|  |  |  |
| --- | --- | --- |
|  | **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  **KHOA CÔNG NGHỆ THÔNG TIN** |  |

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

**Dự đoán giá laptop**

Giảng viên hướng dẫn: TS. Ninh Khánh Duy

|  |  |  |
| --- | --- | --- |
| HỌ VÀ TÊN SINH VIÊN | LỚP HỌC PHẦN | ĐIỂM BẢO VỆ |
|  | 19N11 |  |
|  | 19N11 |  |
|  | 19N11 |  |

ĐÀ NẴNG, 06/2022

**TÓM TẮT**

Hiện nay, laptop thiết bị phổ biến trên thị trường.Laptop hiện nay có rất nhiều mẫu mã đa dạng khiến cho người dùng khá khó khăn khi lựa chọn. Việc dự đoán được giá laptop khi đưa ra một số thông tin của chúng giúp cho người mua biết được trước phần nào giá tiền mà mình cần phải bỏ ra.

Trong bài báo cáo này, chúng em sẽ giới thiệu hai loại mô hình Linear Regression và XGBoost để dự đoán giá laptop có thể nhận được dựa trên thống kê tập dữ liệu mà chúng em đã thu thập. Độ chính xác của hai mô hình sau đó được so sánh bằng cách thông qua các metrics như RMSE, MAE và MSE để chọn ra mô hình tiềm năng.

Kết quả cho thấy rằng ta có thể dự đoán được gần đúng giá của một laptop thông qua các dữ liệu đơn giản thu thập được và đồng thời cho thấy mô hình XGBoost cho kết quả dự đoán gần với giá trị thực và cho hiệu năng tốt hơn so với mô hình Linear Regression

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | Các nhiệm vụ | Kết quả |
|  | -Thu thập dữ liêu  -Viết báo cáo  -Encoder dữ liệu | Đã hoàn thành |
|  | -Chuẩn hóa dữ liệu  -Mô hình hóa dữ liệu  -Viết báo cáo | Đã hoàn thành |
|  | -Trích xuất đặc trưng, trực quan hoá dữ liệu  -Xử lý dữ liệu trống  -Viết báo cáo | Đã hoàn thành |

**MỤC LỤC**

[1. Giới thiệu 5](#_Toc107338023)

[2. Thu thập và mô tả dữ liệu 5](#_Toc107338024)

[3. Trích xuất đặc trưng 12](#_Toc107338025)

[3.1 Làm sạch dữ liệu 12](#_Toc107338026)

[3.2 Xử lý dữ liệu trống 14](#_Toc107338027)

[4. Mô hình hóa dữ liệu 15](#_Toc107338028)

[4.1. Lựa chọn mô hình 15](#_Toc107338029)

[4.2. Chia dataset 17](#_Toc107338031)

[4.3. Các metrics đánh giá mô hình 17](#_Toc107338032)

[4.4. Đánh giá hiệu suất 2 mô hình 18](#_Toc107338033)

[5. Kết luận 20](#_Toc107338034)

[6. Tài liệu tham khảo 21](#_Toc107338035)

