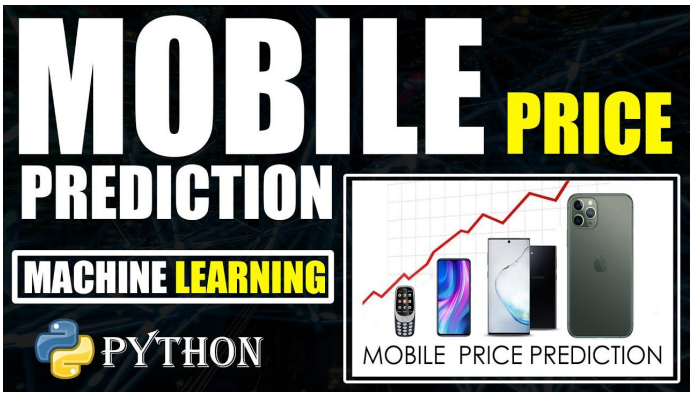
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ
HỌC PHẦN: KHOA HỌC DỮ LIỆU



DỰ ĐOÁN GIÁ ĐIỆN THOẠI

HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN	ĐIỂM BẢO VỆ
Ngô Văn Đông	19N13	
Huỳnh Thị Khánh Linh	19N13	
Nguyễn Minh Dũng	19N13	

ĐÀ NẴNG, 06/2022

TÓM TẮT

Bộ dữ liệu bao gồm các thông tin, thông số kỹ thuật, giá bán của điện thoại di động, được thu thập từ 4 trang web bán các sản phẩm công nghệ lớn ở Việt Nam. Bộ dữ liệu này được dùng để dự đoán giá điện thoại với các thông số kỹ thuật được đưa ra. Dữ liệu sau khi thu thập là các chuỗi ký tự trong đó có các đại lượng cần dùng cho mô hình dự đoán. Khi làm sạch dữ liệu, cần trả về đúng kiểu dữ liệu, thay thế các giá trị dữ liệu trống. Để lựa chọn đặc trưng phù hợp cho mô hình dự đoán, sau khi xử lý dữ liệu ngoại lệ cũng như chuẩn hóa dữ liệu, ta tiến hành trực quan hóa sự tương quan của các đặc trưng so với đặc trưng mục tiêu. Và chọn lựa ra các đặc trưng phù hợp cho mô hình dự đoán. Mô hình dự đoán được xây dựng bởi Linear Regression và cải tiến độ chính xác bằng mô hình XGBoost Regression. Đánh giá hiệu quả của mô hình dự đoán sử dụng RMSE, MAE và R2.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá theo 3 mức
Ngô Văn Đông	- Thu thập dữ liệu - Thống kê mô tả trực quan về dữ liệu	- Đã hoàn thành - Đã hoàn thành
Huỳnh Thị Khánh Linh	- Làm sạch dữ liệu, xử lý dữ liệu trống - Xử lý ngoại lệ, chuẩn hóa dữ liệu - Lựa chọn đặc trưng	- Đã hoàn thành - Đã hoàn thành - Đã hoàn thành
Nguyễn Minh Dũng	- Mô hình hóa dữ liệu - Đánh giá hiệu quả mô hình - Kết luận	- Đã hoàn thành - Đã hoàn thành - Đã hoàn thành

MỤC LỤC

DANH SÁCH HÌNH ẢNH

DANH SÁCH BẢNG BIỂU

NỘI DUNG BÁO CÁO

1. Giới thiệu

Mô hình dự đoán giá điện thoại di động dựa trên các thông số kỹ thuật, đặc trưng của điện thoại sẽ giúp các công ty công nghệ ước tính được giá bán điện thoại phù hợp để cạnh tranh với các công ty công nghệ khác. Ngoài ra, mô hình này cũng giúp cho khách hàng xác định được giá tốt nhất cho điện thoại họ muốn mua với các thông số cụ thể.

2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

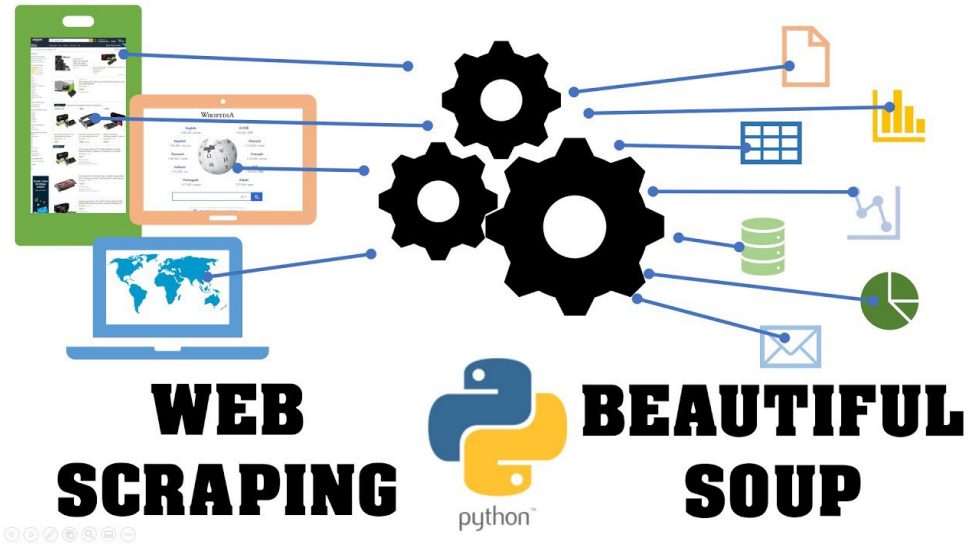
2.1.1 Nguồn dữ liệu

Dữ liệu được thu thập từ 4 trang web bán các sản phẩm công nghệ lớn ở Việt Nam.

- Thế giới di động
- CellphoneS
- Hoàng Hà mobile
- Nguyễn Kim

2.1.2 Công cụ thu thập

Thu thập dữ liệu từ 4 trang web sử dụng **Beautiful Soup**. Đây là một thư viện Python dùng để chuyển đổi HTML và XML. Nó tạo ra một cây phân tích của trang web dùng để trích xuất dữ liệu từ HTML giúp ích cho việc cào dữ liệu.

