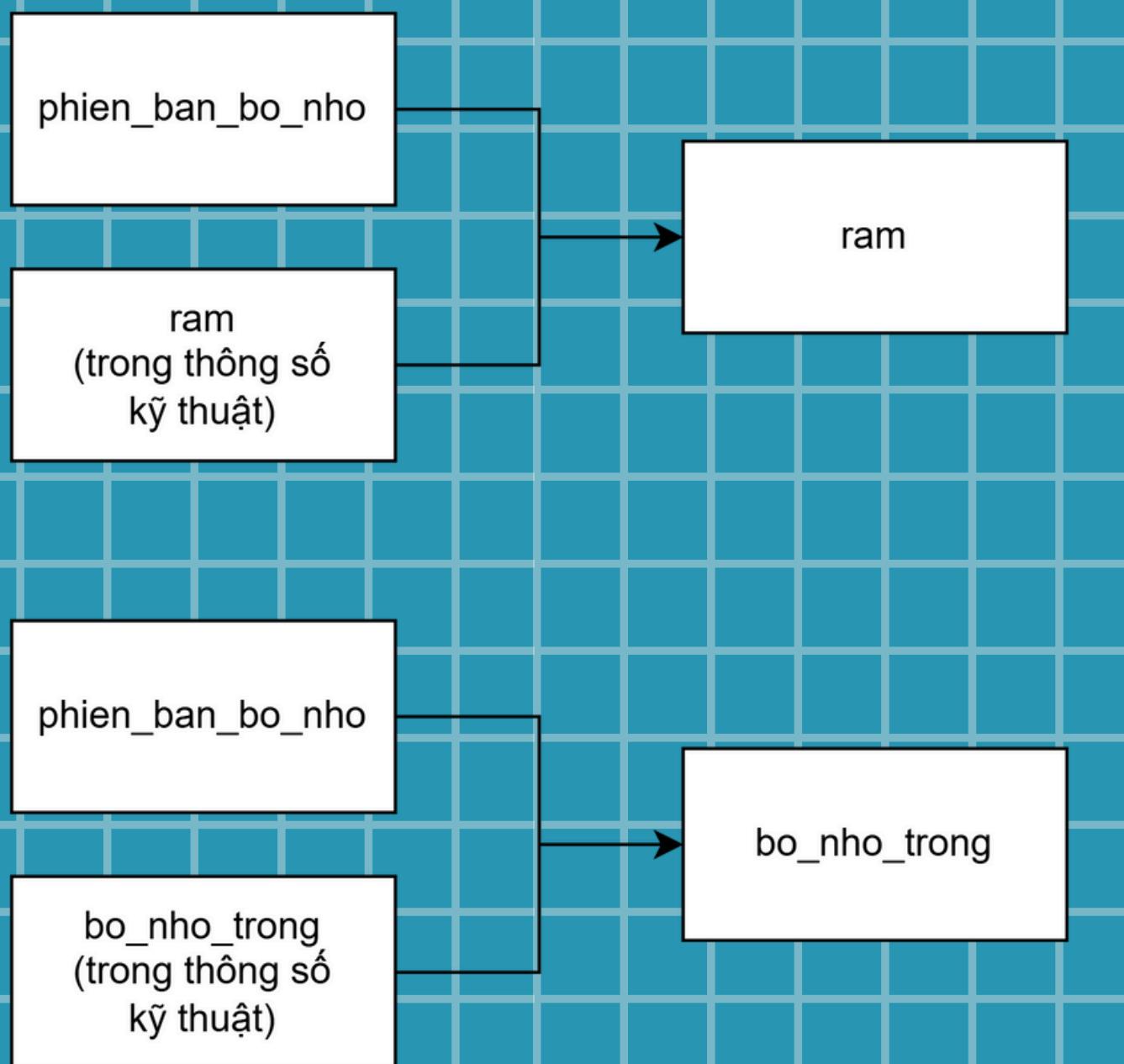
** Có thể sử dụng hàm định nghĩa trên để đồng thời chuyển đổi định dạng dữ liệu



RAW_DATA TO PROCESSED_DATA



RAW_DATA TO PROCESSED_DATA



TIỀN XỬ LÝ DỮ LIỆU

- Dữ liệu sau khi đã được tiền xử lý:

- + Số lượng dòng: 8662 (**mẫu điện thoại**)
- + Số lượng cột: 25 (thuộc tính)

** *Mỗi **mẫu điện thoại** có thể có tên giống nhau nhưng về màu sắc, phiên bản bộ nhớ khác nhau dẫn đến giá tương ứng cũng khác nhau*



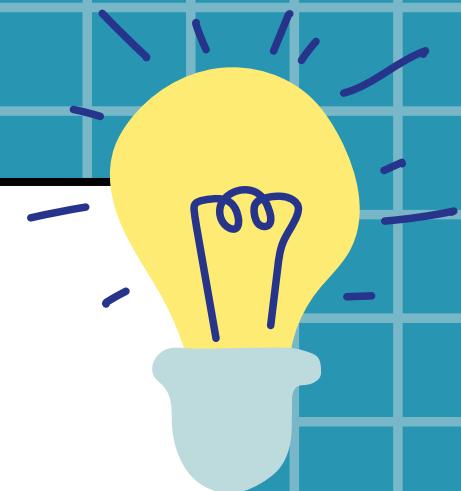
TIỀN XỬ LÝ DỮ LIỆU

- Các công đoạn khác trong tiền xử lý ở đâu?

1. Xử lý dữ liệu bị thiếu
2. Số hóa dữ liệu
3. Xử lý dữ liệu ngoại lai
4. Chuẩn hóa dữ liệu
5. ...



KHÁM PHÁ DỮ LIỆU

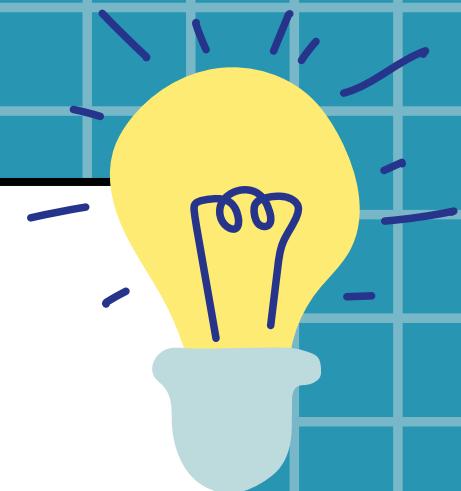


Quan sát chi tiết các dòng và cột để rút ra những thông tin hữu ích, phục vụ cho việc phân tích và xây dựng mô hình.

Quá trình khám phá sẽ được thực hiện theo thứ tự từ tổng quát đến chi tiết.



KHÁM PHÁ DỮ LIỆU

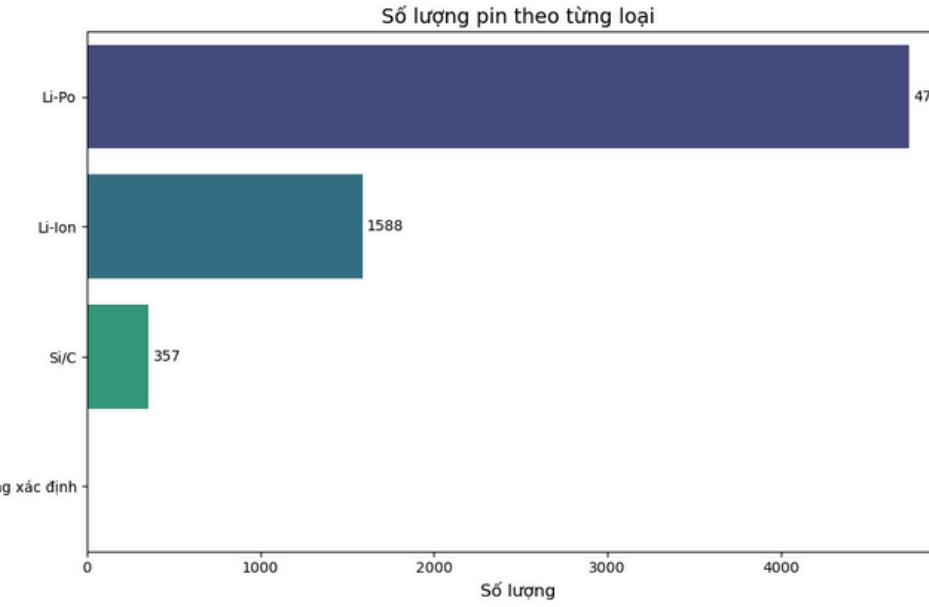
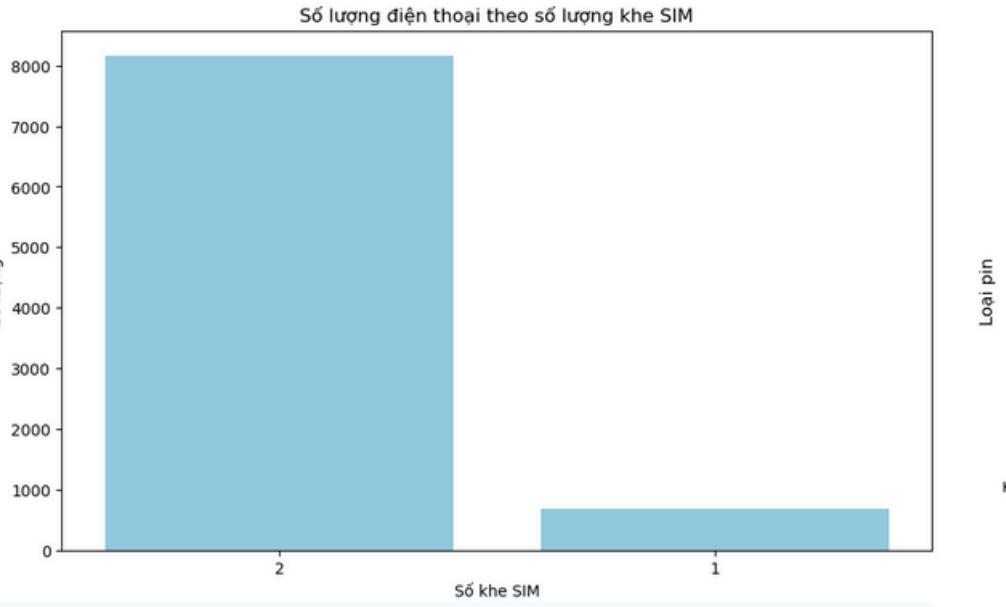
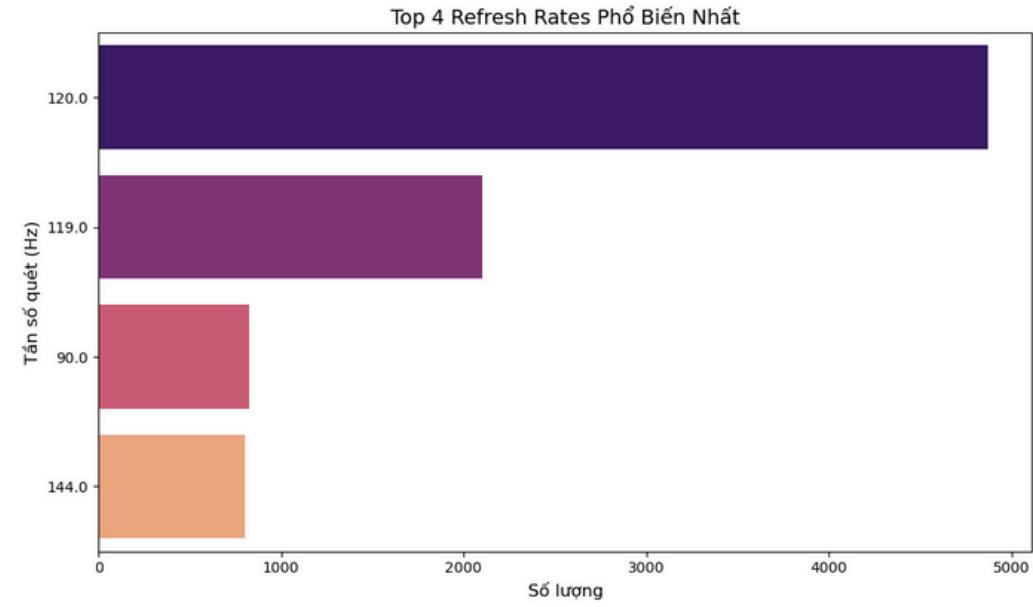
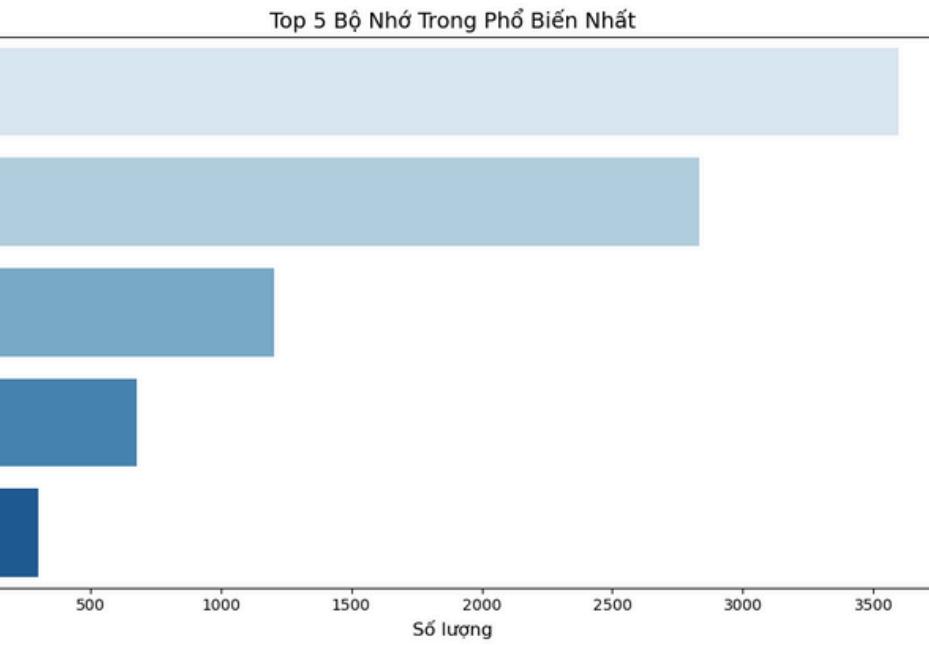
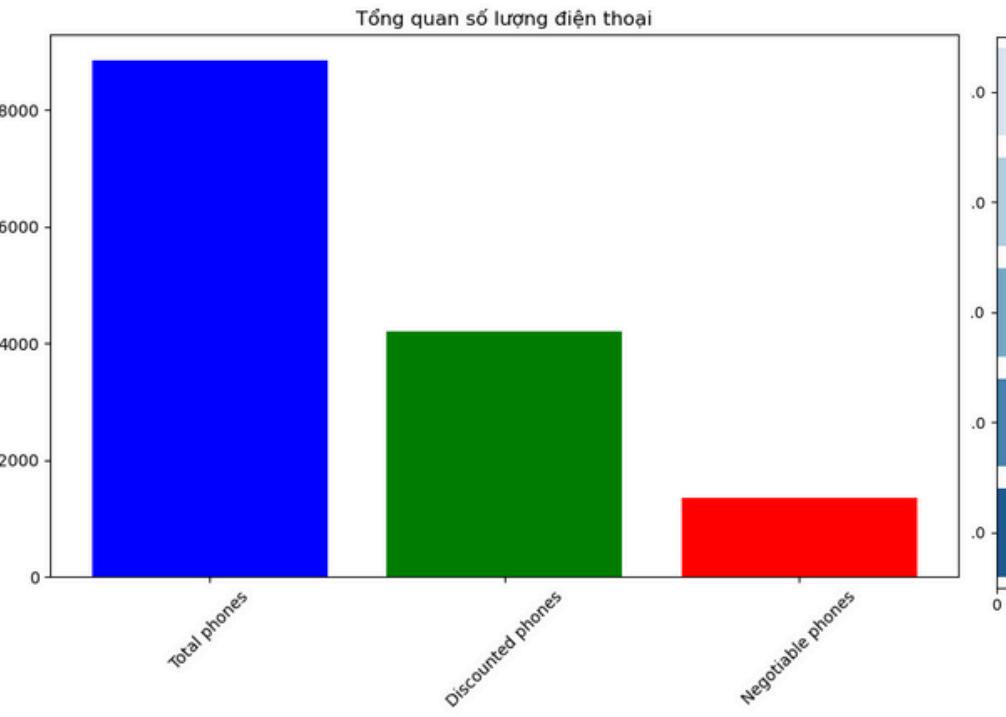
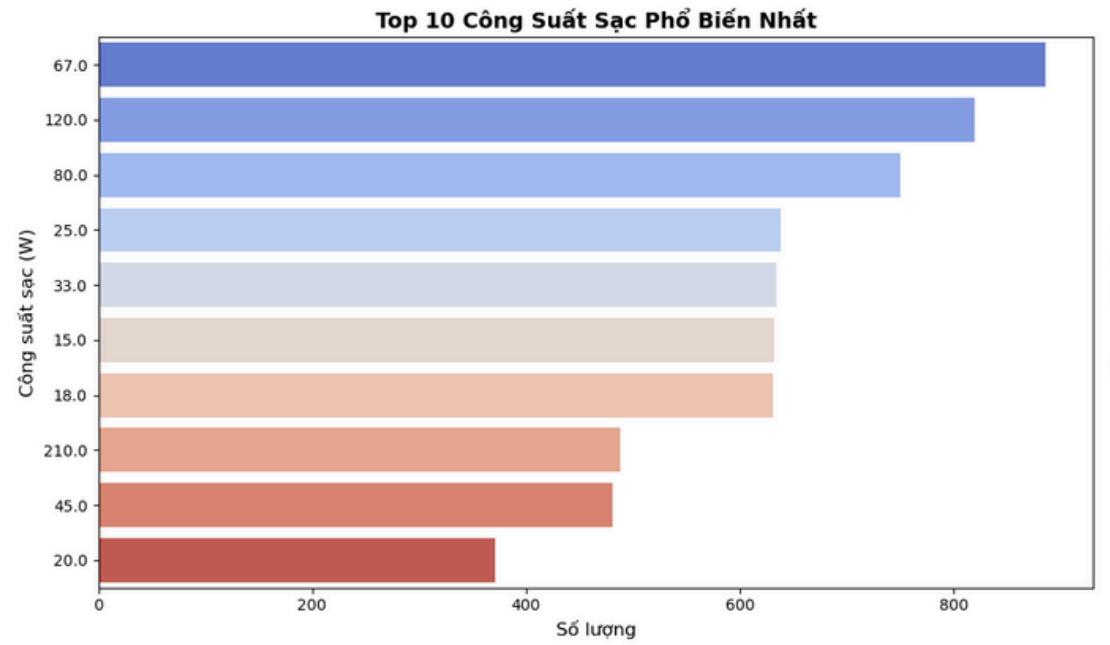
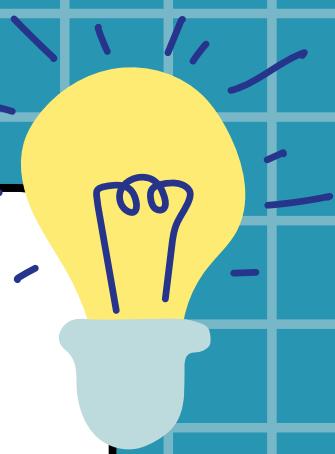