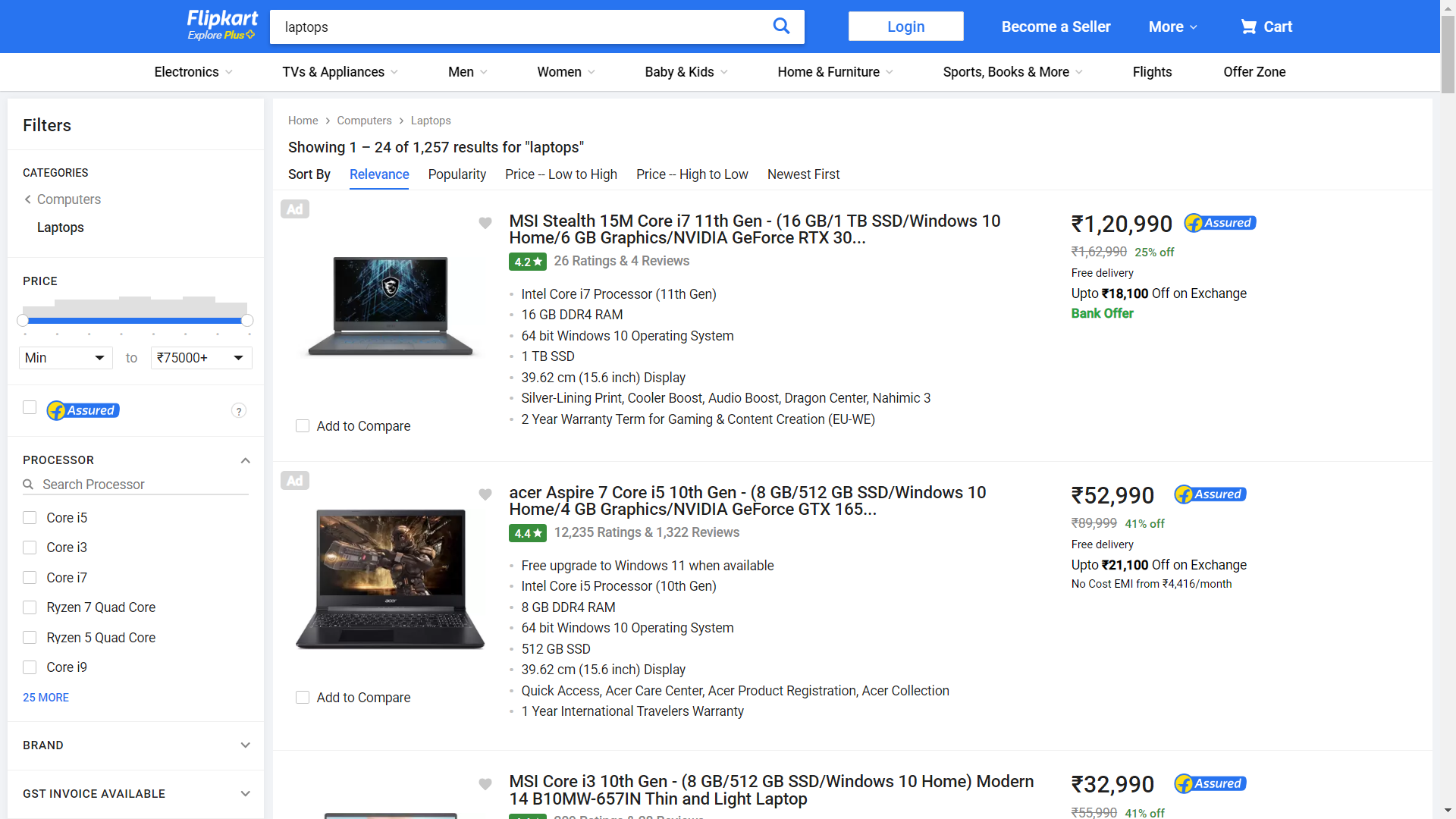
# 1. Giới thiệu

Trong quá trình học tập và tìm hiểu, chúng em nhận thấy việc chọn lựa laptop đúng giá tiền hợp lý là một vấn đề nan giải của nhiều người, để giải quyết vấn đề về chọn laptop thì nhóm xây dựng mô hình dự đoán giá laptop dựa trên các thuộc tính để có thể dễ dàng chọn laptop hơn. Sau khi nghiên cứu và thử nghiệm, hệ thống đã hoạt động tương đối. Tuy nhiên, còn một số điểm thiếu sót nên hệ thống vẫn chưa hoàn hảo. Nhóm sẽ tiếp tục phát triển và hoàn thiện trong tương lai.

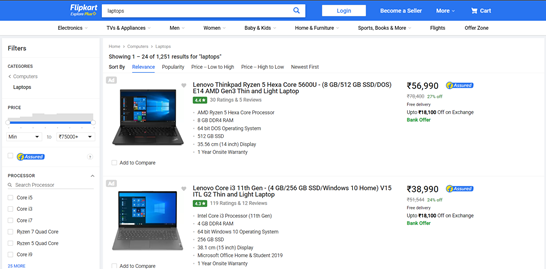
# 2. Thu thập và mô tả dữ liệu

**2.1. Thu thập dữ liệu**

*Nguồn dữ liệu*: [Web Bán Laptop Flipkart](https://www.flipkart.com/search?q=laptops&as=on&as-show=on&otracker=AS_Query_TrendingAutoSuggest_4_0_na_na_na&otracker1=AS_Query_TrendingAutoSuggest_4_0_na_na_na&as-pos=4&as-type=TRENDING&suggestionId=laptops&requestId=8384f3d9-ae2f-423d-ae09-73f116a3865e)