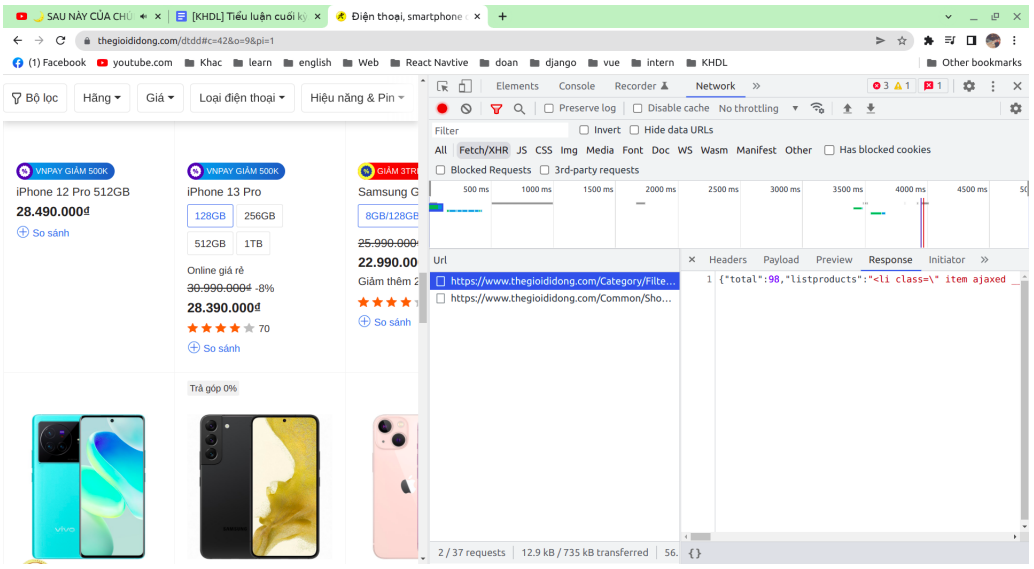


Hình 01: Mô hình Web Scraping sử dụng Beautiful Soup

2.1.3 Cách thức sử dụng công cụ

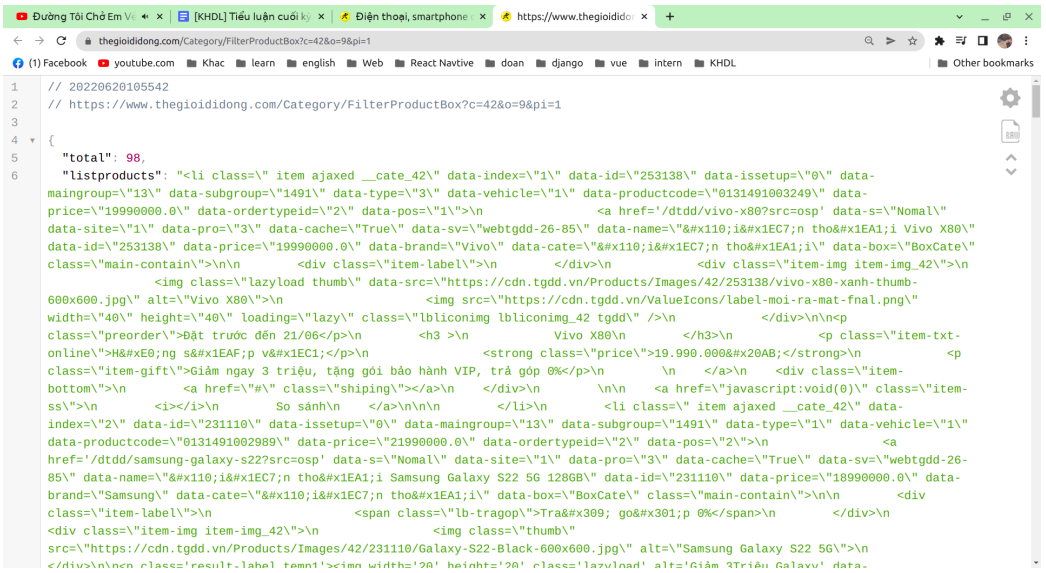
Ví dụ minh họa cho trang web thế giới di động

Bước 1: Lấy ra tất cả link của các sản phẩm



Hình 02: Kiểm tra các tài nguyên trong website của thegioididong.com

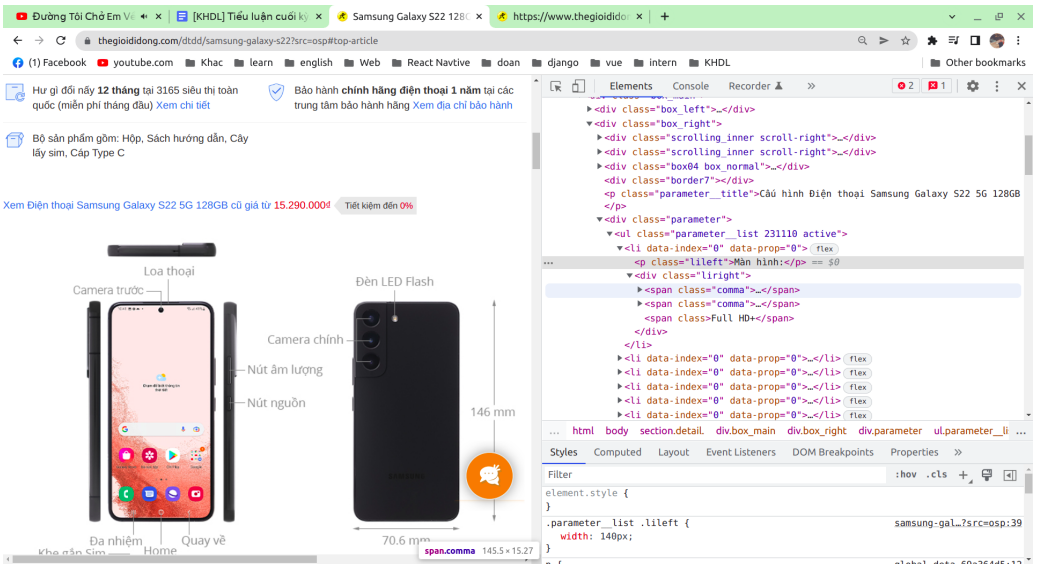
Sau khi kiểm tra thấy được trang web gọi đến một api.



Hình 03: API của thegioididong.com

Trích xuất từ json để lấy được link của tất cả sản phẩm.

Bước 2: Phân tích cấu trúc trang web của từng sản phẩm



Hình 04: Phân tích cấu trúc website thegioididong.com

Tìm kiếm class và id của các thẻ chứa thông tin về thông số kỹ thuật của điện thoại

Bước 3: Trích xuất thông tin và lưu vào file csv

Source code:

Title

```
def getSoup(url):
    """Hàm tải về một trang web từ một url và trả về một cây HTML"""
    # dùng header của postman để không bị từ chối
    headers={'User-Agent':'PostmanRuntime/7.29.0'}
    page = requests.get(url,headers=headers)
    return bs4.BeautifulSoup(page.text,"html.parser")
```

Title

```
# lấy ra thẻ table đầu tiên có id là tskt
table=soup.find('table',id="tskt")
# lấy ra tất cả các thẻ tr có trong table
trs=table.findAll('tr')

```

2.1.4 Ví dụ minh họa

Title

```
import bs4
import pandas as pd
import requests

def getSoup(url):
  """Hàm tải về một trang web từ một url và trả về một cây HTML"""
  # dùng header của postman để không bị từ chối
  headers={'User-Agent':'PostmanRuntime/7.29.0'}
  page = requests.get(url,headers=headers)
  return bs4.BeautifulSoup(page.text,"html.parser")


# Hàm lấy ra tất cả link từ api của tgdđ
def get_link_from_json(url):
  headers={'User-Agent':'PostmanRuntime/7.29.0'}
  page = requests.get(url,headers=headers)
  listproducts=page.json()['listproducts']
  soup=bs4.BeautifulSoup(listproducts,"html.parser")
  links = soup.findAll('a', class_='main-contain')
  return [link.get('href') for link in links]