Từ việc khám phá tổng quan, rút ra một số nhận xét sau:

- Bộ dữ liệu có 8847 dòng và 26 cột sau khi xóa các dòng trùng lặp.
- Mỗi dòng đại diện cho một mẫu điện thoại với các thông số kỹ thuật khác nhau và cần đồng nhất về ý nghĩa để đảm bảo phân tích chính xác.
- Một số thuộc tính có tỷ lệ thiếu dữ liệu cao, sẽ được xử lý hoặc bổ sung thay vì xóa ngay.
- Dữ liệu chứa một số giá trị bất hợp lý và sự phân bố bị lệch, cần được kiểm tra và xử lý.
- Một số thuộc tính phân loại có nhiều giá trị do nhập liệu không nhất quán, cần chuẩn hóa để nâng cao chất lượng phân tích.



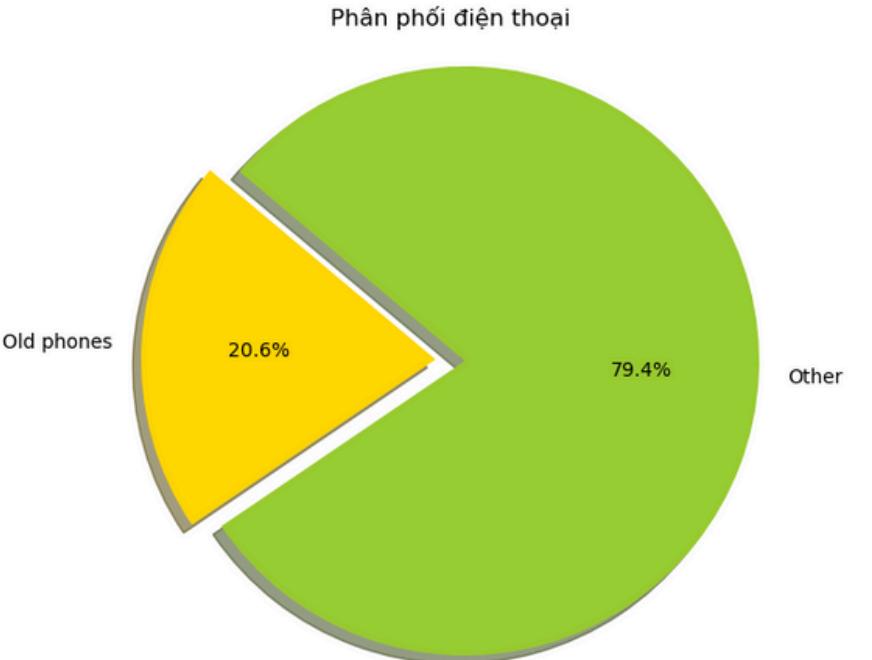
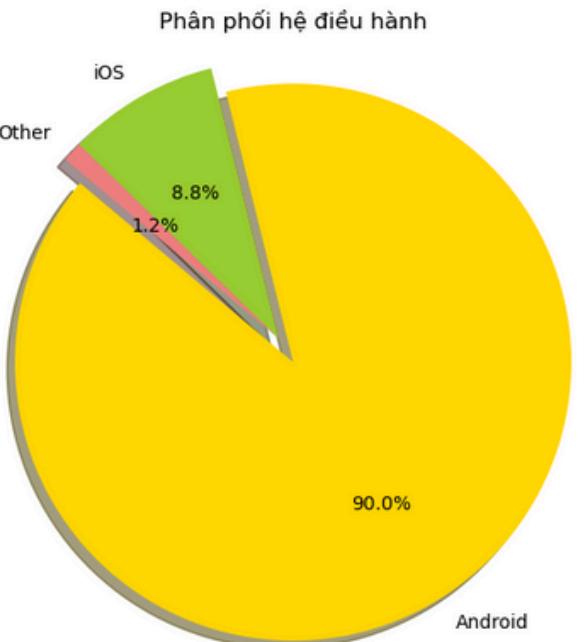
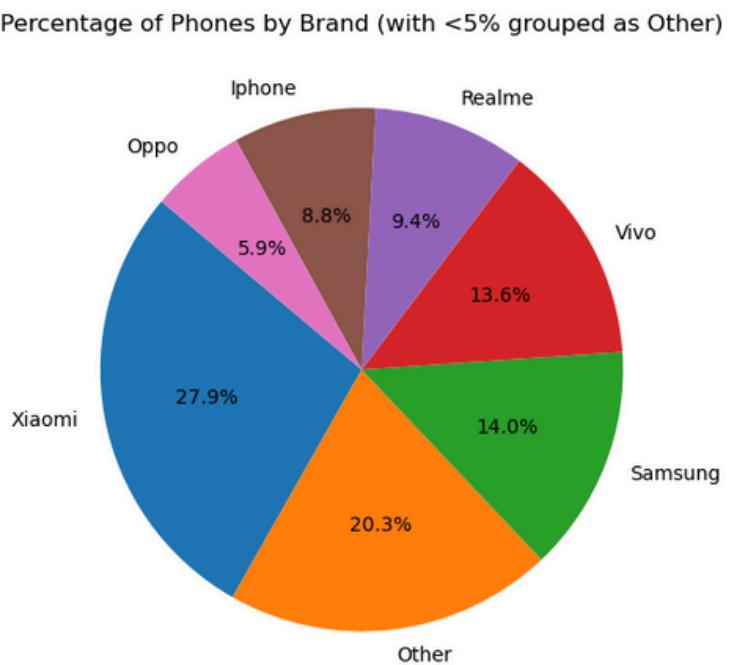
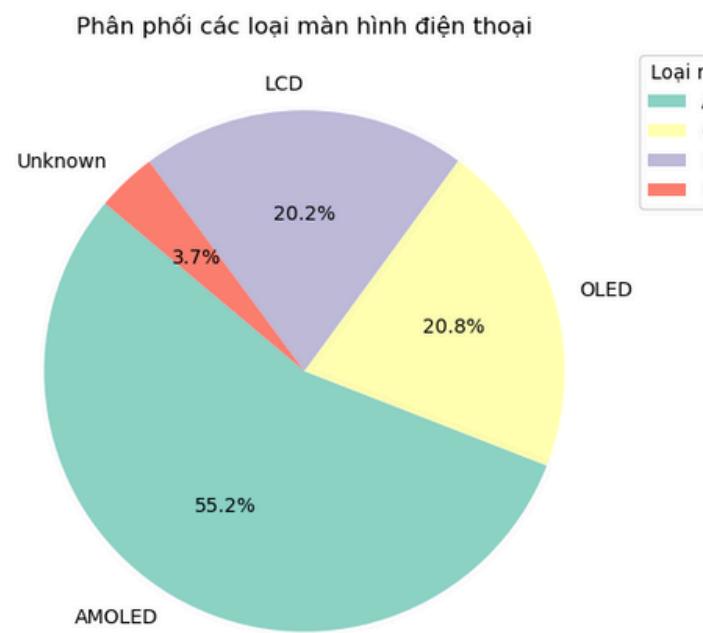
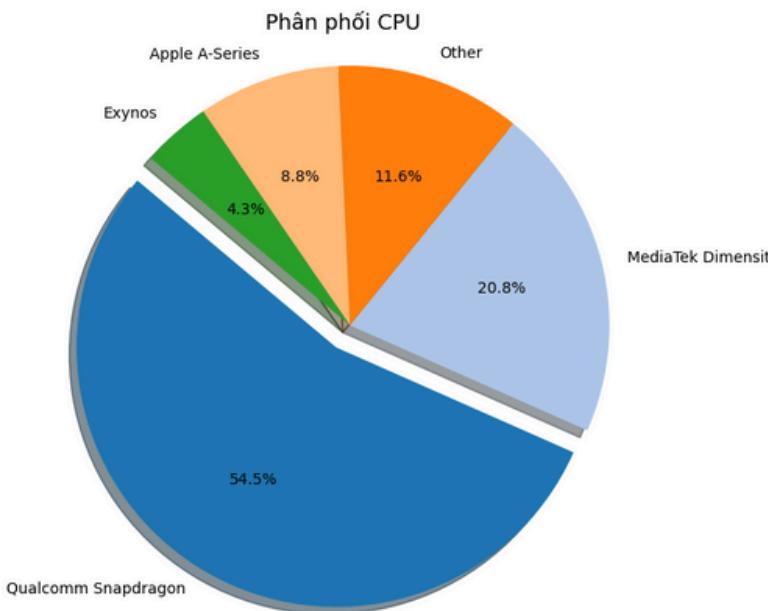
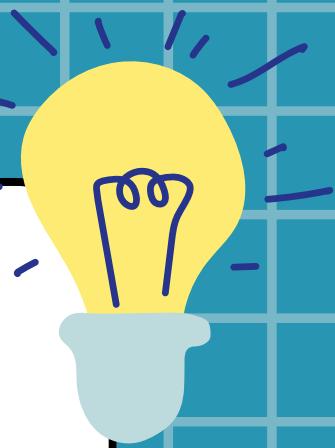
KHÁM PHÁ DỮ LIỆU