Hình 1: ảnh minh hoạ trang web 1



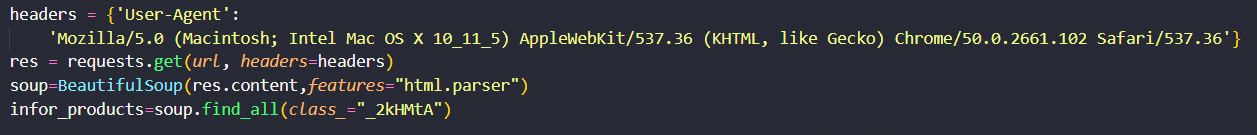
Hình 2: ảnh minh hoạ trang web 2

*Công cụ thu thập dữ liệu (Các thư viện):*

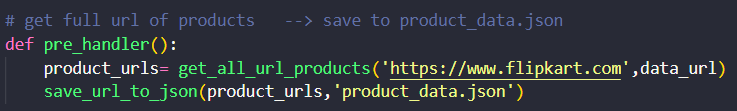
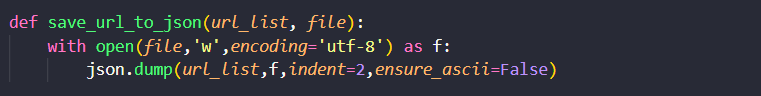
****

* BeautifulSoup: thu thập dữ liệu từ trang web (url, các sản phẩm và đặc trưng của sản phẩm, ..)
* Json : Lưu các liên kết của các trang, lưu các sản phẩm và đặt trưng của sản phẩm, .. dưới dạng file json
* Requests : lấy dữ liệu từ url đã có
* Pandas : đưa dữ liệu thu thập được vào file .csv

*Cách thức sử dụng công cụ:*



* Dùng headers để khai báo trạng thái sử dụng của web, dùng “requests” để lấy trạng thái, nội dung,.. của url
* Dùng “BeautifulSoup” để chuyển nội dung web thành html
* Lấy thông tin các thẻ của sản phẩm qua class “\_2kHMtA”

* Dùng “json” để đưa dữ liệu về json và lưu về các file .json

output = pd.DataFrame()

df\_dictionary = pd.DataFrame(list\_data)

output = pd.concat([output, df\_dictionary], *ignore\_index*=True)

output.to\_csv('product.csv',*encoding*='utf-8', *index*=False)

* Dùng “pandas” lưu các sản phẩm và đặc trưng của sản phầm vào file ‘raw data.csv’

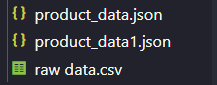
*Đầu vào và đầu ra của quá trình thu thập:*

* Đầu vào : đường link của web [Web Bán Laptop Flipkart](https://www.flipkart.com/search?q=laptops&as=on&as-show=on&otracker=AS_Query_TrendingAutoSuggest_4_0_na_na_na&otracker1=AS_Query_TrendingAutoSuggest_4_0_na_na_na&as-pos=4&as-type=TRENDING&suggestionId=laptops&requestId=8384f3d9-ae2f-423d-ae09-73f116a3865e)
* Đầu ra : các file .json và .csv
* Ví dụ:

Đầu vào:

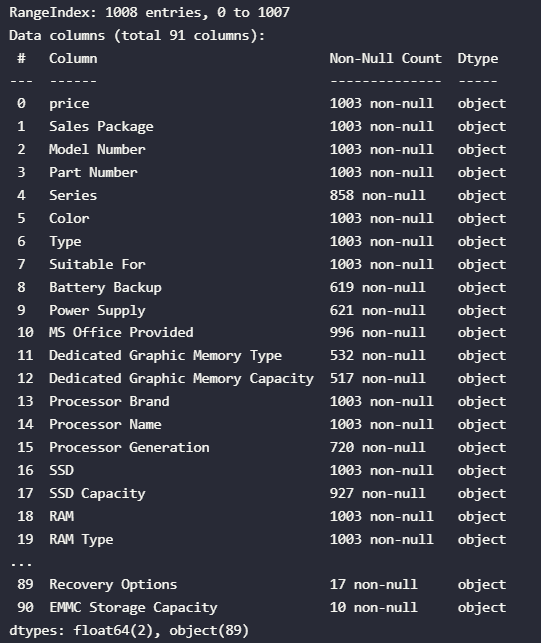
data\_url = 'https://www.flipkart.com/search?q=laptops&as=on&as-show=on&otracker=AS\_Query\_TrendingAutoSuggest\_4\_0\_na\_na\_na&otracker1=AS\_Query\_TrendingAutoSuggest\_4\_0\_na\_na\_na&as-pos=4&as-type=TRENDING&suggestionId=laptops&requestId=8384f3d9-ae2f-423d-ae09-73f116a3865e'