```

Title

```
mainURL="https://www.thegioididong.com"
# url của api của tgdd
url="https://www.thegioididong.com/Category/FilterProductBox?c=42&o=9&pi="
#lấy ra tất cả link sản phẩm
allLinks=[]
for i in range(5):
  allLinks+=get_link_from_json(url+str(i))

#lấy ra thông tin của các sản phẩm và tổng hợp trong mảng tgdd
tgdd=[]
for (i,link) in enumerate(allLinks):
  try:
    print(i,mainURL+link)
    soup=getSoup(mainURL+link)
    row=[]
    # lấy ra tên sản phẩm
    row.append(soup.h1.text)
    # ------ lấy ra các thông số--------
    specs=soup.find('ul', class_='parameter__list')
    lis=specs.findAll('li')
    for li in lis[:9]:
      content=""
      for span in li.span:
        content+=span.string
      row.append(content)
    # -----------------------------------
    # lấy ra giá
    row.append(soup.find('p',class_='box-price-present').string)
    tgdd.append(row)
  except Exception as e:
    print('error: ',i, e)
    continue

# lưu dữ liệu vào file csv
df = pd.DataFrame(tgdd, columns =["Name","Screen",\
"OS","Rear camera","Front camera","Chip",\
"RAM","ROM","SIM","Batery","Price"])
df.to_csv("thegioididong.csv")
```

2.2. Mô tả dữ liệu

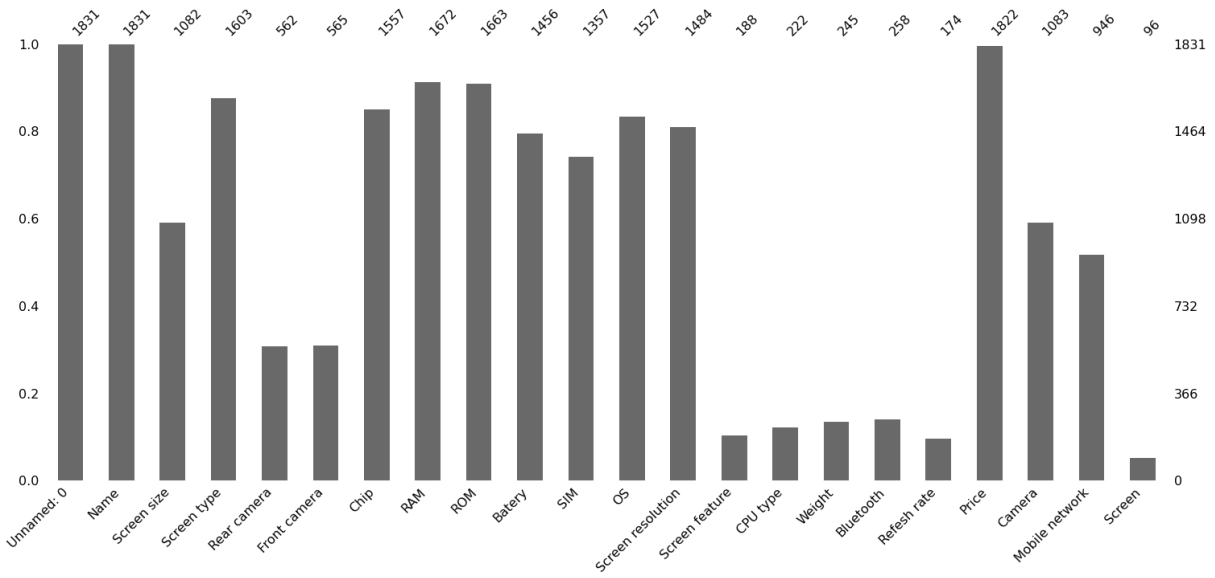
Dữ liệu được crawl từ 4 trang web khác nhau, mỗi trang web lại có một cấu trúc khác nhau nên cho ra 4 bảng dữ liệu có cấu trúc khác nhau cụ thể như sau:

- Thế Giới Di Động: 96 hàng, 11 cột
- CellphoneS: 267 hàng, 18 cột

- Hoàng Hà Mobile: 1263 hàng, 13 cột
- Nguyễn Kim: 209 hàng, 12 cột

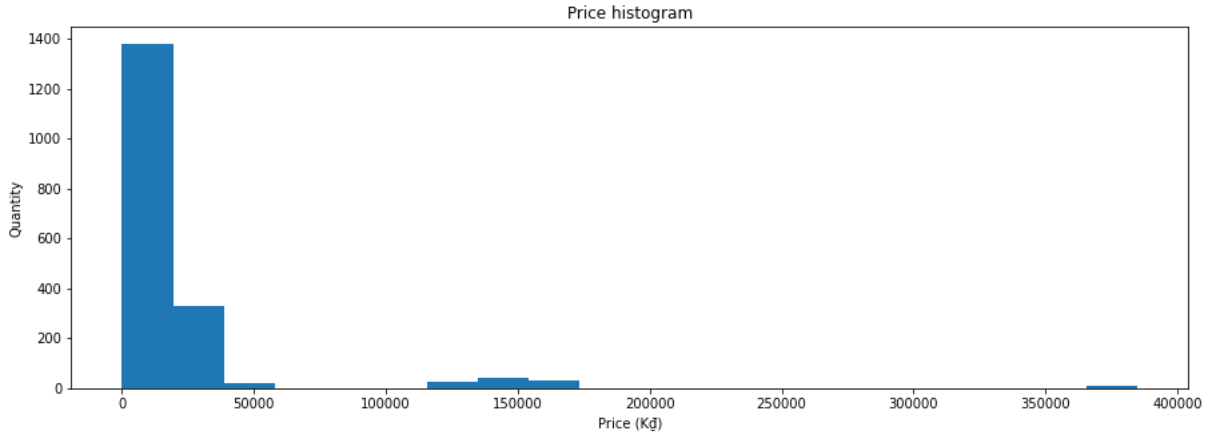
Gộp 4 bảng dữ liệu với nhau ta được bảng dữ liệu với kích thước: **1831 hàng** và **22 cột**, cụ thể như sau

Số lượng dữ liệu trống của toàn bộ dữ liệu được thể hiện như sau:



Hình 05: Số lượng dữ liệu trống của toàn bộ dữ liệu thô

Phân bố dữ liệu của trường dự đoán Price:



Hình 06: Phân bố dữ liệu của trường dự đoán Price

Hầu hết điện thoại trên 50 triệu là điện thoại Vertu có giá trị không phụ thuộc vào thông số kỹ thuật mà dựa vào thương hiệu và chất liệu là chủ yếu. Nên trong phần làm sạch dữ liệu sẽ xóa các điện thoại đó ra tập dữ liệu.

3. Trích xuất đặc trưng

3.1. Loại bỏ các hàng và các cột dữ liệu không cần thiết

Toàn bộ dữ liệu thô bao gồm tổng cộng **1831 hàng** với **22 cột**. Trong đó, có các cột không cần thiết cho mô hình dự đoán giá điện thoại, ta sẽ loại bỏ các cột này. Đó là:

- *'Unnamed: 0'*: số thứ tự của mỗi file dữ liệu thô
- *'Name'*: tên của điện thoại di động
- *'SIM'*: số lượng sim của điện thoại di động
- *'Weight'*: cân năng của điện thoại di động

Mô hình dự đoán giá điện thoại này được xây dựng để dự đoán giá của các loại điện thoại cảm ứng, nên ta sẽ loại bỏ các hàng dữ liệu điện thoại không phải cảm ứng, đó là các hàng có dữ liệu ***'Mobile network'*** là ***'2G'***. Ngoài ra, vì nó dự đoán giá dựa trên các thông số kỹ thuật, đặc trưng của điện thoại, nên với các loại điện thoại '*vertu'* có giá rất cao ảnh hưởng bởi tên thương hiệu, chất liệu hay quá trình chế tác. Nên ta sẽ loại bỏ các hàng dữ liệu điện thoại vertu này, đó là các hàng có dữ liệu ***'Name'*** là ***'XOR'***.