TRỰC QUAN HÓA



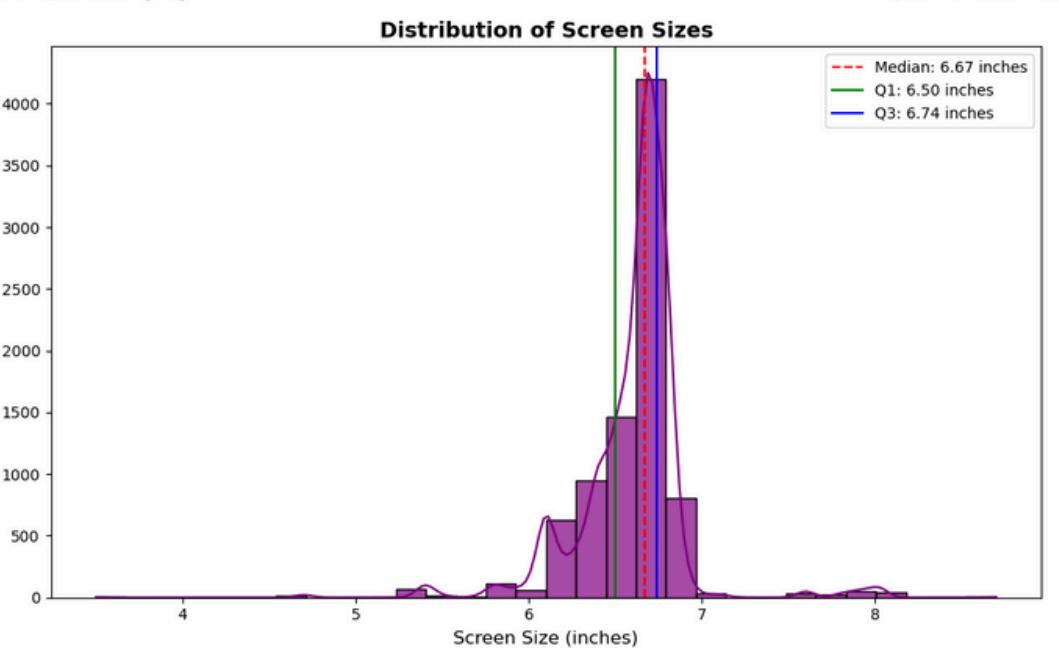
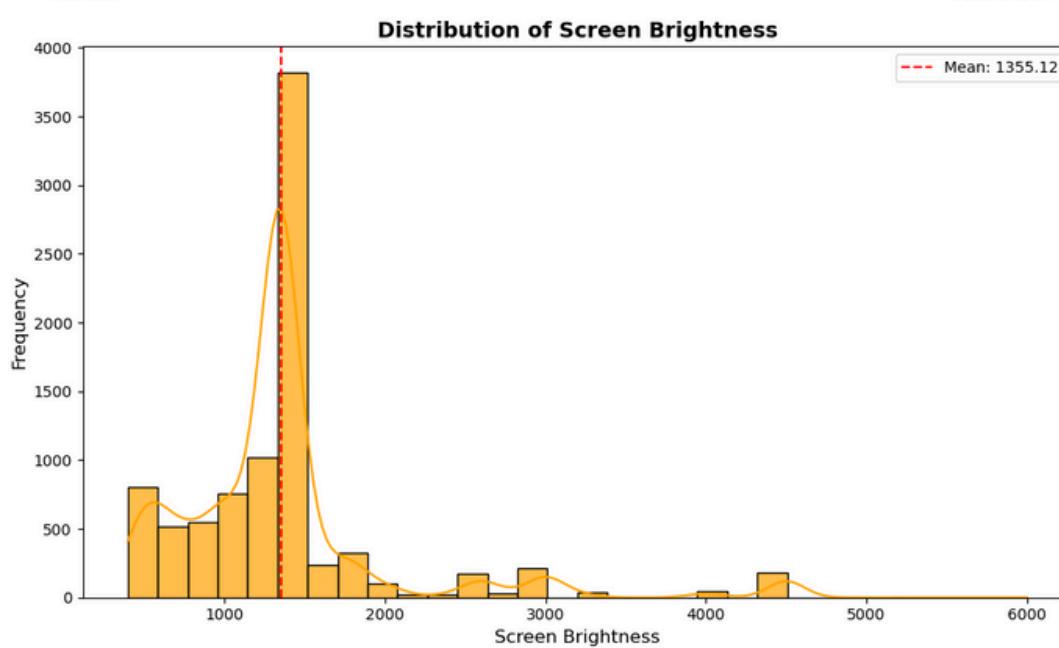
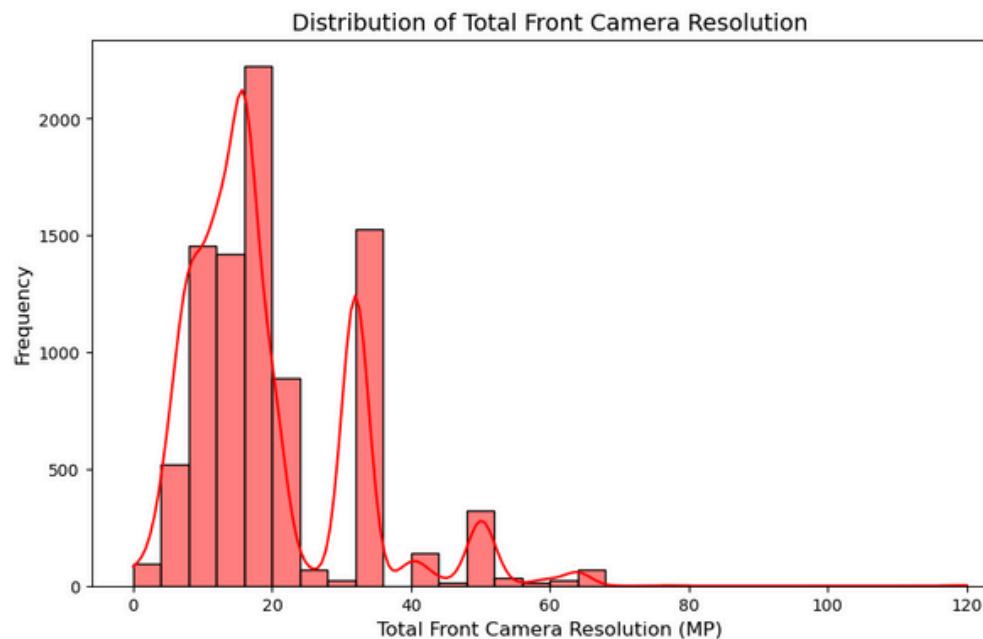
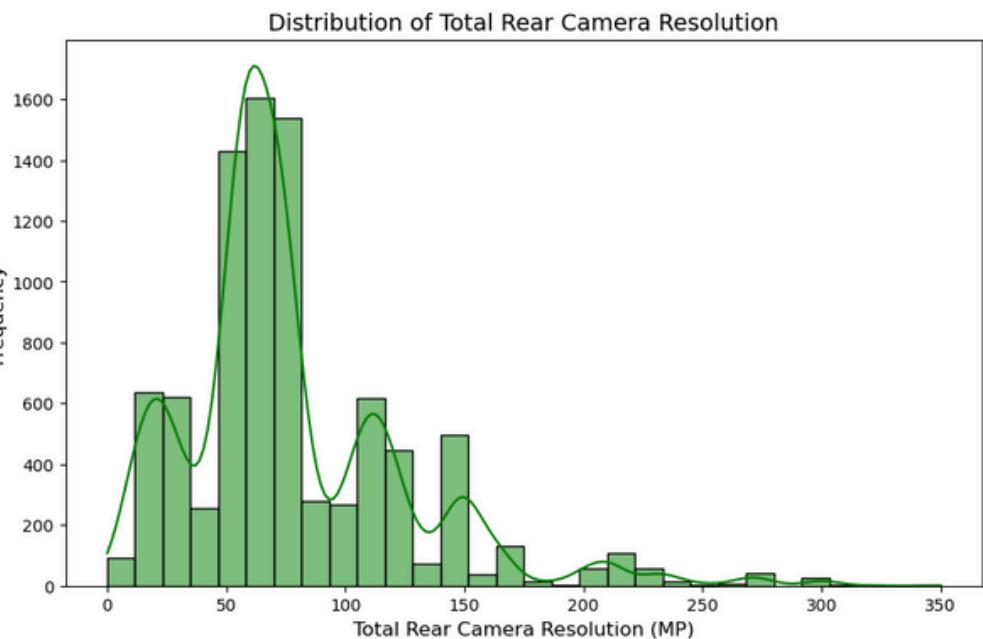
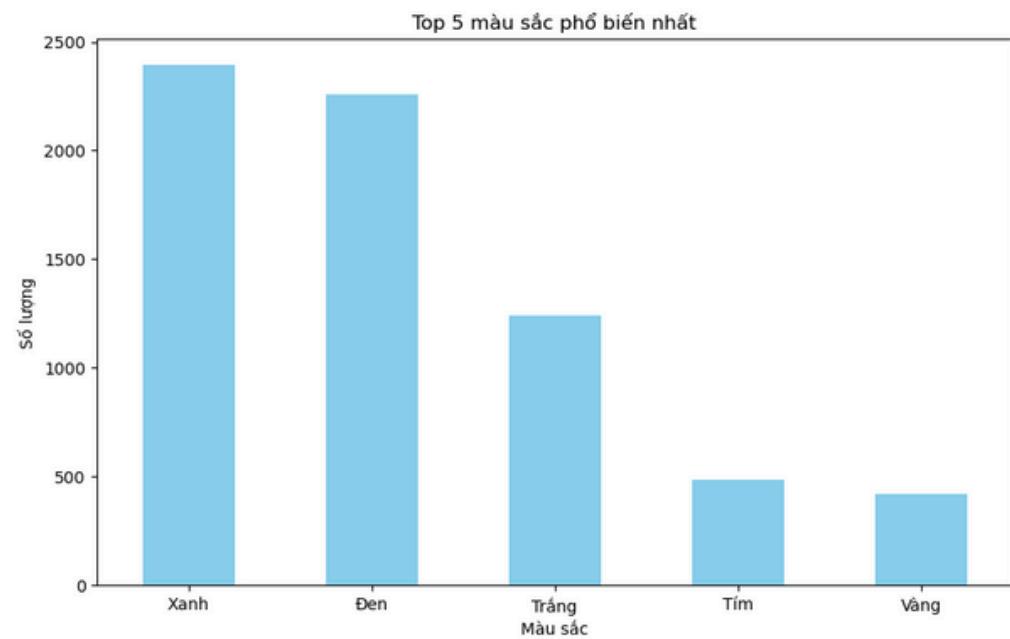
KHÁM PHÁ DỮ LIỆU