Screen size	645
Screen type	228
Rear camera	1117
Front camera	1114
Chip	122
RAM	55
ROM	64
Batery	271
OS	256
Screen resolution	243
Screen feature	1491
CPU type	1457
Bluetooth	1421
Refesh rate	1505
Price	9
Camera	644
Mobile network	781
Screen	1583

Tiến hành kiểm tra số lượng dữ liệu trống của các cột dữ liệu, ta thấy có 9 hàng thiếu dữ liệu giá điện thoại di động, vì đây là đặc trưng mục tiêu nên ta sẽ loại bỏ 9 hàng dữ liệu thiếu này.

Bên cạnh đó, có 7 cột dữ liệu với số lượng dữ liệu trống rất lớn, hơn 60% kích thước của bộ dữ liệu sau khi loại bỏ các hàng trên. Ta sẽ tiếp tục loại bỏ 7 cột dữ liệu này để đảm bảo kết quả dự đoán chính xác hơn.

Sau khi loại bỏ các hàng và các cột dữ liệu không cần thiết, bộ dữ liệu còn tổng cộng **1670 hàng** với **11 cột** đặc trưng.

3.2. Làm sạch dữ liệu với đúng kiểu dữ liệu

Bộ dữ liệu bao gồm 11 cột với 7 dữ liệu phân loại và 4 dữ liệu dạng số.

a) Dữ liệu phân loại:

Đối với các dữ liệu phân loại, ta đổi toàn bộ thành chữ viết thường, chọn ra các loại phổ biến nhất trong bộ dữ liệu, sau đó chuyển các chuỗi ký tự dài thành các loại tương ứng. Cụ thể:

- *'Screen type'*: Loại màn hình
  - + *OLED*
  - + *LCD*
- *'Chip'*: Loại chip
  - + *snapdragon*
  - + *apple*
  - + *mediatek*
  - + *exynos*
- *'OS':* Hệ điều hành
  - + *android*
  - + *ios*

Đặc biệt, với đặc trưng *'OS'* có các dữ liệu là hệ điều hành *'android'* nhưng ghi là *'ColorOS'*, và điện thoại *'ios'* có giá cao hơn điện thoại *'android'*. Để tăng độ chính xác dự đoán, với với các dữ liệu có loại *'Chip'* là *'apple'* thì sẽ điền dữ liệu trống cho cột '*OS'* là *'ios'*. Sau đó toàn bộ dữ liệu trống đều được thay thế cho *'android'*.

b) Dữ liệu dạng số

Đối với các dữ liệu dạng số, ta lấy số tương ứng với đơn vị của từng đặc trưng. Sau đó chuyển thành kiểu dữ liệu số thực. Cụ thể:

- *'Screen size'*: Kích thước màn hình, đơn vị *inch* hoặc ký hiệu *''*
- *'RAM' - 'ROM':* Dung lượng bộ nhớ, đơn vị *MB, GB, TB*. Đổi về cùng đơn vị *MB*
- *'Battery':* Dung lượng pin, đơn vị *mAh*

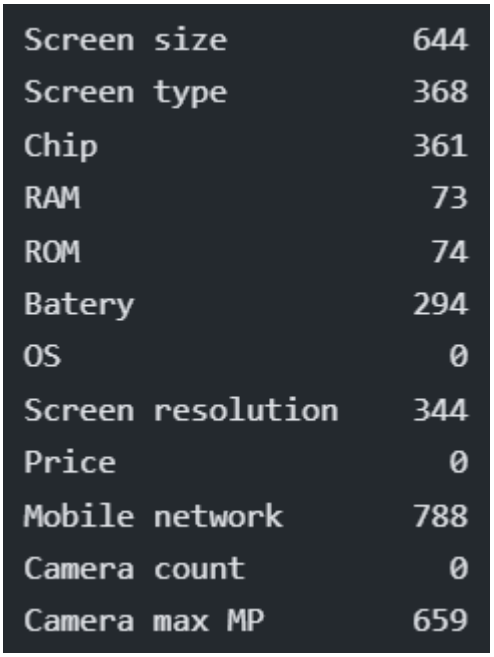
- *'Screen resolution':* Độ phân giải, đơn vị *Pixels*. Ta nhân kích thước độ phân giải lại với nhau
- *'Camera count':* Số lượng camera
- *'Camera max MP'*: Camera với MP lớn nhất
- *'Mobile network':* Hỗ trợ mạng. *4G* hoặc *5G*
- *'Price'*: Giá điện thoại di động

Dữ liệu sau khi làm sạch về đúng kiểu dữ liệu

	Screen size	Screen type	Chip	RAM	ROM	Batery	OS	Screen resolution	Price	Mobile network	Camera count	Camera max MP
0	6.70	OLED	snapdragon	8192.0	262144.0	5000.0	android	2592000.0	10190	NaN	0	NaN
1	6.10	LCD	apple	4096.0	65536.0	3110.0	ios	1483776.0	11490	NaN	0	NaN
2	6.70	OLED	apple	6144.0	131072.0	4325.0	ios	3566952.0	29790	NaN	0	NaN
3	6.43	OLED	snapdragon	4096.0	65536.0	5000.0	android	NaN	4490	NaN	0	NaN
4	6.70	OLED	apple	6144.0	131072.0	NaN	ios	3566952.0	27000	NaN	0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
1817	NaN	None	None	3072.0	32768.0	4000.0	android	NaN	2290	NaN	0	NaN
1818	NaN	None	None	2048.0	32768.0	3000.0	android	NaN	1990	NaN	0	NaN
1819	NaN	None	None	2048.0	32768.0	2950.0	android	NaN	1990	NaN	0	NaN
1826	NaN	None	None	NaN	NaN	NaN	android	NaN	650	NaN	0	NaN
1828	NaN	None	None	NaN	NaN	NaN	android	NaN	550	NaN	0	NaN

Hình 07: Dữ liệu sau khi trả về đúng kiểu dữ liệu

3.3. Xử lý dữ liệu trống



Kiểm tra số lượng dữ liệu trống của các cột dữ liệu, ta có cột *'OS', 'Price'* và *'Camera count'* không có dữ liệu trống.

Tiến hành thay thế dữ liệu trống cho các cột dữ liệu số thành giá trị mean và cho các cột dữ liệu phân loại bằng giá trị random trong các loại đã có trong bộ dữ liệu.

Đặc biệt, đối với cột dữ liệu '*Chip'*, để tăng hiệu quả dự đoán, với các dữ liệu có hệ điều hành là '*ios'* thì sẽ điền dữ liệu trống cho cột '*Chip'* là '*apple'*. Sau đó mới thay thế dữ liệu trống cho các loại *'Chip'* còn lại ngoài '*apple'*.

Dữ liệu sau khi xử lý dữ liệu trống:

	Screen size	Screen type	Chip	RAM	ROM	Batery	OS	Screen resolution	Price	Mobile network	Camera count	Camera max MP
0	6.700000	OLED	snapdragon	8192.000000	262144.000000	5000.000000	android	2.592000e+06	10190	4.0	0	37.530168
1	6.100000	LCD	apple	4096.000000	65536.000000	3110.000000	ios	1.483776e+06	11490	4.0	0	37.530168
2	6.700000	OLED	apple	6144.000000	131072.000000	4325.000000	ios	3.566952e+06	29790	4.0	0	37.530168
3	6.430000	OLED	snapdragon	4096.000000	65536.000000	5000.000000	android	2.275550e+06	4490	4.0	0	37.530168
4	6.700000	OLED	apple	6144.000000	131072.000000	4401.634448	ios	3.566952e+06	27000	4.0	0	37.530168
...	...	...	...	...	...	...	...	...	...	...	...	...
1817	6.133899	LCD	exynos	3072.000000	32768.000000	4000.000000	android	2.275550e+06	2290	4.0	0	37.530168
1818	6.133899	LCD	exynos	2048.000000	32768.000000	3000.000000	android	2.275550e+06	1990	4.0	0	37.530168
1819	6.133899	LCD	exynos	2048.000000	32768.000000	2950.000000	android	2.275550e+06	1990	4.0	0	37.530168
1826	6.133899	LCD	exynos	5882.149029	152752.761905	4401.634448	android	2.275550e+06	650	4.0	0	37.530168
1828	6.133899	LCD	exynos	5882.149029	152752.761905	4401.634448	android	2.275550e+06	550	4.0	0	37.530168

Hình 08: Dữ liệu sau khi xử lý dữ liệu trống

3.4. Chuyển các dữ liệu phân loại thành dữ liệu dạng số

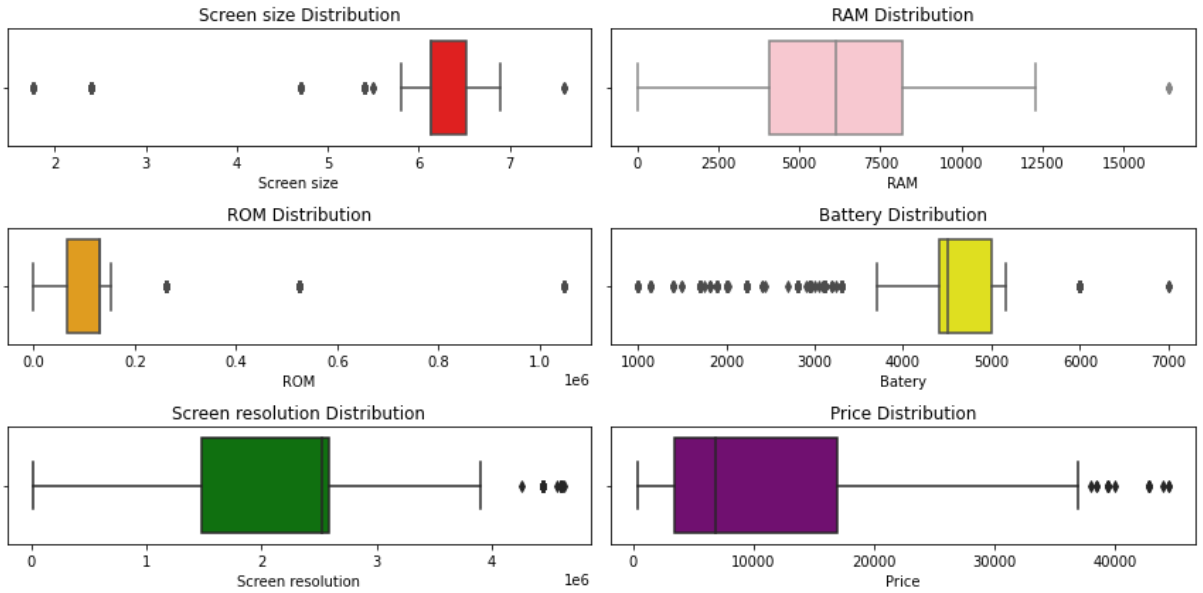
Sử dụng *LabelEncoder* của thư viện *sklearn*, chuyển các cột dữ liệu *'Screen type', 'Chip', 'OS'* thành dạng số. Dữ liệu sau khi chuyển thành dạng số:

	Screen size	Screen type	Chip	RAM	ROM	Batery	OS	Screen resolution	Mobile network	Camera count	Camera max MP	Price
0	6.700000	1	3	8192.000000	262144.000000	5000.000000	0	2.592000e+06	4.0	0	37.530168	10190
1	6.100000	0	0	4096.000000	65536.000000	3110.000000	1	1.483776e+06	4.0	0	37.530168	11490
2	6.700000	1	0	6144.000000	131072.000000	4325.000000	1	3.566952e+06	4.0	0	37.530168	29790
3	6.430000	1	3	4096.000000	65536.000000	5000.000000	0	2.275550e+06	4.0	0	37.530168	4490
4	6.700000	1	0	6144.000000	131072.000000	4401.634448	1	3.566952e+06	4.0	0	37.530168	27000
...	...	...	...	...	...	...	...	...	...	...	...	...
1665	6.133899	0	1	3072.000000	32768.000000	4000.000000	0	2.275550e+06	4.0	0	37.530168	2290
1666	6.133899	0	1	2048.000000	32768.000000	3000.000000	0	2.275550e+06	4.0	0	37.530168	1990
1667	6.133899	0	1	2048.000000	32768.000000	2950.000000	0	2.275550e+06	4.0	0	37.530168	1990
1668	6.133899	0	1	5882.149029	152752.761905	4401.634448	0	2.275550e+06	4.0	0	37.530168	650
1669	6.133899	0	1	5882.149029	152752.761905	4401.634448	0	2.275550e+06	4.0	0	37.530168	550

Hình 09: Dữ liệu sau khi chuyển dữ liệu phân loại thành dạng số

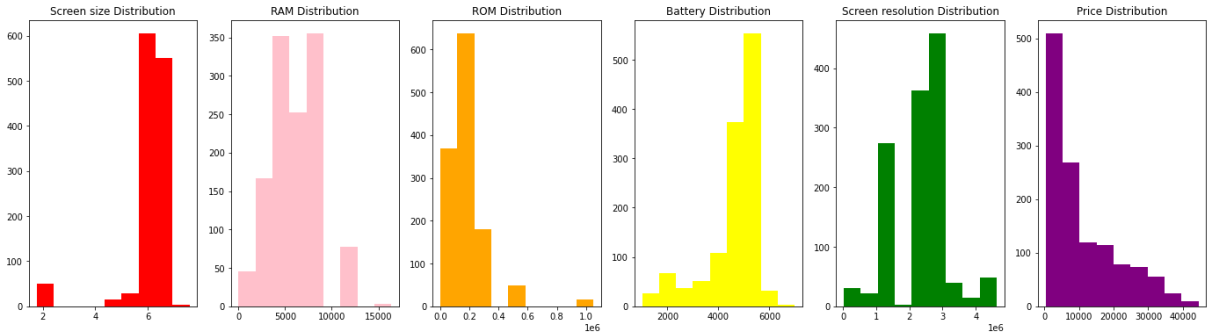
3.5. Xử lý ngoại lệ

Sau khi chia bộ dữ liệu thành 2 tập huấn luyện và kiểm thử với kích thước tập kiểm thử là 25%. Ta vẽ biểu đồ boxplot để kiểm tra ngoại lệ của 6 đặc trưng trong tập dữ liệu huấn luyện.