TRỰC QUAN HÓA



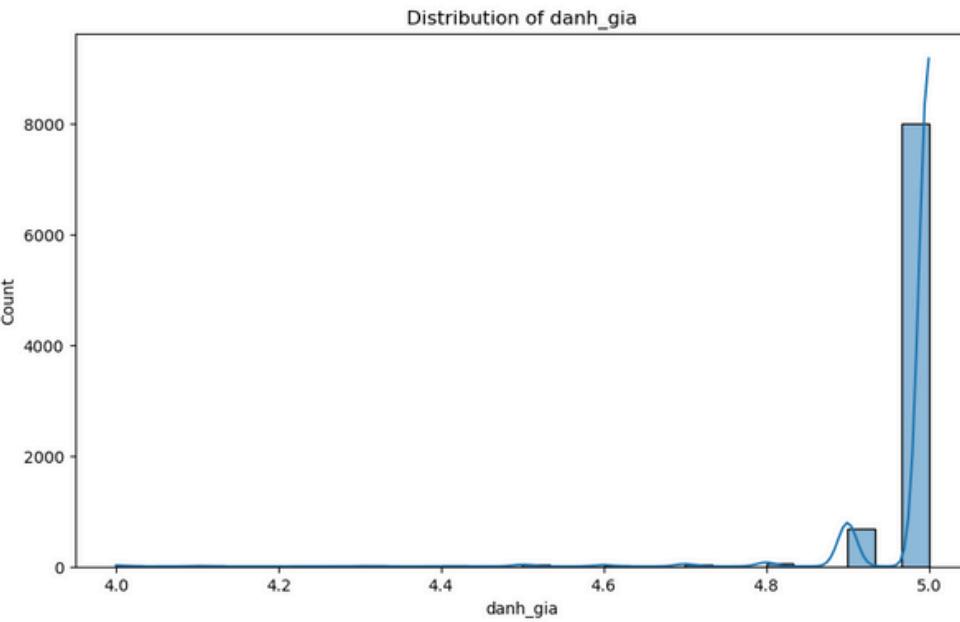
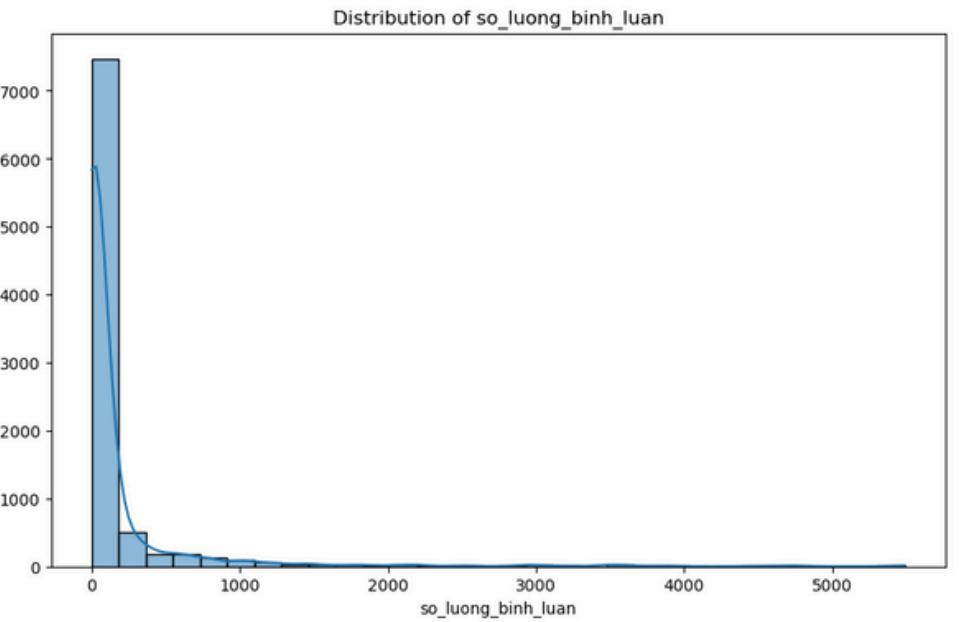
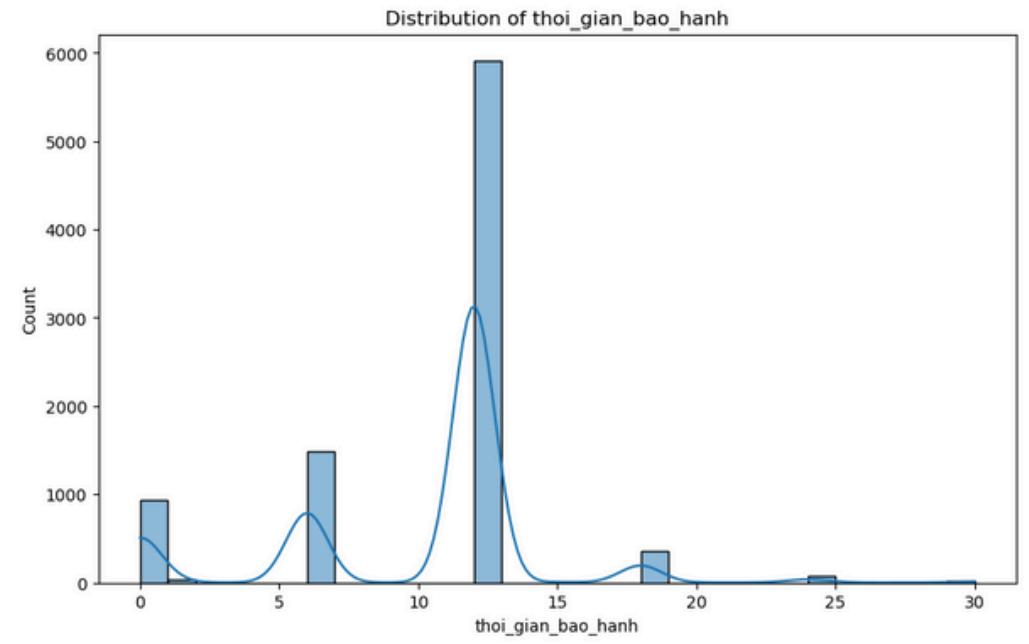
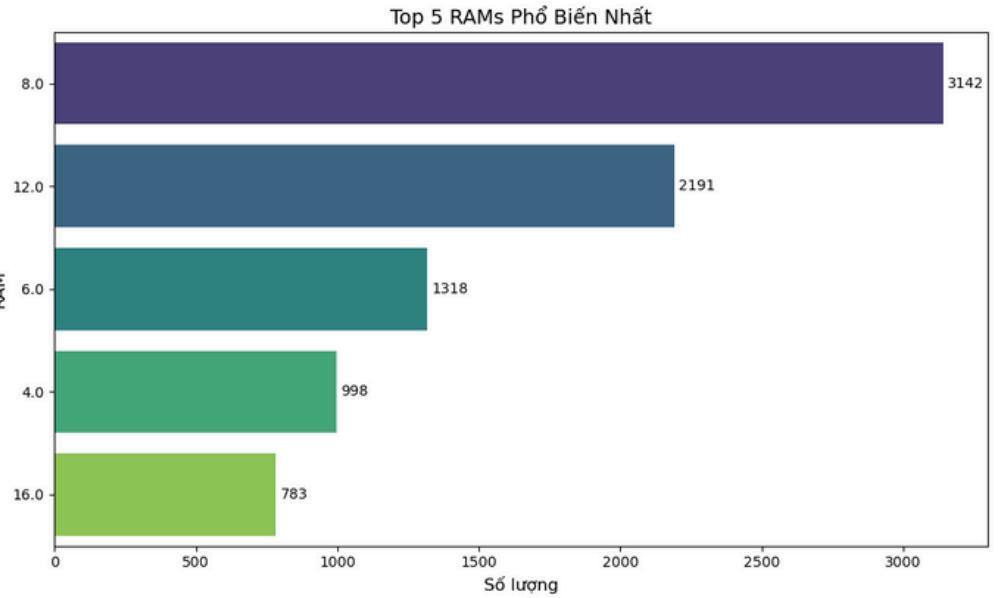
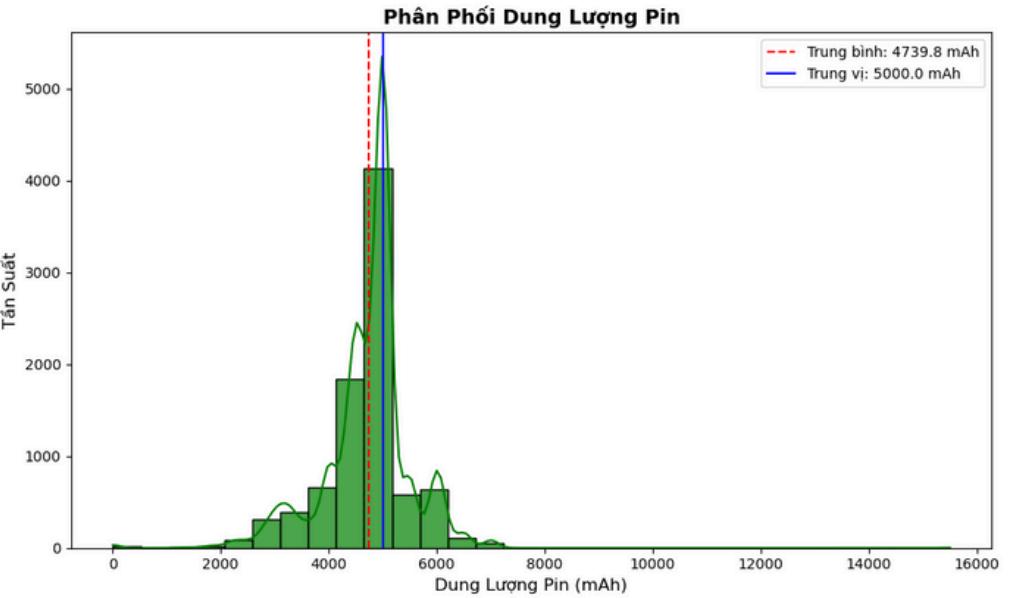
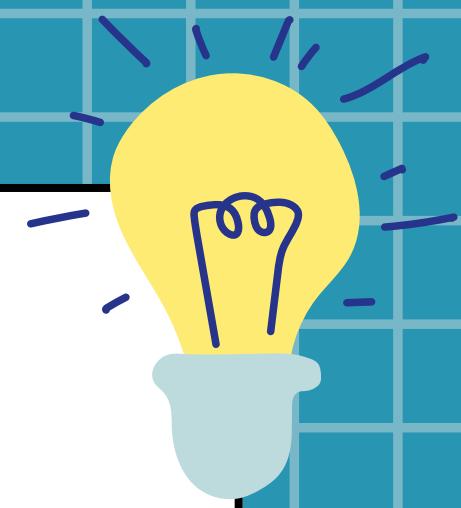
KHÁM PHÁ DỮ LIỆU



TRỰC QUAN HÓA

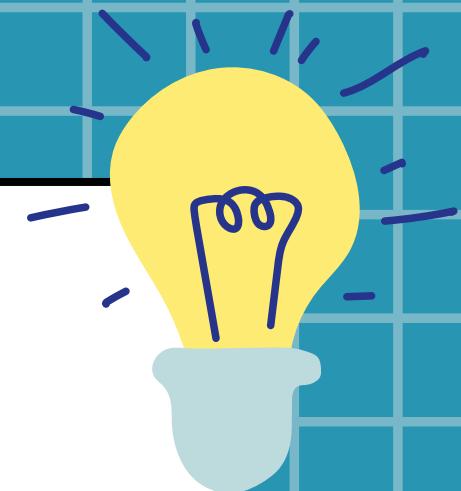


KHÁM PHÁ DỮ LIỆU



TRỰC QUAN HÓA

KHÁM PHÁ DỮ LIỆU



Từ việc khám phá chi tiết, rút ra một số nhận xét sau:

Xu hướng điện thoại trên thị trường Việt Nam như sau:

- **Màu sắc:** Xanh và Đen là phổ biến nhất, theo sau là Trắng, Tím, và Vàng.
- **Pin và màn hình:** Li-Po và màn hình AMOLED chiếm ưu thế; IPS LCD phổ biến nhất.
- **Phân khúc giá:** Cao cấp (>15 triệu), tầm trung (7-15 triệu), giá rẻ (<7 triệu).
- **Hiệu năng:** Snapdragon và Dimensity là CPU phổ biến; RAM 8-12GB, lưu trữ 128-256GB.
- **Kích thước màn hình:** Chủ yếu 6.50-6.74 inch, trung bình 6.67 inch; độ sáng phổ biến 1000 nits.
- **Tần số quét và bảo hành:** Tần số 120Hz và bảo hành tiêu chuẩn 12-24 tháng.

Thị trường điện thoại Việt Nam đáp ứng tốt nhu cầu với đa dạng sản phẩm.



PHÂN TÍCH DỮ LIỆU

- Các câu hỏi phân tích, gồm 7 câu:

- 1) Mức giá giảm có khiến cho điện thoại trở nên phổ biến và nổi bật hơn hay không?
- 2) Thời gian bảo hành có ảnh hưởng đến sự hài lòng của khách hàng không?
- 3) Dòng điện thoại nào được quan tâm nhiều nhất từ trước đến nay dựa trên số lượt đánh giá và hỏi đáp? (Top 10)
- 4) Ứng với mỗi dòng điện thoại thì những mẫu điện thoại nào được quan tâm nhiều nhất? (Top 3)
- 5) Trong các thông số kỹ thuật của một chiếc điện thoại, những thông số nào thường có ảnh hưởng lớn nhất đến giá bán?
- 6) Kiểu thiết kế điện thoại nào phổ biến nhất hiện nay, dựa trên các kiểu thiết kế của các mẫu điện thoại hiện có trong cửa hàng?
- 7) Phân bố giá bán của các hãng điện thoại được thể hiện như thế nào?



PHÂN TÍCH DỮ LIỆU

- Chuẩn bị dữ liệu:

```
1 # Đọc file csv
2 data = pd.read_csv('../data/processed_data.csv', index_col=0)
3
4 # Chuyển đổi object thành list
5 data["do_phan_giai_cam_truoc"] = data["do_phan_giai_cam_truoc"].apply(ast.literal_eval)
6 data["do_phan_giai_cam_sau"] = data["do_phan_giai_cam_sau"].apply(ast.literal_eval)
7
8 data.info()
9 data.head()
```



CÂU HỎI 1

Mức giá giảm có khiến cho điện thoại trở nên
phổ biến và nổi bật hơn hay không?

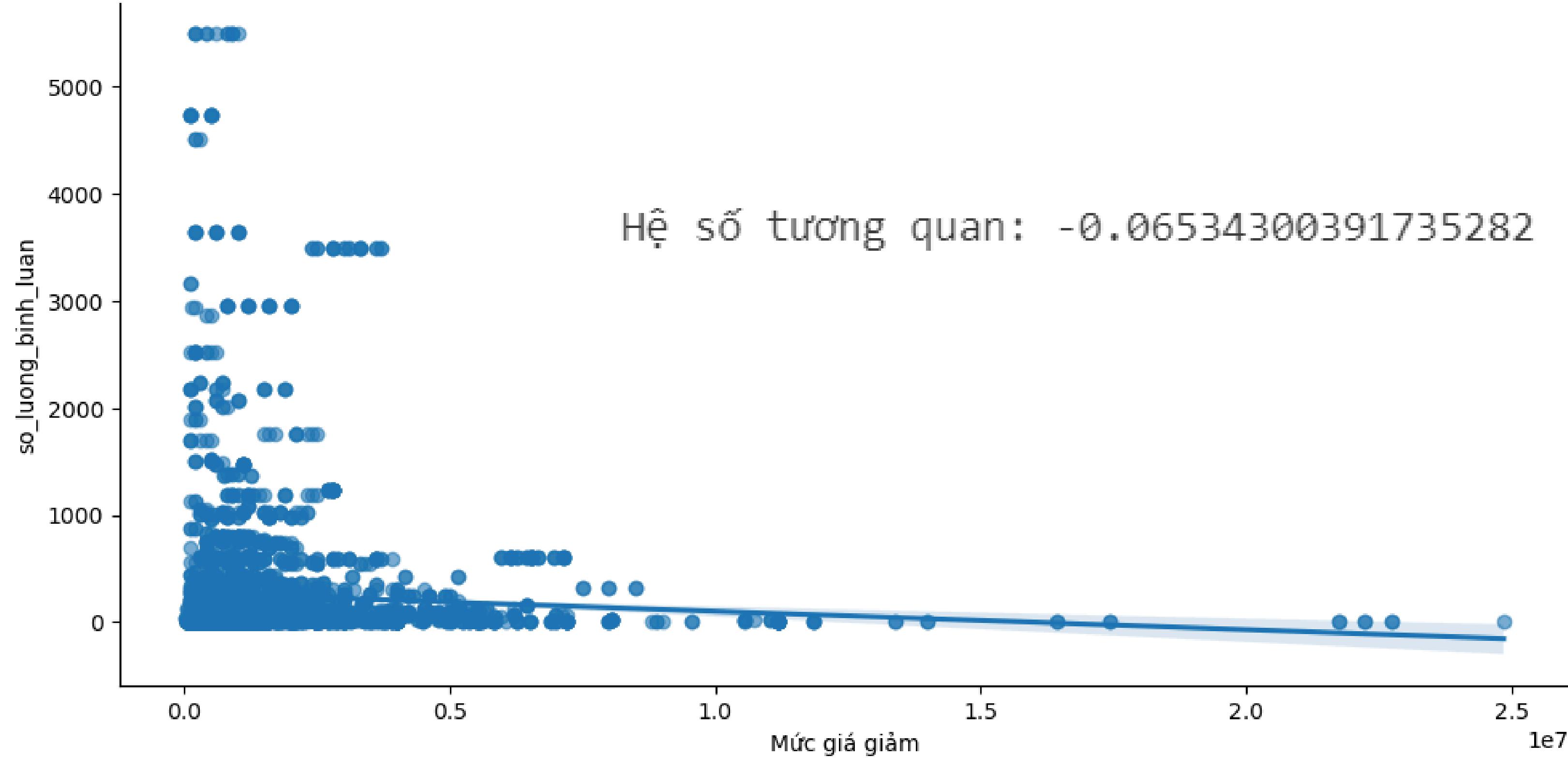


TIỀN XỬ LÝ

```
1 df = data[["gia_cu", "gia_moi", "so_luong_binh_luan"]].copy().dropna()  
2  
3 # Tính toán mức giá giảm = Giá cũ - Giá mới  
4 df['Mức giá giảm'] = df['gia_cu'] - df['gia_moi']  
5 df['Có giảm giá'] = df['Mức giá giảm'] > 0
```

TRỰC QUAN HÓA

Hồi quy tuyến tính: Mức giá giảm vs Số lượt bình luận



CÂU HỎI 4

Ứng với mỗi dòng điện thoại thì những mẫu điện thoại nào được quan tâm nhiều nhất? (Top 3)