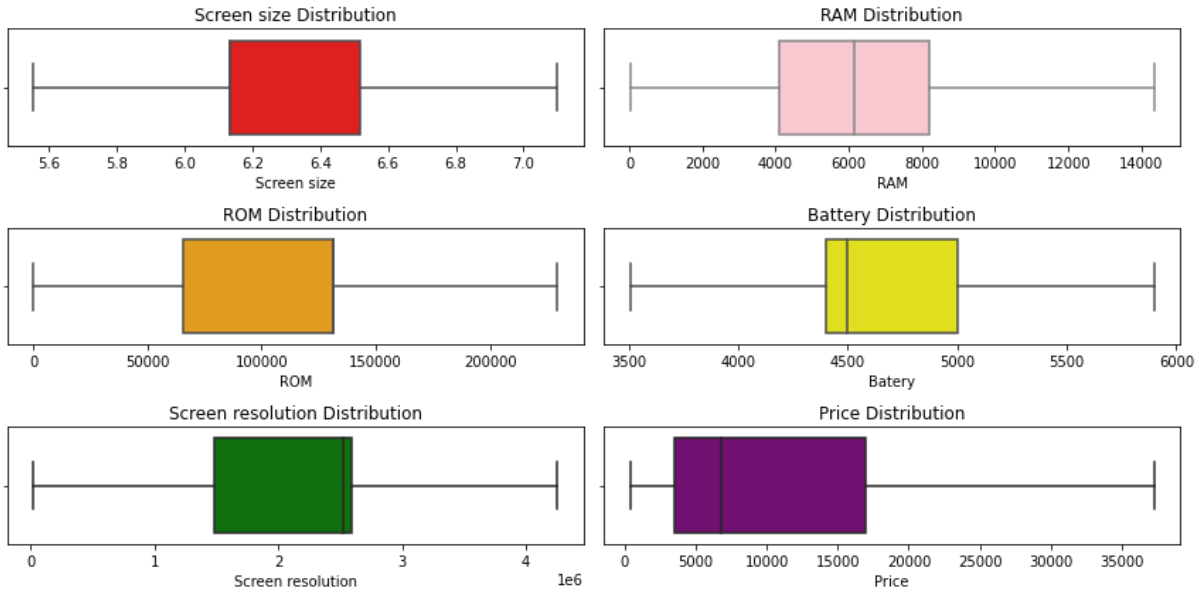
Hình 10: Phân bố dữ liệu huấn luyện khi chưa xử lý ngoại lệ

Ta thấy các tất cả đặc trưng đều có ngoại lệ. Ta vẽ biểu đồ histogram cho các đặc trưng này để kiểm tra phân bố dữ liệu



Hình 11: Dạng phân bố của dữ liệu huấn luyện

Các đặc trưng trên đều có phân bố lệch, nên ta sẽ sử dụng IQR để tìm biên cho phần xử lý ngoại lệ. Kết quả phân bố dữ liệu sau khi xử lý ngoại lệ được thể hiện qua biểu đồ sau



Hình 12: Phân bố dữ liệu huấn luyện khi đã xử lý ngoại lệ

3.6. Chuẩn hóa dữ liệu

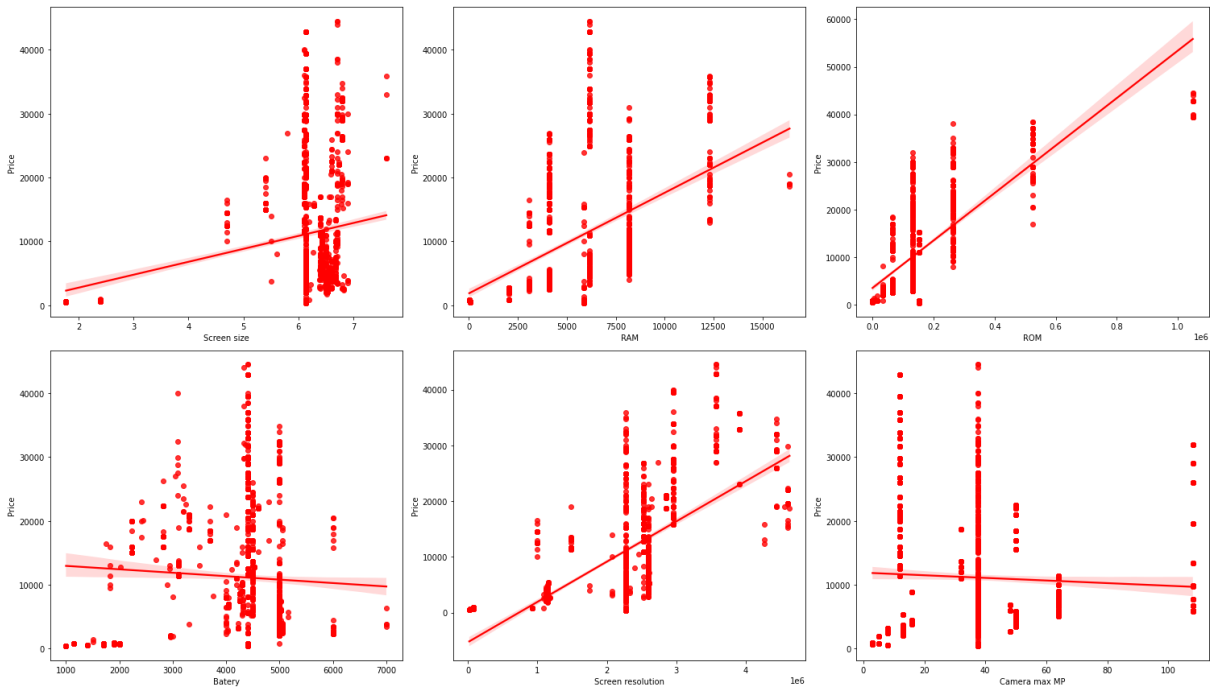
Chuẩn hóa dữ liệu sử dụng *StandardScaler* của thư viện *sklearn* với thuộc tính *fit_transform* cho tập dữ liệu huấn luyện và *transform* cho tập dữ liệu kiểm thử.

	Screen size	Screen type	Chip	RAM	ROM	Batery	OS	Screen resolution	Mobile network	Camera count	Camera max MP	Price
1545	6.100000	1	0	-0.635926	1.586480	-0.289721	1	0.821500	4.0	0	37.530168	23690.0
912	6.133899	1	3	0.102449	0.079245	-0.556342	0	0.386249	5.0	4	64.000000	7490.0
1069	6.600000	0	1	-0.635926	0.079245	0.762394	0	0.396401	4.0	4	50.000000	4490.0
1410	6.100000	1	0	0.102449	0.079245	-0.289721	1	0.821500	4.0	0	37.530168	27490.0
1329	6.560000	0	3	0.102449	0.079245	0.762394	0	-1.295536	4.0	3	13.000000	5350.0
...	...	...	...	...	...	...	...	...	...	...	...	...
715	6.133899	1	0	0.102449	1.586480	-0.289721	1	1.531775	5.0	2	12.000000	36990.0
905	6.133899	0	2	0.840825	0.079245	0.762394	0	0.396401	4.0	4	50.000000	5790.0
1096	5.554747	0	1	0.008043	-1.867600	-1.867893	0	-2.569001	4.0	1	3.000000	890.0
235	6.700000	1	1	0.840825	1.586480	0.410731	0	0.386249	4.0	0	37.530168	23000.0
1061	5.554747	1	1	-2.095371	-1.926476	-1.867893	0	0.014435	4.0	1	8.000000	590.0