TIỀN XỬ LÝ

```
1 df = data[["hang_dien_thoai", "so_luong_binh_luan"]].copy().dropna()
2 df = df.groupby("hang_dien_thoai")["so_luong_binh_luan"].sum().sort_values(ascending=False).to_frame().reset_index().head(10)
3 brands = df["hang_dien_thoai"].to_list()
4 df = data[["hang_dien_thoai", "ten", "so_luong_binh_luan"]].copy().drop_duplicates().dropna()
5 df = (
6     df.reset_index()
7     .sort_values(["hang_dien_thoai", "so_luong_binh_luan", "ten"], ascending=[False, False, False])
8     .groupby("hang_dien_thoai")
9     .head(3)
10 )
11 df = df[df["hang_dien_thoai"].isin(brands)][["hang_dien_thoai", "so_luong_binh_luan", "ten"]].reset_index(drop=True)
12 df["order"] = pd.Categorical(df["hang_dien_thoai"], categories=brands, ordered=True).codes + 1
13 df = df.sort_values("order").reset_index(drop=True).drop(columns="order")
14 df = df[["hang_dien_thoai", "ten", "so_luong_binh_luan"]]
15 df.columns = ["Dòng điện thoại", "Tên điện thoại", "Số lượng bình luận"]
16 df
```

TRỰC QUAN HÓA

Dòng điện thoại	Tên điện thoại	Số lượng bình luận
Xiaomi	Điện thoại Xiaomi Redmi K20 Pro (Snap 855)	5490
Xiaomi	Điện thoại Xiaomi Redmi Note 8 Pro (Helio G90T)	4730
Xiaomi	Điện thoại Xiaomi Redmi K20	3639
Realme	Điện thoại Realme X2 Pro	2870
Realme	Điện thoại Realme X (Camera Popup)	2934
Realme	Điện thoại Realme Q (Realme 5 Pro)	4514
iPhone	Điện thoại iPhone XS cũ (Chính hãng)	1763
iPhone	Điện thoại iPhone 7 Plus Lock Nhật, Mỹ (Dùng như Quốc tế)	1388
iPhone	Điện thoại iPhone XR cũ (Chính hãng)	1239
Samsung	Điện thoại Samsung Galaxy Note 9 cũ (128GB - 512GB)	805
Samsung	Điện thoại Samsung Galaxy A90 5G Cũ (Snapdragon 855)	1359
Samsung	Điện thoại Samsung Galaxy S10 5G cũ (99,9%) (Màn 6.7" 2K+)	801
Vivo	Điện thoại Vivo iQOO Z1x	1697
Vivo	Điện thoại Vivo iQOO Neo 855	3873
Vivo	Điện thoại Vivo iQOO Neo	1100
LG	Điện thoại LG V50S ThinQ 5G cũ (Snapdragon 855, Sạc 21W)	806
LG	Điện thoại LG G8 ThinQ Cũ (Snapdragon 855, màn P-OLED 2K)	1090
LG	Điện thoại LG G7 ThinQ cũ (Mỹ, Hàn Quốc)	1191
Vsmart	Điện thoại Vsmart Live 4	434
Vsmart	Điện thoại Vsmart Joy 4	466
Vsmart	Điện thoại Vsmart Live (Chính Hãng)	1505
Asus	Điện thoại Asus ROG Phone 2 (Snapdragon 855+)	3508
Asus	Điện thoại Asus ROG Phone 5 5G (Snapdragon 888)	797
Asus	Điện thoại Asus ROG Phone 3 Tencent	1022
OnePlus	Điện thoại OnePlus 8T	566
OnePlus	Điện thoại OnePlus 7 Pro cũ (99%)	1481
OnePlus	Điện thoại OnePlus 8	470
Nubia	Điện thoại Nubia Red Magic 8 Pro 5G (Snapdragon 8 Gen 2)	309
Nubia	Điện thoại Nubia Red Magic 5G (Gaming Phone)	585
Nubia	Điện thoại Nubia Red Magic 7 (Snapdragon 8 Gen 1, Màn AMOLED 165Hz)	325

CÂU HỎI 5

Trong các thông số kỹ thuật của một chiếc điện thoại, những thông số nào thường có ảnh hưởng lớn nhất đến giá bán?



TIỀN XỬ LÝ

```
1 data_copy = data.dropna(subset=["gia_moi"]).copy() # Tạo một copy df nhưng giá trị "gia_moi" không có Nan
2 data_copy = data_copy[["gia_moi", "he_dieu_hanh", "cpu", "ram", "bo_nho_trong", "dung_luong_pin",
3                         "loai_man_hinh", "kich_thuoc_man_hinh", "tan_so_quet", "do_sang_man_hinh",
4                         "so_the_sim", "loai_pin", "cong_suат_sac", "do_phan_giai_cam_truoc",
5                         "do_phan_giai_cam_sau", "thiet_ke"]] # Chỉ lấy những thuộc tính thông số kỹ thuật
6 data_copy.columns = ["Giá", "Hệ điều hành", "CPU", "RAM", "Bộ nhớ trong", "Dung lượng PIN",
7                       "Loại màn hình", "Kích thước màn hình", "Tần số quét", "Độ sáng màn hình",
8                       "Số thẻ SIM", "Loại PIN", "Công suất sạc", "Độ phân giải camera trước",
9                       "Độ phân giải camera sau", "Thiết kế"] # Đặt lại tên cột
10 data_copy.info() # Đặt lại tên cột
```

TIỀN XỬ LÝ

```
1 # Tạo DataFrame mới để xử lý cột "Hệ điều hành" (không có nan)
2 df = pd.DataFrame({"Hệ điều hành": data_copy["Hệ điều hành"].astype(str).copy()})
3
4 # Tách cột "Hệ điều hành" thành hai cột Brand và Version
5 df[["Brand", "Version"]] = df["Hệ điều hành"].str.split(" ", n=1, expand=True)
6
7 # Chuyển đổi Version thành số thực
8 df["Version"] = (
9     df["Version"]
10    .str.extract(r"(\d+\.\d+|\d+)")
11    .astype(float)
12 )
13
14 # Tính giá trị số hóa của "Hệ điều hành"
15 data_copy["Hệ điều hành"] = df["Version"] + df["Brand"].apply(lambda x: 0.5 if x == "Android" else 0)
16
17 # Lý do: Android thường dễ sử dụng hơn iOS nên ta gán giá trị 0.5 cho Android và 0 cho iOS
```

TIỀN XỬ LÝ

```
1 # Tạo một df mới chứa các thông số của CPU (có Nan)
2 df = pd.DataFrame({"CPU": data_copy["CPU"].astype(str).copy()})
3
4 # Tạo cột Process chứa giá trị nm của CPU
5 df['Process'] = df['CPU'].str.extract(r'\\((\d+)\s*nm\\)')[0].astype(float) # Trích xuất Process
6 df['Process'] = df['Process'].fillna(df['Process'].mean()) # Fillna với giá trị trung bình
7
8 # Hàm tính Clock_Speed
9 def calculate_clock_speed(row):
10     if isinstance(row, str):
11         core_info = re.findall(r'(\d+)x([\d.]+)\s*GHz', row)
12         return sum(int(cores) * float(speed) for cores, speed in core_info)
13     return 0
14
15 # Tạo cột Clock_Speed chứa giá trị clock_Speed của CPU
16 df['Clock_Speed'] = df['CPU'].apply(calculate_clock_speed) # Tính Clock_Speed
17 df['Clock_Speed'] = df['Clock_Speed'].replace(0, df['Clock_Speed'].mean()) # Fill các giá trị 0 thành giá trị mean
18
19 # Số hóa thuộc tính CPU
20 data_copy['CPU'] = df['clock_Speed'] / df['Process']
21
22 # Lý do: Tất cả các chip đều có những thuộc tính chung như: số nhân, tốc độ xử lý, và số tiến trình sản xuất,
23 # do đó chúng có thể được số hóa để tiện phân tích.
24 # - Tiến trình sản xuất (Process): Giá trị càng nhỏ thì chip càng cao cấp.
25 # - Tốc độ xử lý (Clock Speed): Giá trị càng lớn thì hiệu năng càng cao.
26
```

TIỀN XỬ LÝ

```
1 # Tính giá trị số hóa cho 'Loại màn hình' (có Nan)
2 freq_encoding = data_copy['Loại màn hình'].value_counts().to_dict()
3 data_copy['Loại màn hình'] = data_copy['Loại màn hình'].map(freq_encoding)
4
5 # Lý do: AMOLED có giá trị cao hơn OLED, trong khi LCD có giá trị thấp hơn cả hai loại màn hình này.
```

```
1 # Tính giá trị số hóa cho 'Loại Pin' (có Nan)
2 battery_value = {'Li-Po': 2, 'Li-Ion': 1, 'Si/C': 3}
3 data_copy['Loại PIN'] = data_copy['Loại PIN'].map(battery_value)
4
5 # Lý do: Loại pin Li-Po thường có tuổi thọ cao hơn so với Li-Ion, còn Si/c là loại pin mới nhất và có tuổi thọ cao nhất.
```

```
1 # Hàm số hóa camera
2 def encode_camera_resolution(camera_list):
3     if not camera_list: # Nếu danh sách trống
4         return 0
5     total_resolution = sum(camera_list) # Tổng độ phân giải
6     additional_value = 1 / len(camera_list) # Giá trị cộng thêm
7     return total_resolution + additional_value
8
9 # Tính giá trị số hóa cho 'Độ phân giải camera trước' và 'Độ phân giải camera sau' (không có Nan)
10 data_copy['Độ phân giải camera trước'] = data_copy['Độ phân giải camera trước'].apply(encode_camera_resolution)
11 data_copy['Độ phân giải camera sau'] = data_copy['Độ phân giải camera sau'].apply(encode_camera_resolution)
12
13 # Lý do: Độ phân giải càng cao thì chất lượng ảnh càng tốt, do đó ta có thể số hóa các giá trị này để dễ dàng phân tích.
14 # Một camera có độ phân giải cao có giá trị hơn tổng độ phân giải của các camera có độ phân giải thấp.
```

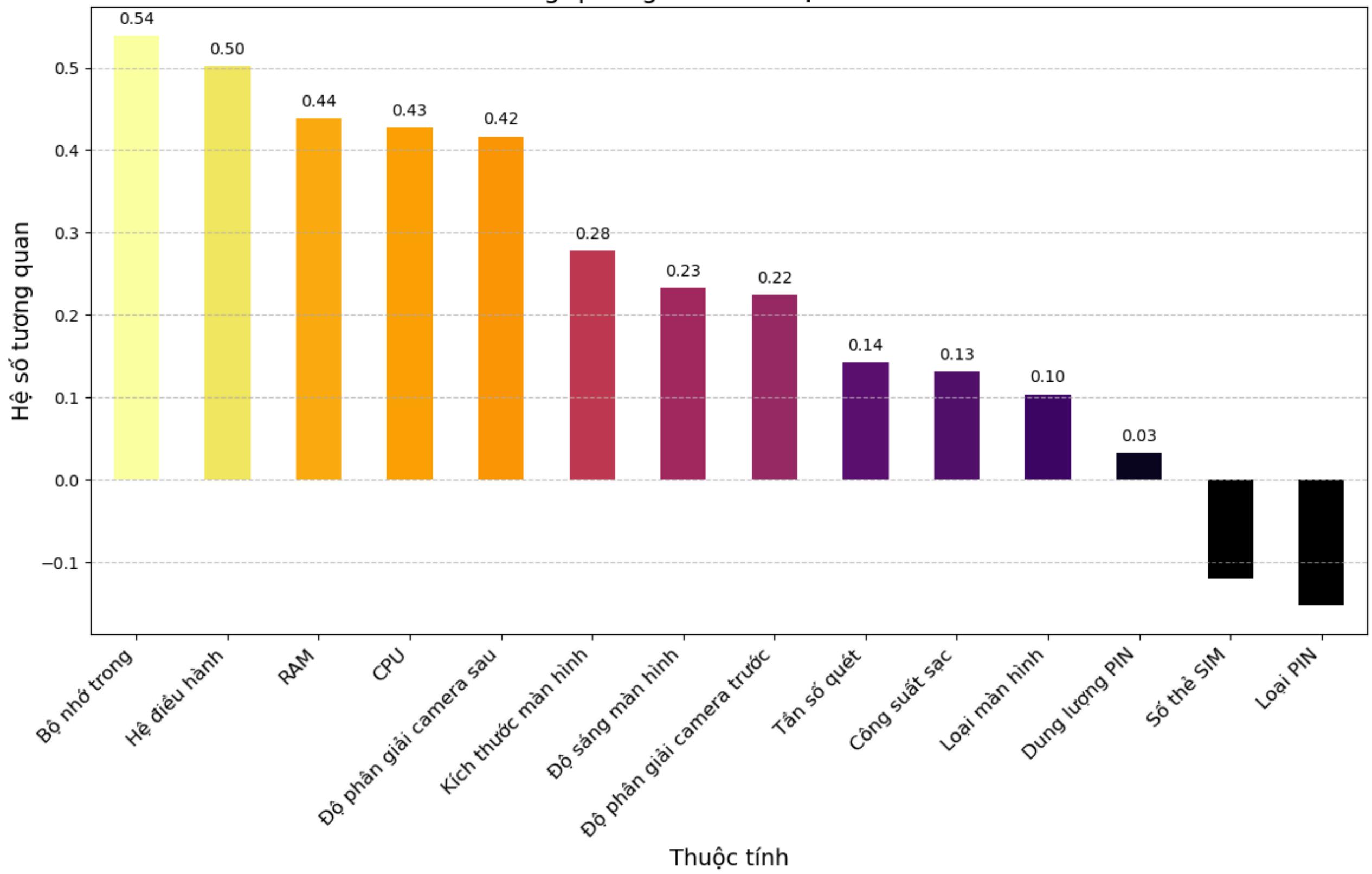
TIỀN XỬ LÝ

```
1 # Tạo một df mới chứa các thông số của "Thiết kế" (có Nan)
2 df = pd.DataFrame({"Thiết kế": data_copy["Thiết kế"].astype(str).copy()})
3 df["Thiết kế"] = df["Thiết kế"].str.lower() # Chuyển tất cả thành chữ thường
4
5 # Tính số giá trị của "Thiết kế"
6 design_value = df["Thiết kế"].value_counts().to_dict()
7
8 # Xóa cột "Thiết kế"
9 data_copy = data_copy.drop(columns=["Thiết kế"])
10
11 # Lý do: Thiết kế của điện thoại bao gồm các yếu tố như vật liệu, khung, họa tiết, kiểu dáng, và các tính năng bổ trợ đặc biệt, v.v.
12 # Kiểu dữ liệu này không tuân thủ theo một quy tắc cụ thể nào, do đó ta không thể số hóa nó một cách chính xác.
13 # Vì vậy, ta sẽ xóa cột này và việc đánh giá thiết kế sẽ được thực hiện sau.
```

```
1 # Xử lý cột "SIM"
2 data_copy['Số thẻ SIM'] = data_copy['Số thẻ SIM'].fillna(data_copy['Số thẻ SIM'].min()) # Fillna với giá trị nhỏ nhất
3
4 # Xử lý tất cả các cột còn lại với giá trị mean
5 data_copy = data_copy.fillna(data_copy.mean())
```

TRỰC QUAN HÓA

Tương quan giữa các thuộc tính và Giá



CÂU HỎI 6



Kiểu thiết kế điện thoại nào phổ biến nhất hiện nay, dựa trên các kiểu thiết kế của các mẫu điện thoại hiện có trong cửa hàng?