Hình 13: Dữ liệu huấn luyện sau khi chuẩn hóa dữ liệu

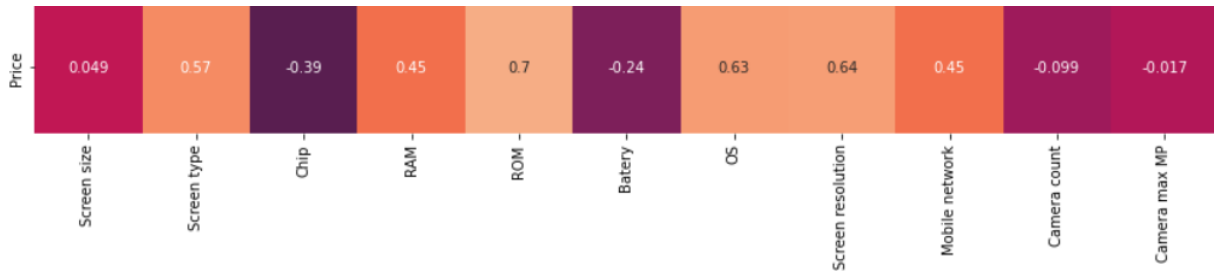
3.7. Độ tương quan giữa các đặc trưng với mục tiêu

Tiến hành trực quan hóa sự tương quan của các đặc trưng dạng số so với đặc trưng mục tiêu là *'Price'* bằng sơ đồ *regplot*. Ta thu được sơ đồ dưới đây:



Hình 14: Sự tương quan giữa các đặc trưng dạng số với đặc trưng mục tiêu

Có thể đặc trưng *'RAM', 'ROM', 'Screen resolution'* có độ tương quan rất cao. Khi giá trị của các đặc trưng này tăng thì giá điện thoại cũng tăng. Ngoài ra, đặc trưng *'Screen size'* cũng có sự tương quan nhẹ. Còn đặc trưng *'Battery'* và *'Camera max MP'* có độ tương quan không cao.



Hình 15: Sự tương quan giữa tất cả đặc trưng với đặc trưng mục tiêu

Bên cạnh là thứ tự sắp xếp độ tương quan giảm dần của các đặc trưng so với đặc trưng target. Có thể thấy *'ROM'* có độ tương quan với giá điện thoại cao nhất. Ngoài ra, các đặc trưng, *'Screen resolution', 'OS', 'Screen type', 'RAM', 'Mobile network', 'Chip'* cũng cao đáng kể. Nên ta sẽ lựa chọn các đặc trưng này làm mô hình dự đoán. Có

Price	1.000000
ROM	0.729842
Screen resolution	0.630131
OS	0.624367
Mobile network	0.479628
Screen type	0.457111
RAM	0.446790
Screen size	0.177021
Camera max MP	-0.048290
Batery	-0.052104
Camera count	-0.099076
Chip	-0.225053

thể thấy, bộ dữ liệu có các đặc trưng với mức độ tương quan khá cao so với đặc trưng mục tiêu là *'Price'* - Giá điện thoại di động.

Tổng cộng có 8 đặc trưng được lựa chọn cho mô hình dự đoán: *'ROM', 'Screen resolution', 'OS', 'Screen type', 'Mobile network', 'RAM', 'Battery', 'Chip'.*

4. Mô hình hóa dữ liệu

Để đáp ứng cho yêu cầu bài toán Dự đoán giá điện thoại, nhóm đã chọn ra 2 mô hình cho để giải quyết bài toán này: Hồi quy tuyến tính (Linear Regression) và Hồi quy của thư viện XGBoost (XGBoost Regression).

4.1. Mô hình Hồi quy tuyến tính - Linear Regression

Cơ sở lý thuyết: Hồi quy tuyến tính là một thuật toán học máy dựa trên việc học có giám sát. Nó thực hiện một nhiệm vụ hồi quy. Mô hình hồi quy một giá trị dự đoán mục tiêu dựa trên các biến độc lập. Nó chủ yếu được sử dụng để tìm ra mối quan hệ giữa các biến và dự báo. Được sử dụng trong bài toán dự đoán.

Bộ tham số của mô hình: Không sử dụng tham số nào.

Chia tập dữ liệu Huấn luyện/Kiểm thử: theo mô hình 25% kiểm thử.

Các tham số sử dụng trong quá trình huấn luyện: Không có tham số.

Đồ thị thể hiện hiệu suất của mô hình trên tập dữ liệu kiểm thử:

Hình 16: Sự chênh lệch giá cả dự đoán so với thực tế của Mô hình Linear Regression

Chú thích: Biểu đồ biểu diễn kết quả chênh lệch giữa giá dự đoán và giá thực tế của từng mẫu (lấy kết quả của 100 mẫu đầu tiên trong tập kiểm thử), đường màu tím là giá dự đoán, đường màu cam là giá thực tế.

Kết quả: Nhìn trên đồ thị ta thấy đường màu tím và màu cam còn chênh lệch khác nhiều, dao động trong khoảng 3 đến 6 triệu. Kết quả dự đoán đánh giá trên hàm score có sẵn của mô hình của mô hình có độ chính xác đạt **73.17%** trên tập dữ liệu kiểm thử.

Đánh giá: Độ hiệu quả dự đoán với các metrics, các metrics sử dụng để đánh giá bao gồm: *RMSE*, *MAE* và *R2 [3]*

	RMSE *(nghìn đồng)*	**MAE** *(nghìn đồng)*	**R2** *(%)*
Số liệu	5241.0791	3947.1287	73.17

Bảng 01: Bảng giá trị các metrics của mô hình Linear Regression

4.2. Mô hình Hồi quy của thư viện XGBoost - XGBoost Regression

Cơ sở lý thuyết: XGBoost (Extreme Gradient Boosting) là một giải thuật được base trên gradient boosting, tuy nhiên kèm theo đó là những cải tiến to lớn về mặt tối ưu thuật toán, về sự kết hợp hoàn hảo giữa sức mạnh phần mềm và phần cứng, giúp đạt được những kết quả vượt trội cả về thời gian training cũng như bộ nhớ sử dụng. Có nhiều các mô hình cho việc Hồi quy (Regression), Phân lớp (Classification),...[2]

Bộ tham số của mô hình: Sử dụng RandomizedSearchCV để tìm siêu tham số cho mô hình với *param_distributions* tự định nghĩa gồm:

Tên tham số	**Mảng giá trị**	**Ý nghĩa**
n_estimators	[100, 500, 900, 1100, 1500]	Số estimators trong mô hình tìm kiếm
max_depth	[2, 3, 5, 10, 15]	Độ sâu tìm kiếm tối đa
booster	[‘gbtree’, ‘gblinear’]	Mô hình để chạy
learning_rate	[0.05, 0.1, 0.15, 0.20]	Tốc độ học
min_child_weight	[1,2,3,4]	Trọng lượng tối thiểu của giá trị con
base_scrore	[0.25, 0.5, 0.75, 1]	Điểm khởi tạo tìm tham số