TIỀN XỬ LÝ

```
1 # Tạo ra một DataFrame mới chứa các thuộc tính 'Thiết kế'  
2 design_df = pd.DataFrame({'Thiết kế': data['thiet_ke'].astype(str).copy()})  
3  
4 # Xử lý cột 'Thiết kế'  
5 design_df['Thiết kế'] = design_df['Thiết kế'].dropna().str.replace(' + ', ' ', regex=False)  
6 design_df['Thiết kế'] = design_df['Thiết kế'].str.replace('\r\n', ' ', '').str.split(',')  
7 design_df['Thiết kế'] = design_df['Thiết kế'].apply(lambda x: [i.lower() for i in x])  
8 design_df['Thiết kế'] = design_df['Thiết kế'].apply(lambda x: [i.strip() for i in x])  
9  
10 n = design_df.explode('Thiết kế')['Thiết kế'].value_counts()  
11  
12 print(n)
```

TRỰC QUAN HÓA

Word Cloud - Thiết kế



CÂU HỎI 7

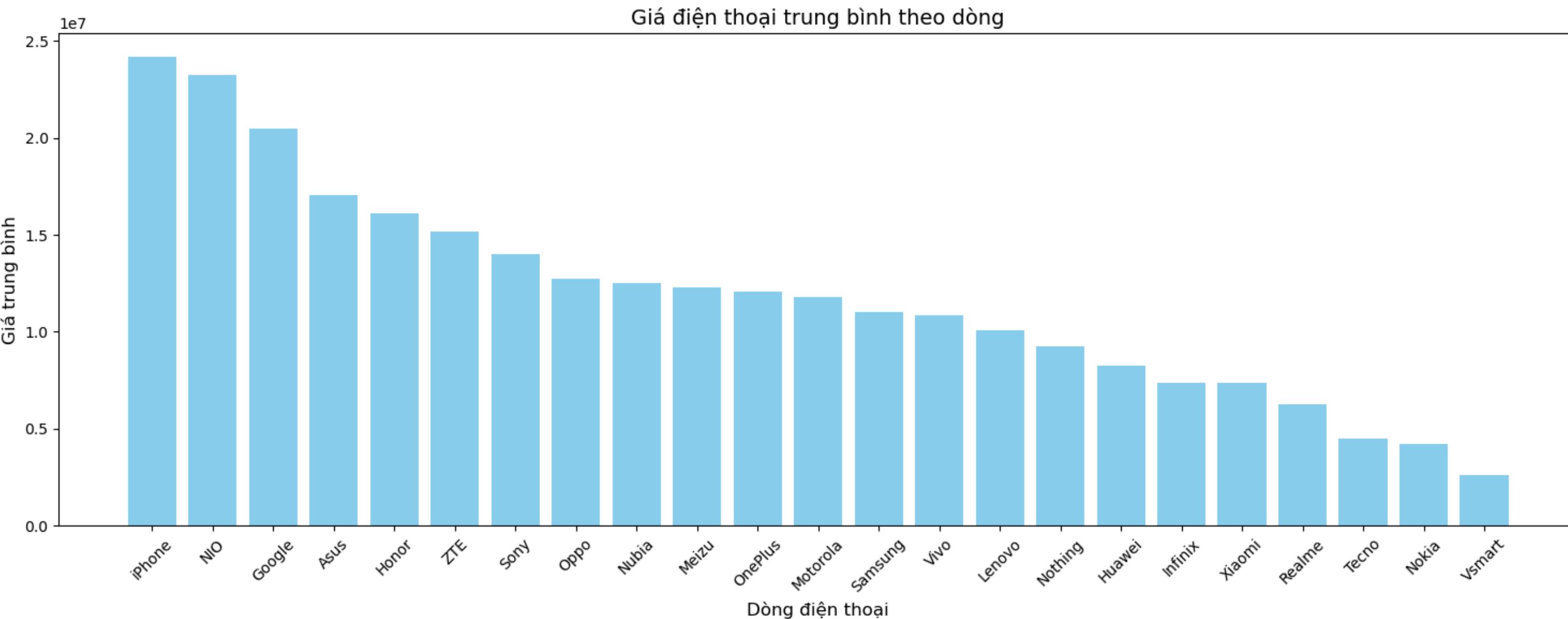
Phân bổ giá bán của các hãng điện thoại được thể hiện như thế nào?



TIỀN XỬ LÝ

```
1 #Không lấy những điện thoại cũ
2 data_new = data[~data['ten'].str.contains('cũ', na=False)]
3 df = data_new[["hang_dien_thoai", "gia_moi"]].copy().dropna()
4 df = df.groupby("hang_dien_thoai").agg(['mean', 'count']).sort_values(('gia_moi', 'mean'), ascending=False).reset_index()
5 df.columns = ['hang_dien_thoai', 'mean', 'count']
6 df = df[df['count'] > 10]
7 df
```

TRỰC QUAN HÓA



XÂY DỰNG CÁC MÔ HÌNH HỌC MÁY

Vấn đề được áp dụng học máy:

Dự đoán giá điện thoại dựa trên các đặc trưng quan trọng và cần thiết.

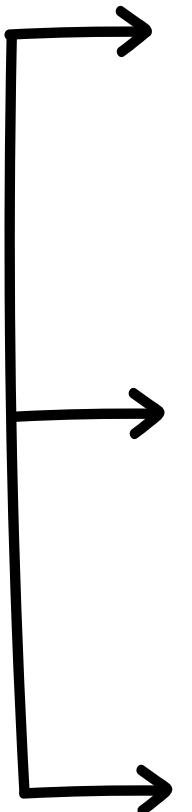


LỰA CHỌN CÁC MÔ HÌNH

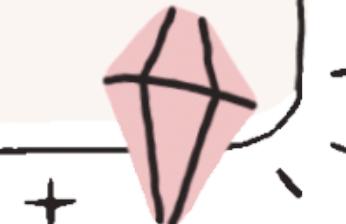
XGBRegressor

DecisionTreeRegressor

RandomForestRegressor



- Khả năng xử lý dữ liệu phức tạp
- Hiệu suất cao
- Khả năng điều chỉnh
- Xử lý dữ liệu không đồng nhất
- Xử lý outliers tốt



CÁC THƯ VIỆN CẦN THIẾT

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from xgboost import XGBRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

import ast
import time
```

TIỀN XỬ LÝ DỮ LIỆU

Loại bỏ các cột không ảnh hưởng đến giá:

- `ten` là tên của các mẫu điện thoại.

```
df = df.drop(columns=[ 'ten' ]) ...
```

- `duong_dan` là liên kết đến trang thông tin của điện thoại trên website.

```
df = df.drop(columns=[ 'duong_dan' ]) ...
```

- `loai_dien_thoai` lưu thông tin cụ thể hơn cho `hang_dien_thoai`, không ảnh hưởng đến giá.

```
df = df.drop(columns=[ 'loai_dien_thoai' ]) ...
```

- `mau_sac` lưu các màu sắc khác nhau của từng mẫu điện thoại, không ảnh hưởng đến giá.

```
df = df.drop(columns=[ 'mau_sac' ]) ...
```

- `thiet_ke` mô tả cấu tạo của một chiếc điện thoại, không ảnh hưởng đến giá.

```
df = df.drop(columns=[ 'thiet_ke' ]) ...
```

TIỀN XỬ LÝ DỮ LIỆU

	Features	Missing ratio
0	gia_cu	43.3
1	do_sang_man_hinh	42.5
2	loai_pin	27.9
3	tan_so_quet	24.6
4	gia_moi	19.1
5	bo_nho_trong	7.4
6	cong_suат_sac	7.3
7	kich_thuoc_man_hinh	5.7
8	loai_man_hinh	5.4
9	dung_luong_pin	2.3
10	ram	0.5
11	thoi_gian_bao_hanh	0.0
12	do_phan_giai_cam_sau	0.0
13	so_the_sim	0.0
14	la_dien_thoai_cu	0.0
15	danh_gia	0.0
16	hang_dien_thoai	0.0
17	he_dieu_hanh	0.0
18	so_luong_binh_luan	0.0
19	do_phan_giai_cam_truoc	0.0

→ Xóa

Xử lý các cột non-numeric

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1577 entries, 18 to 8532
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype  
--- 
 0   he_dieu_hanh    1577 non-null    object 
 1   hang_dien_thoai 1577 non-null    object 
 2   loai_man_hinh   1577 non-null    object 
 3   loai_pin         1577 non-null    object 
 4   do_phan_giai_cam_sau 1577 non-null    object 
 5   do_phan_giai_cam_truoc 1577 non-null    object 
dtypes: object(6)
memory usage: 86.2+ KB
```

TIỀN XỬ LÝ DỮ LIỆU

Bộ dữ liệu sau khi đã xử lý:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1577 entries, 18 to 8532
Data columns (total 50 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   thoi_gian_bao_hanh    1577 non-null   float64
 1   danh_gia            1577 non-null   float64
 2   so_luong_binh_luan  1577 non-null   int64  
 3   gia_moi             1577 non-null   float64
 4   ram                 1577 non-null   float64
 5   bo_nho_trong        1577 non-null   float64
 6   dung_luong_pin      1577 non-null   float64
 7   la_dien_thoai_cu    1577 non-null   int64  
 8   kich_thuoc_man_hinh 1577 non-null   float64
 9   tan_so_quet         1577 non-null   float64
 10  so_the_sim          1577 non-null   int64  
 11  cong_suat_sac       1577 non-null   float64
 12  do_phan_giai_cam_sau 1577 non-null   float64
 13  do_phan_giai_cam_truoc 1577 non-null   float64
 14  Android             1577 non-null   int64  
 15  iOS                 1577 non-null   int64  
 16  hang_dien_thoai_Asus 1577 non-null   bool   
 17  hang_dien_thoai_BLU  1577 non-null   bool   
 18  hang_dien_thoai_Bphone 1577 non-null   bool   
 19  hang_dien_thoai_Fairphone 1577 non-null   bool  
 ...
 48  loai_pin_Li-Po      1577 non-null   bool   
 49  loai_pin_Si/C       1577 non-null   bool  
dtypes: bool(34), float64(11), int64(5)
memory usage: 261.8 KB
```

TẠO CÁC MÔ HÌNH

Tạo DataFrame X và y:

```
x = df.drop(columns=["gia_moi"])
y = df["gia_moi"]
```

[35]

Chia dữ liệu thành tập huấn luyện và tập kiểm tra:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=1/3, random_state=42)
```

[36]

MODEL: XGBREGRESSOR

Khởi tạo mô hình:

```
[37] model_1 = XGBRegressor()
```

Huấn luyện mô hình và tính thời gian:

```
[38] time_start = time.time()
model_1.fit(X_train, y_train)
time_end = time.time()
```

```
[39] training_time_model_1 = time_end - time_start
print(f"Training time: {training_time_model_1:.2f} seconds")
```

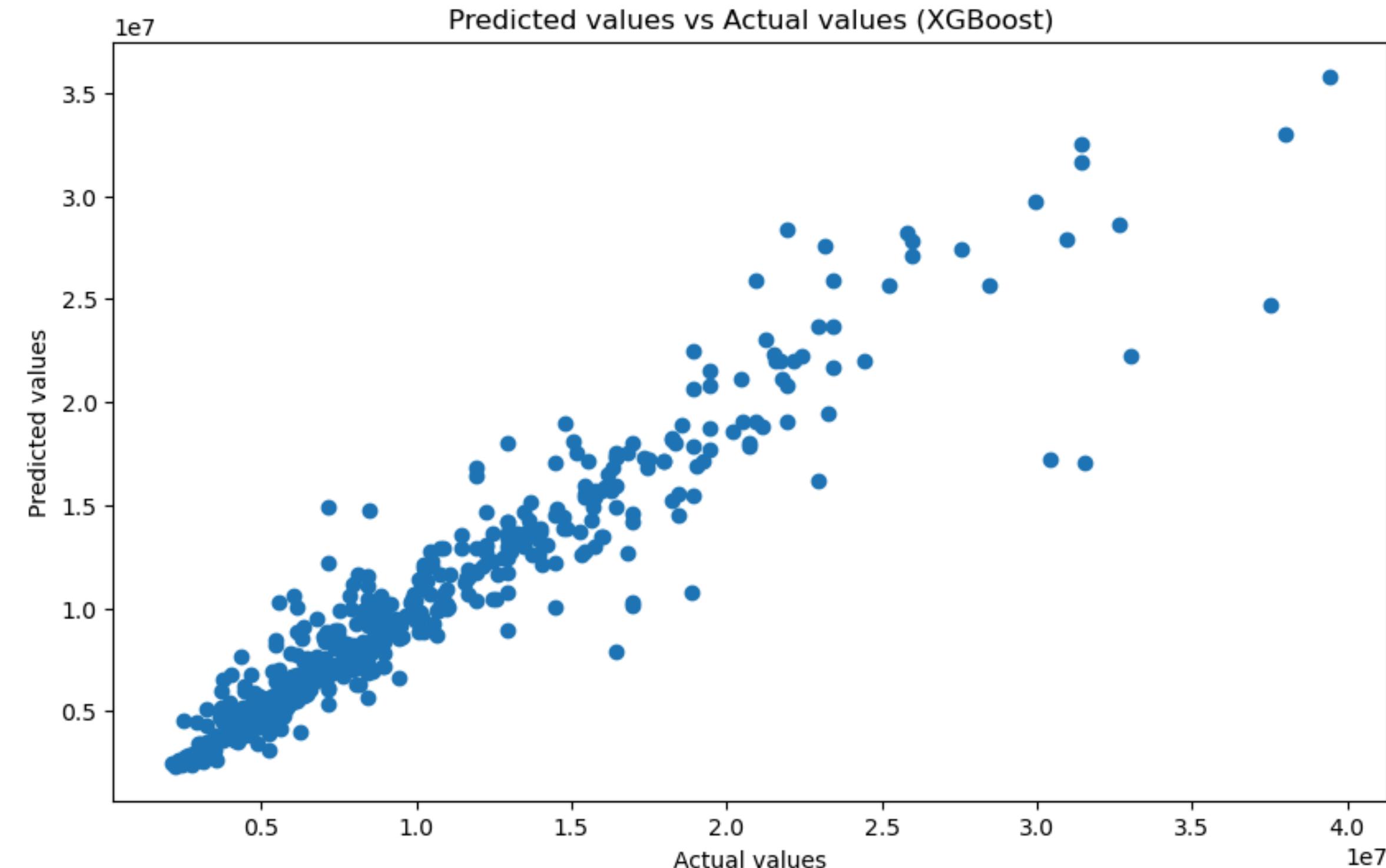
... Training time: 0.15 seconds

Đánh giá mô hình:

Dự đoán giá trị đầu ra trên tập dữ liệu kiểm tra bằng mô hình tốt nhất:

```
[40] y_pred_model_1 = model_1.predict(X_test)
y_pred_model_1 = np.maximum(y_pred_model_1, 0) # Đảm bảo giá không âm
```

MODEL: XGBREGRESSOR



MODEL: DECISION TREEREGRESSOR

Khởi tạo mô hình:

```
[44] model_2 = DecisionTreeRegressor()
```

Huấn luyện mô hình và tính thời gian:

```
[45] time_start = time.time()
model_2.fit(X_train, y_train)
time_end = time.time()
```

```
[46] training_time_model_2 = time_end - time_start
print(f"Training time: {training_time_model_2:.2f} seconds")
```

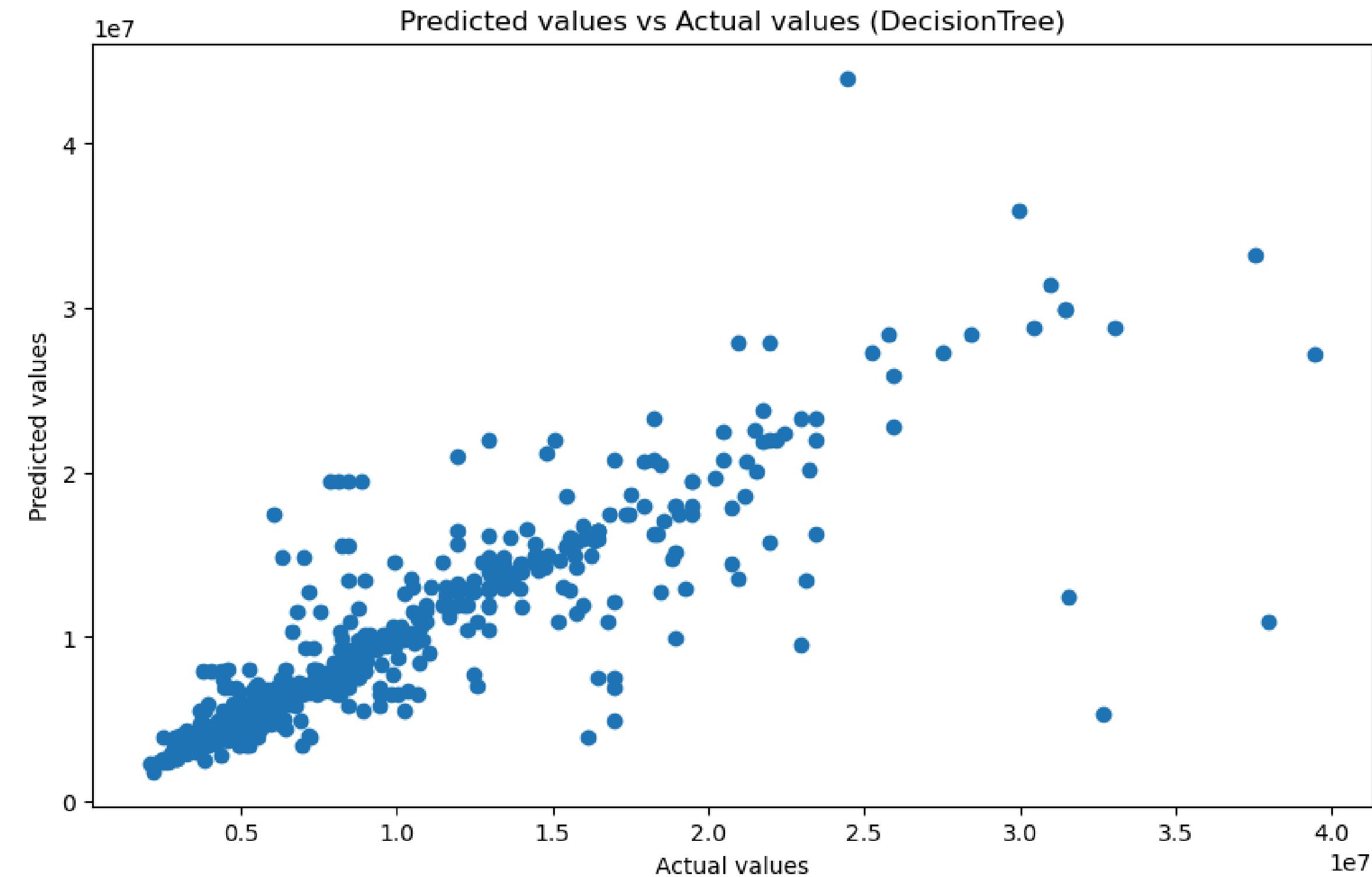
... Training time: 0.01 seconds

Đánh giá mô hình:

Dự đoán giá trị đầu ra trên tập dữ liệu kiểm tra bằng mô hình tốt nhất:

```
[47] y_pred_model_2 = model_2.predict(X_test)
y_pred_model_2 = np.maximum(y_pred_model_2, 0) # Đảm bảo giá không âm
```

MODEL: DECISIONTREEREGRESSOR



MODEL: RANDOMFORESTREGRESSOR

Khởi tạo mô hình:

```
model_3 = RandomForestRegressor()
```

[51]

Huấn luyện mô hình và tính thời gian:

```
time_start = time.time()
model_3.fit(X_train, y_train)
time_end = time.time()
```

[52]

```
training_time_model_3 = time_end - time_start
print(f"Training time: {training_time_model_3:.2f} seconds")
```

[53]

... Training time: 0.57 seconds

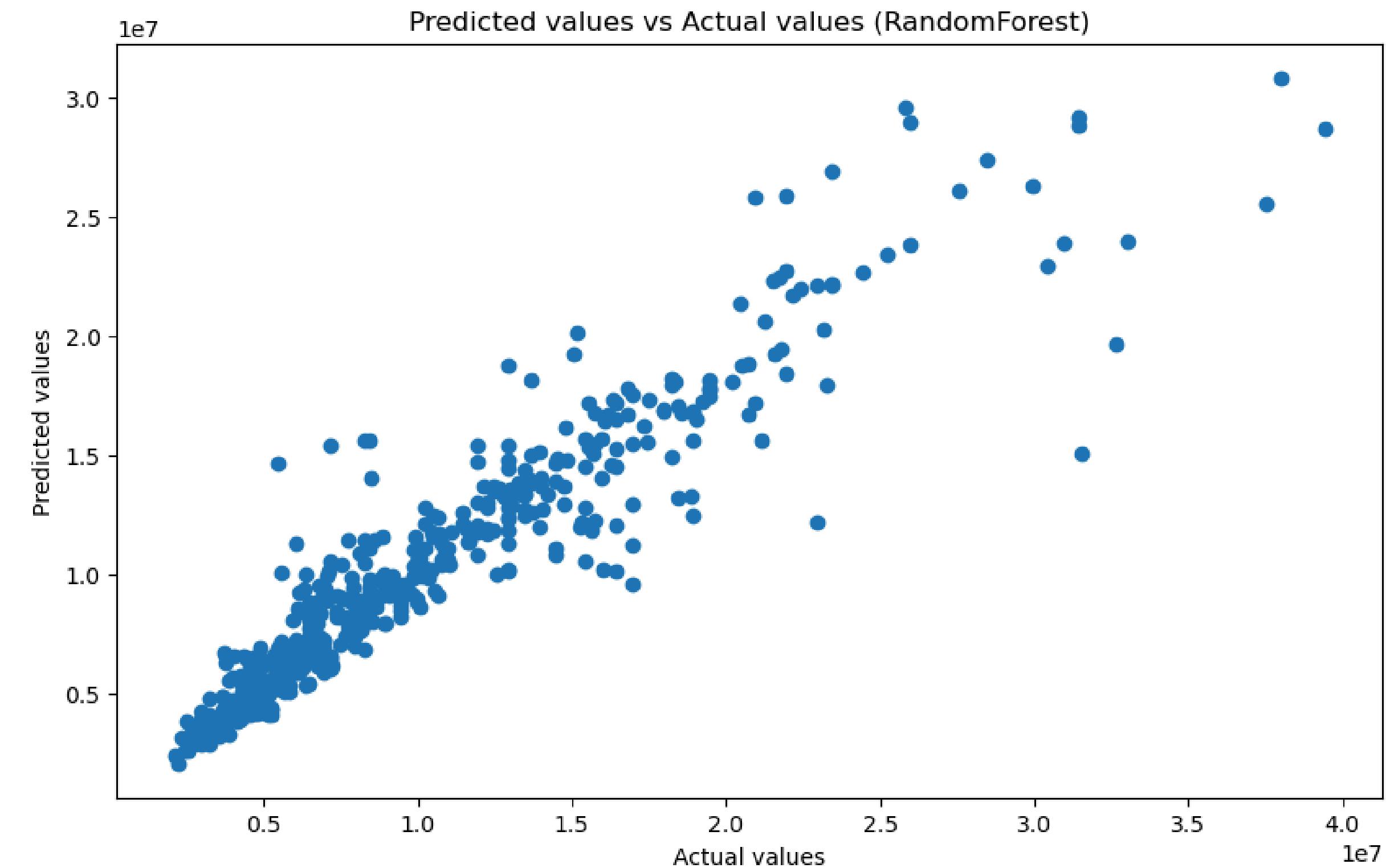
Đánh giá mô hình:

Dự đoán giá trị đầu ra trên tập dữ liệu kiểm tra bằng mô hình tốt nhất:

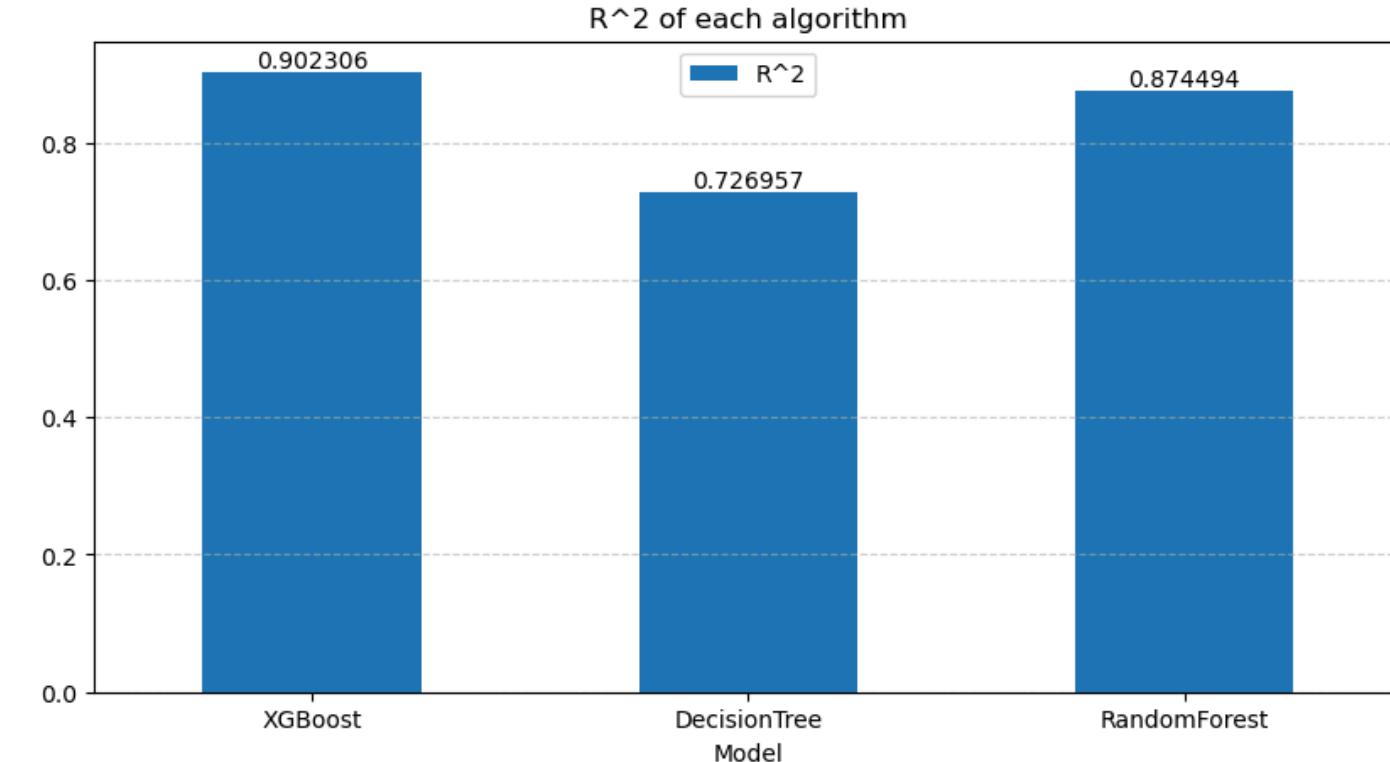
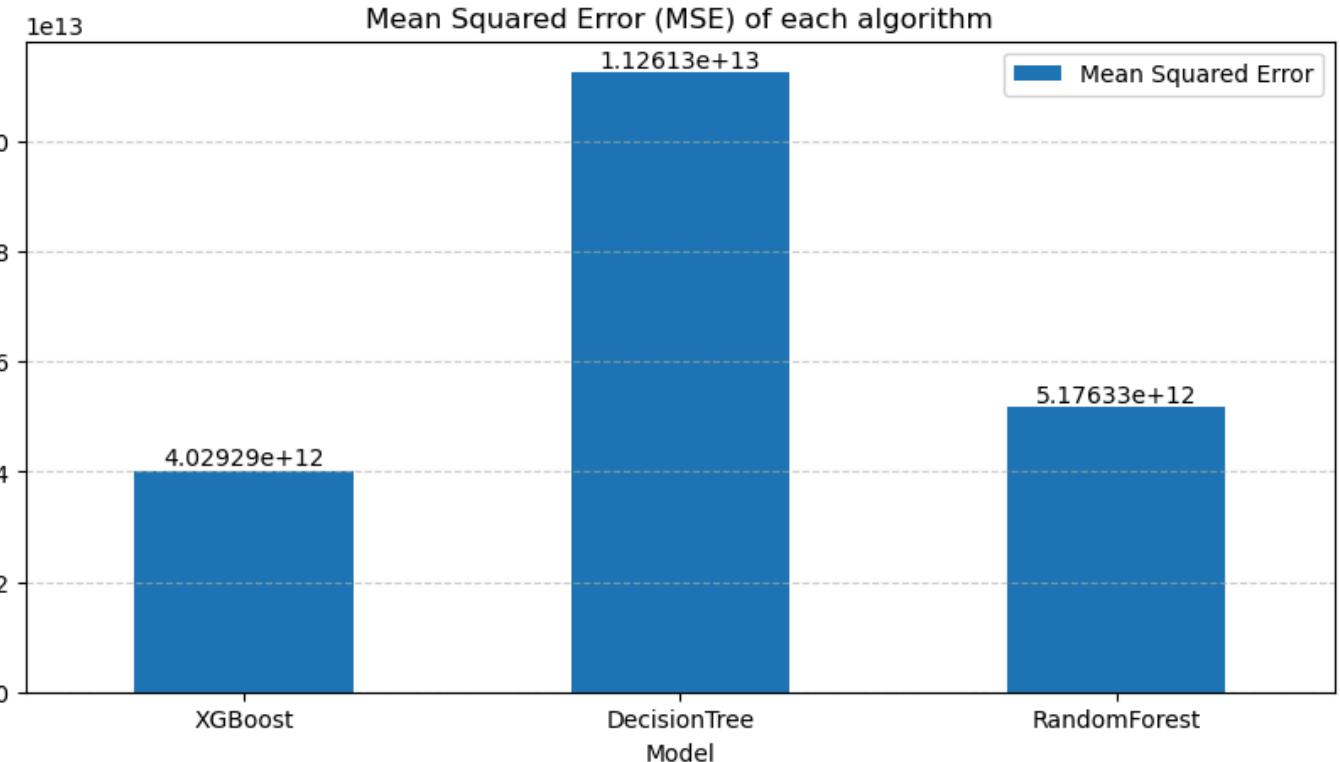
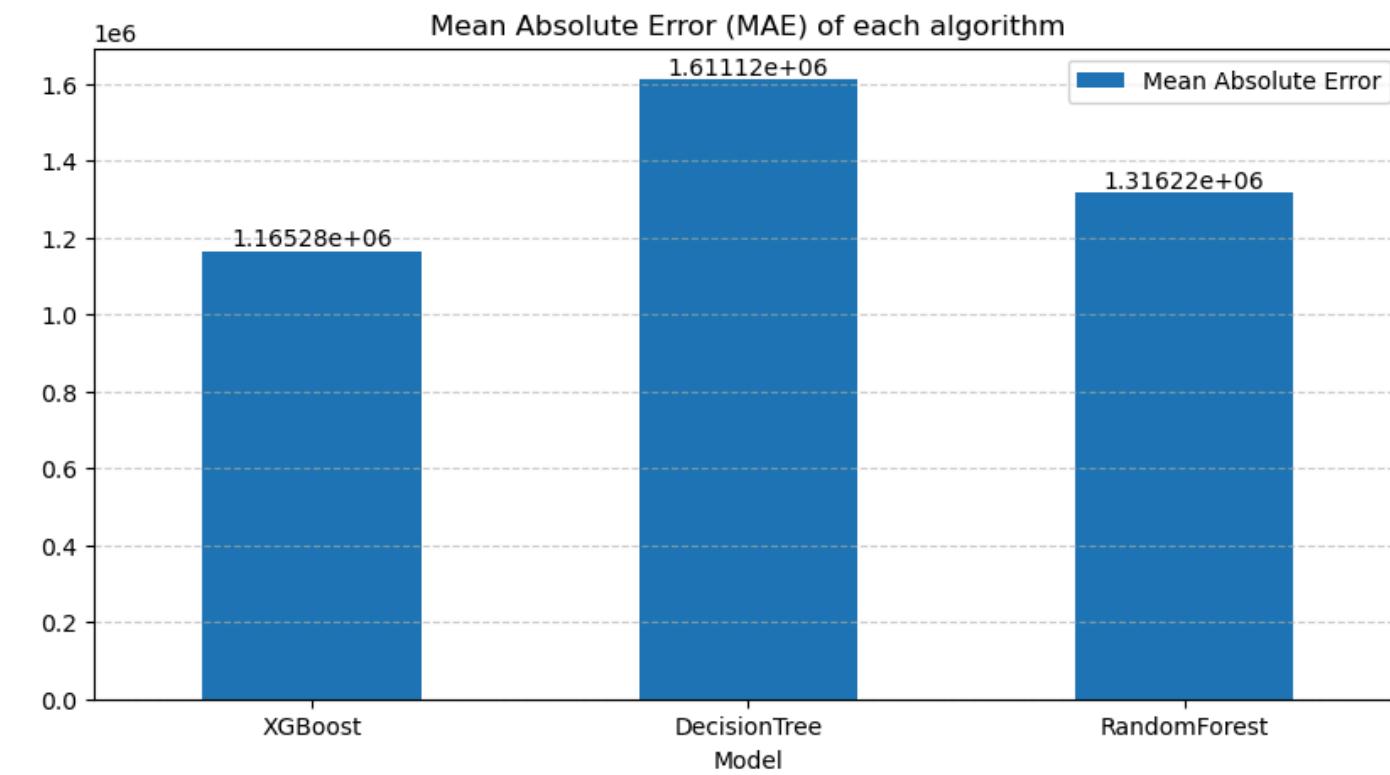
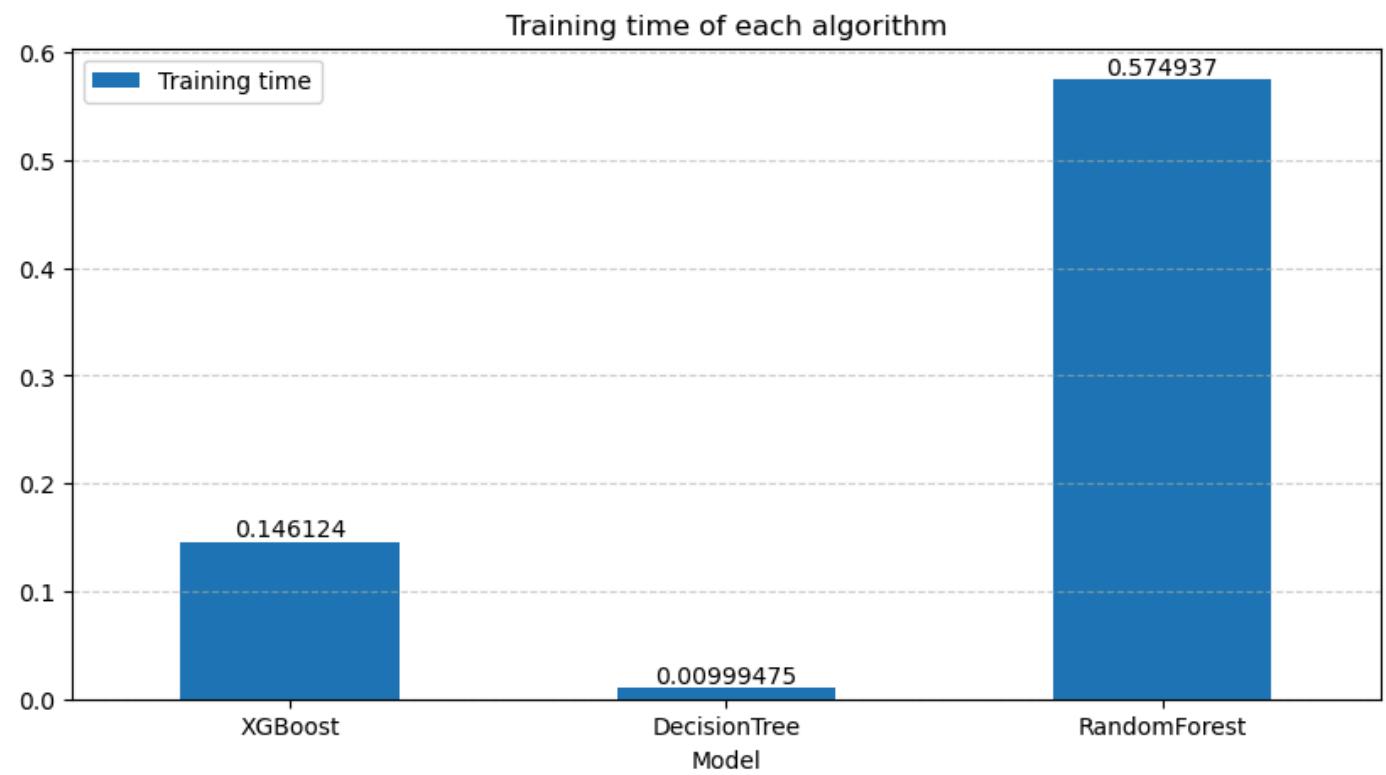
```
y_pred_model_3 = model_3.predict(X_test)
y_pred_model_3 = np.maximum(y_pred_model_3, 0) # Đảm bảo giá không âm
```

[54]

MODEL: RANDOMFORESTREGRESSOR



ĐÁNH GIÁ CÁC MÔ HÌNH



KẾT LUẬN

**Trong 3 thuật toán đã triển khai,
XGBRegressor là thuật toán tối
ưu nhât cho bài toán dự đoán giá
điện thoại.**



TỐI ƯU MÔ HÌNH

```
1 # Tách thêm tập validation từ tập train
2 X_train_split, X_val, y_train_split, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
3
4 # Định nghĩa mô hình XGBRegressor
5 model = XGBRegressor(
6     objective="reg:squarederror",      # Hồi quy squared error
7     learning_rate=0.2,                # Learning rate
8     max_depth=6,                    # Độ sâu cây
9     seed=42,                        # Để tái lập kết quả
10    n_estimators=1,                 # Bắt đầu với 1 vòng (huấn luyện thủ công)
11 )
12
13 # Khởi tạo các biến để kiểm soát early stopping
14 n_estimators = 100                # Số vòng tối đa
15 early_stopping_rounds = 10         # Số vòng không cải thiện trước khi dừng
16 best_rmse = np.inf                 # Giá trị RMSE tốt nhất ban đầu
17 no_improve_rounds = 0             # Đếm số vòng không cải thiện
18
```



TỐI ƯU MÔ HÌNH

```
19 # Huấn luyện từng vòng
20 for i in range(1, n_estimators + 1):
21     model.n_estimators = i          # Cập nhật số lượng vòng boosting
22     model.fit(X_train_split, y_train_split, verbose=False)
23
24     # Dự đoán và đánh giá trên tập validation
25     y_val_pred = model.predict(X_val)
26     rmse = root_mean_squared_error(y_val, y_val_pred)
27     print(f"Iteration {i}: Validation RMSE = {rmse:.4f}")
28
29     # Kiểm tra cải thiện
30     if rmse < best_rmse:
31         best_rmse = rmse
32         no_improve_rounds = 0
33     else:
34         no_improve_rounds += 1
35
36     # Dừng nếu không cải thiện trong `early_stopping_rounds` vòng
37     if no_improve_rounds >= early_stopping_rounds:
38         print(f"Early stopping at iteration {i}, best RMSE = {best_rmse:.4f}")
39         break
40
41 # Đánh giá mô hình cuối cùng trên tập test
42 y_pred = model.predict(X_test)
43 y_pred = np.maximum(y_pred, 0) # Đưa các giá trị âm về 0
44
```



TỐI ƯU MÔ HÌNH

Test RMSE: 1621403.1351

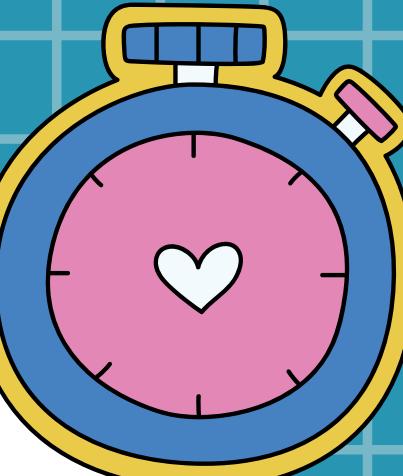
Test MAE: 1020102.7619

Test MSE: 2628948126351.7363

Test R^2: 0.9432



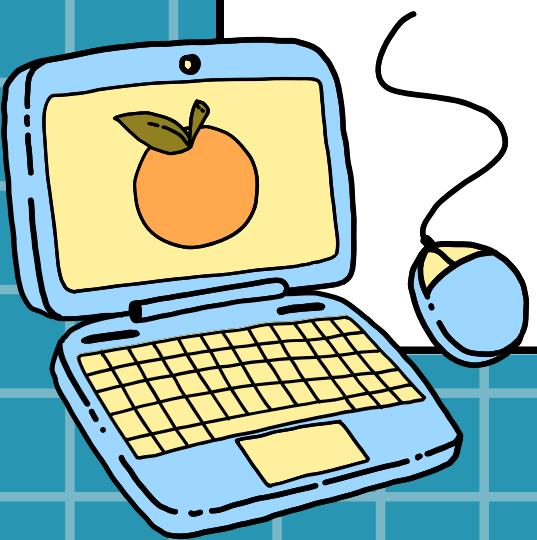
ĐÁNH GIÁ



Nhóm 02 đã hoàn thành đồ án theo đúng nội dung và thời hạn quy định trong đề cương.

Tuy nhiên, trong quá trình thực hiện, chúng tôi vẫn gặp một số sai sót và khó khăn, đã gây ảnh hưởng đến thời gian của Thầy và các bạn.

Chúng em xin chân thành cảm ơn Thầy đã hướng dẫn nhiệt tình và trao đổi kỹ lưỡng, giúp nhóm đạt được kết quả tốt nhất trong đồ án này!



TÀI LIỆU THAM KHẢO

[1] Tài liệu học tập trong môn học

URL: Nhập môn khoa học dữ liệu - CQ2022/21

[2] Thư viện pandas.

URL: <https://pandas.pydata.org/docs/>.

[3] Thư viện Seaborn.

URL: <https://seaborn.pydata.org/>.

[4] Thư viện Matplotlib.

URL: <https://matplotlib.org/>.

[5] Tutorial Data Science.

URL: <https://github.com/academic/awesome-datascience>.

[6] Thư viện hỗ trợ học máy.

URL: <https://scikit-learn.org/stable/>.



CẢM ƠN

Thầy và các bạn đã lắng nghe!