Bảng 02: Bảng tham số sử dụng để tìm siêu tham số cho mô hình XGBoost Regression

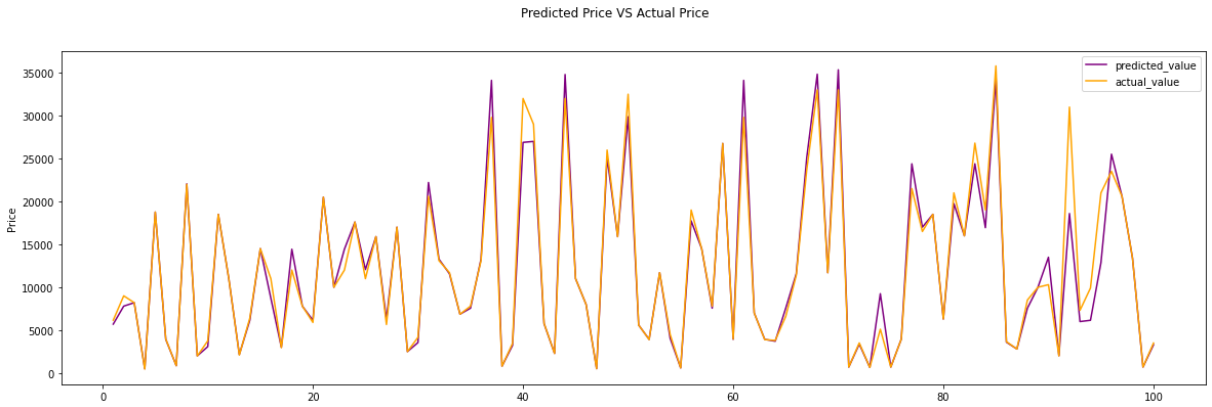
Chia tập dữ liệu Huấn luyện/Kiểm thử: theo mô hình 25% kiểm thử.

Các tham số sử dụng trong quá trình huấn luyện mô hình: Sau khi sử dụng *RandomizedSearchCV* để tìm siêu tham số cho mô hình, các tham số tìm được như sau:

Tên tham số	Giá trị	Ý nghĩa
n_estimators	900	Số estimators trong tìm kiếm
max_depth	15	Độ sâu tìm kiếm tối đa
booster	gbtree	Mô hình để chạy
learning_rate	0.1	Tốc độ học
min_child_weight	1	Trọng lượng tối thiểu của giá trị con
base_scrore	1	Mức điểm làm mốc tìm tham số

Bảng 03: Bảng siêu tham số tốt nhất vừa tìm được

Đồ thị thể hiện hiệu suất của mô hình trên tập dữ liệu kiểm thử:



Hình 17: Sự chênh lệch giá cả dự đoán so với thực tế của mô hình XGBoost Regression

Chú thích: Biểu đồ biểu diễn kết quả chênh lệch giữa giá dự đoán và giá thực tế của từng mẫu (lấy kết quả của 100 mẫu đầu tiên trong tập kiểm thử), đường màu tím là giá dự đoán, đường màu cam là giá thực tế.

Kết quả: Nhìn trên đồ thị ta thấy đường màu tím và màu cam chênh lệch rất ít, dao động trung bình trong khoảng 2 đến 3 triệu. Kết quả dự đoán đánh giá trên hàm score có sẵn của mô hình của mô hình có độ chính xác đạt **94.45%** trên tập dữ liệu kiểm thử.

Đánh giá: Độ hiệu quả dự đoán với các metrics, các metrics sử dụng để đánh giá bao gồm: *RMSE*, *MAE* và *R2 [3]*

Metrics	RMSE *(nghìn đồng)*	MAE *(nghìn đồng)*	R2 *(%)*
Số liệu	2383.5217	1057.4660	94.45

Bảng 04: Bảng giá trị các metrics của mô hình XGBoost Regression

4.3. So sánh hiệu quả của các mô hình

	RMSE *(nghìn đồng)*	MAE *(nghìn đồng)*	R2 *(%)*
LinearRegression	5241.0791	3947.1287	73.17
XGBoostRegression	2383.5217	1057.4660	94.45

Bảng 05: Bảng số liệu của các metrics giữa 2 mô hình

Chú thích:

- Số đo R2 của mô hình nào cao hơn thì mô hình đó có độ tin cậy cao hơn
- Số đo RMSE và MAE của mô hình nào thấp hơn thì mô hình đó có độ tin cậy càng cao, càng dần về 0 thì chứng minh mô hình ít bị sai số nhất, giúp xác định được độ tin cậy và hiệu quả của mô hình.

Kết quả: Dựa vào kết quả so sánh, ta thấy XGBoost Regression cho số liệu đánh giá tốt hơn, vì mô hình tìm được bộ tham số hiệu quả, bản thân thư viện XGBoost nói chung là thư viện được tối ưu hóa mạnh nên độ hiệu quả được các metric đánh giá cho kết quả cao, đáng tin cậy.

5. Kết luận

5.1. Crawl dữ liệu

Vì số lượng điện thoại trên thị trường có giới hạn nên không thể tăng kích thước của bộ dữ liệu để có được độ chính xác cao hơn. Tổng kích thước của toàn bộ dữ liệu thô là 1831 hàng và 22 cột. Dữ liệu được thu thập chứa nhiều giá trị không cần thiết, cần xử lý trả về đúng kiểu dữ liệu.

5.2. Làm sạch dữ liệu, Feature Engineering

Dữ liệu được làm sạch xử lý trả về đúng kiểu dữ liệu phù hợp với từng đặc trưng. Nhờ loại bỏ các dữ liệu gây nhiễu, các đặc trưng có số lượng dữ liệu trống quá lớn, các đặc trưng được sử dụng trong mô hình dự đoán có độ tương quan với đặc trưng mục tiêu cao, giúp cho hiệu quả mô hình dự đoán được cải thiện đáng kể. Xử lý dữ liệu trống hợp lý cho loại điện thoại iphone, với đặc trưng hệ điều hành và loại Chip được thay thế phù hợp, logic, tránh ảnh hưởng kết quả dự đoán vì giá điện thoại iphone có giá cao hơn nhiều. Tổng kích thước của toàn bộ dữ liệu sau khi xử lý và dùng để dự đoán là 1670 hàng với 12 cột. Sau khi tính toán độ tương quan của toàn bộ đặc trưng và thử nghiệm mô hình dự đoán cho từng đặc trưng, đã đưa ra được các đặc trưng cần thiết cho mô hình.

5.3. Mô hình hóa dữ liệu

Đã xây dựng được 2 mô hình dự đoán giá điện thoại là Linear Regression và XGBoost Regression với độ chính xác được đánh giá cao. Linear Regression cho R2 là 73.17%, RMSE là 5241.0791 (nghìn đồng) và MAE là 3947.1287 (nghìn đồng). Với mô hình XGBoost Regression cho số liệu đáng tin cậy hơn Linear Regression với kết quả R2 là 94.45%, RMSE là 2383.5217 (nghìn đồng) và MAE là 1057.4660 (nghìn đồng). Qua đó chọn ra mô hình phù hợp hơn cho bài toán đã đề ra là XGBoost Regression.

Tuy nhiên, không dễ để có được các thông số kỹ thuật đầy đủ để sử dụng cho mô hình dự đoán. Với mô hình Linear Regression, cần đầy đủ các thông số kỹ thuật thì mới dự đoán giá của điện thoại đó được. Nhưng với mô hình XGBoost Regression, không cần đầy đủ các thông số đầu vào, nhưng vẫn đảm bảo được kết quả dự đoán cao.

6. Tài liệu tham khảo

[1] Beautiful Soup

[2] XGBoost For Regression

[3] Comparing Robustness of MAE, MSE and RMSE