Đầu ra:

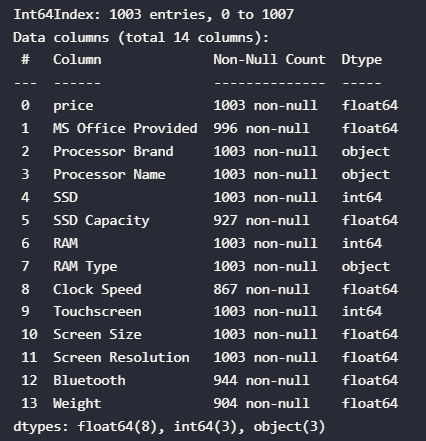


**2.2. Mô tả dữ liệu**

Dữ liệu sau khi thu thập được bao gồm 1008 mẫu với 91 đặc trưng

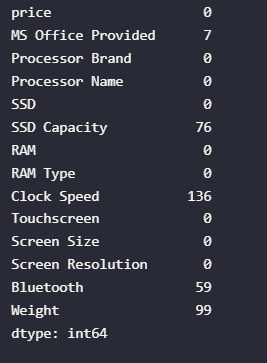


Sau khi bỏ bớt và xử lý lại dữ liệu cơ bản:

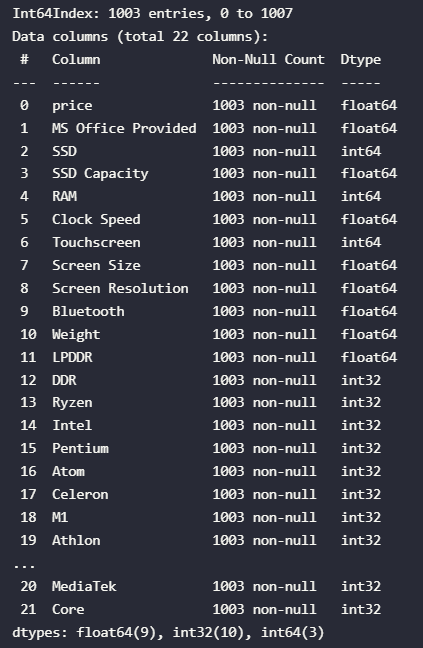


Có 11 đặc trưng là dữ liệu số học và 3 đặc trưng là dữ liệu phân loại

Số mẫu dữ liệu trống của mỗi đặc trưng :



Sau khi xử lý tách các cột phân loại:



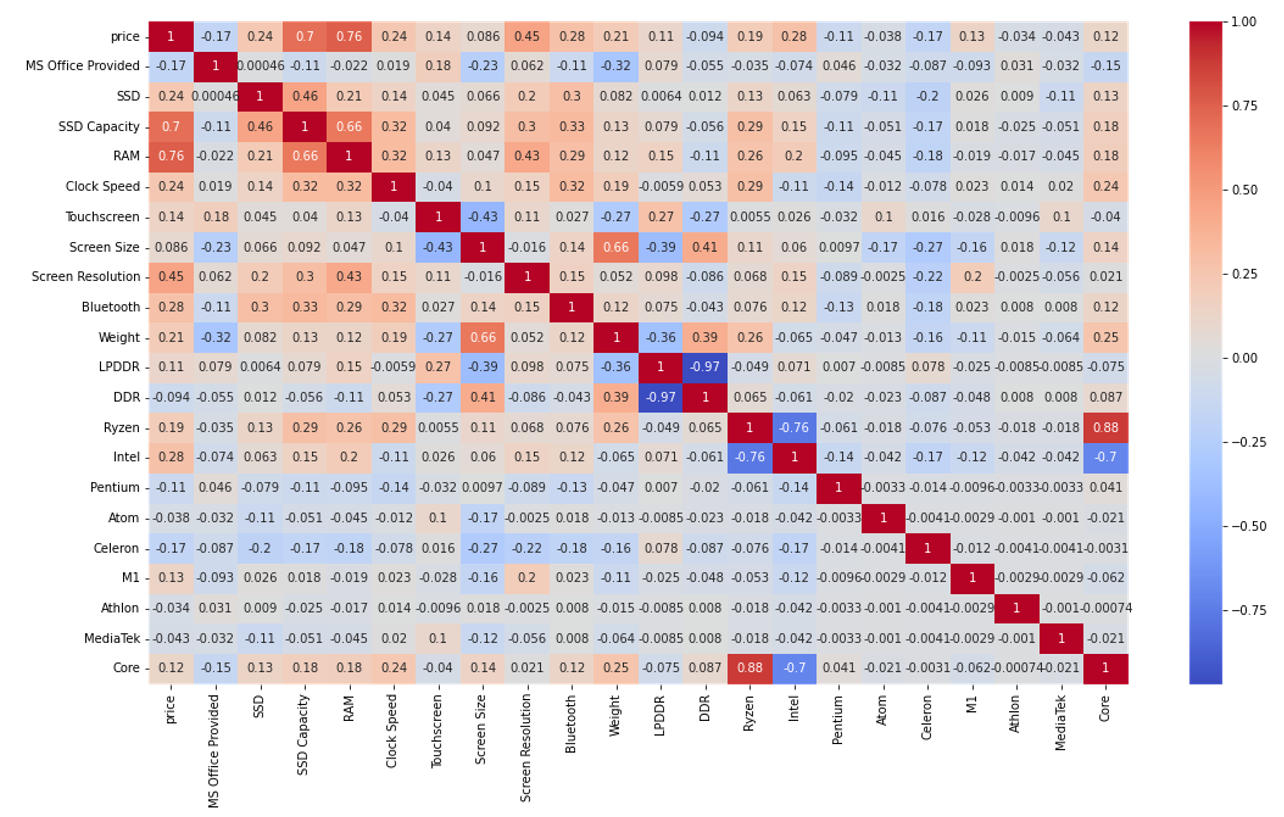
Có được 22 đặc trưng là dữ liệu số học

Mô tả các đặc trưng :

* *price*: Giá của sản phẩm.
* *MS Office Provided :* Máy có cung cấp MS Office không.
* *SSD:* Bộ nhớ SSD.
* *SSD Capacity:* Dung lượng cung cấp của SSD.
* *RAM:* Kích thước RAM của máy.
* *Clock Speed:* Tốc độ xung nhịp.
* *Touchscreen:* Màn hình cảm ứng.
* *Screen Size:* Kích thước màn hình.
* *Screen Resolution:* Độ phân giải màn hình.
* *Bluetooth:* Kết nối Bluetooth.
* *Weight:* Cân nặng của laptop.
* *LPDDR*: các loại ram LPDDR
* *DDR*: các loại ram DDR
* *Ryzen*: thế hệ Ryzen
* *Intel*: thế hệ Intel
* *Pentium*: vi Pentium
* *Atom*: vi Atom
* *Celeron*: vi Celeron
* *M1*: vi M1
* *Athlon*: vi Athlon
* *MediaTek*: vi MediaTek
* *Core*: số nhân của vi xử lý

Thống kê mô tả:

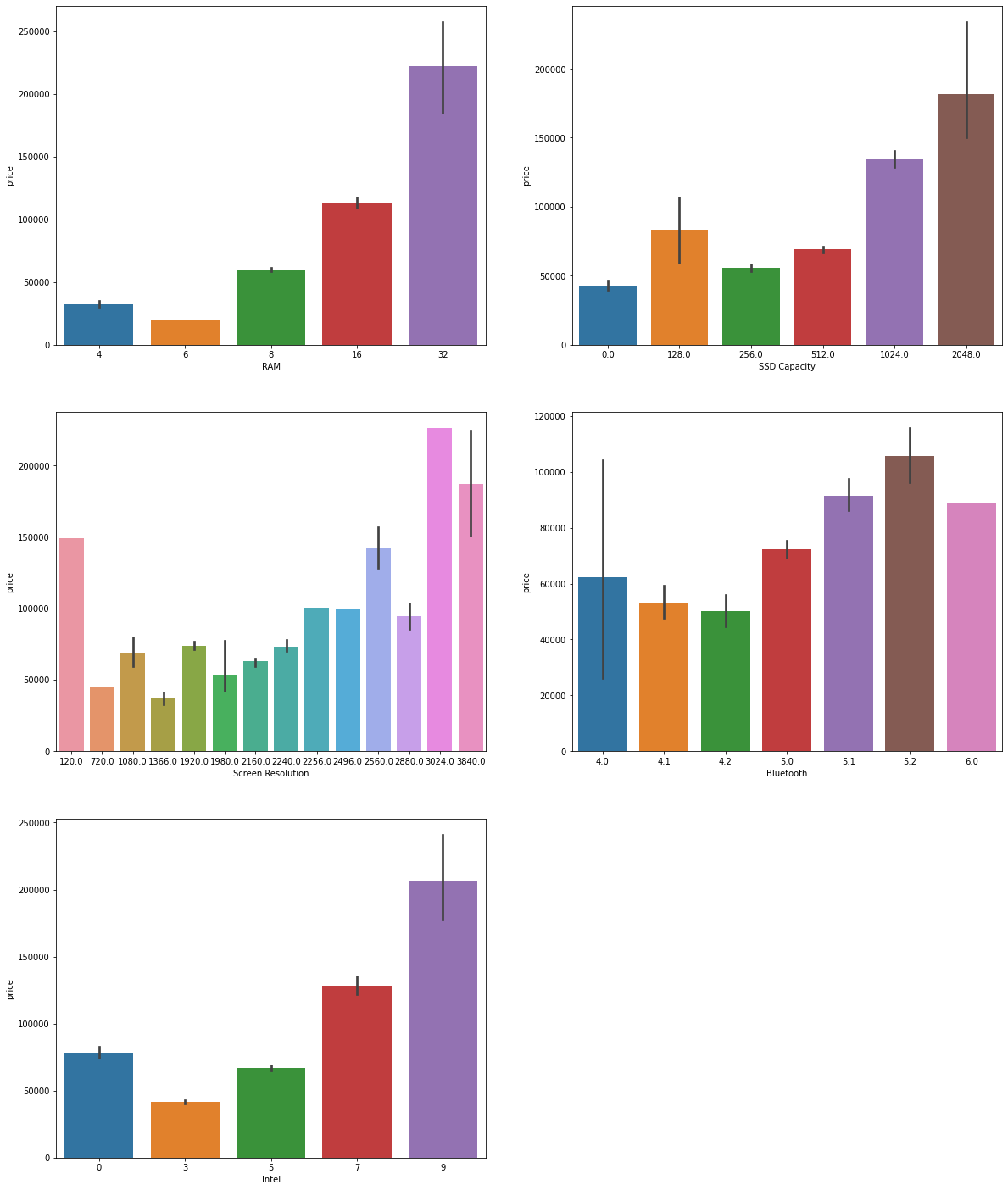
· Biểu đồ thể hiện sự tương quan giữa các biến



Hình 3: Hình vẽ minh hoạ đồ thị tương quan giữa các đặc trưng

=> Có dự tương quan cao giữa *RAM*, *SSD Capacity* với *price*

Biểu đồ cột thể hiện sự ảnh hưởng của các đặc trưng *RAM, SSD Capacity, Screen Resolution, Bluetooth, Intel* với *price*

****

Hình 4: Biểu đồ thể hiện giá của từng loại trong các đặc trưng *RAM, SSD Capacity, Screen Resolution, Bluetooth, Intel*

Nhìn vào biểu đồ có thể thấy được các đặc trưng trên có sự tương quan tích tực với giá

# 3. Trích xuất đặc trưng

## 3.1 Làm sạch dữ liệu

* Xóa đi các trường dữ liệu có dữ liệu trống lớn hơn 10% số mẫu dữ liệu

count\_null=df.isnull().sum(axis = 0)

index=[]

threshold = int(df.shape[0] \*0.1)

*for* i *in* range(len(count\_null)):

*if* count\_null[i]>threshold:

        index.append(i)

df.drop(df.columns[index],axis=1,inplace=True)

* Xóa đi các row bị trống giá

df = df[df['price'].notna()]

* Xóa đi các cột chứa thông tin không thể mã hóa

df.drop(['Type','Color','Graphic Processor','Suitable For','Sales Package','Model Number','Dimensions','Part Number','USB Port','Covered in Warranty','Not Covered in Warranty','Keyboard','Web Camera','Warranty Summary','Screen Type','Speakers','Wireless LAN','Warranty Service Type','Operating System','Disk Drive'],axis=1,inplace=True)

* Đưa dữ liệu giá về float

df['price'] = df['price'].str.replace(',', '')

df = df.astype({'price':'float'})

* Map các dữ liệu yes-no về 0-1

map={'Yes':1,'No':0}

df['MS Office Provided'] = df['MS Office Provided'].map(map)

df['SSD'] = df['SSD'].map(map)

df['Touchscreen'] = df['Touchscreen'].map(map)

* Lấy các cột có chứa thông tin dữ liệu số trong chuỗi string

df['RAM'] = pd.to\_numeric(df['RAM'].astype(str).str[:-2], errors='coerce')

df['SSD Capacity'] =df['SSD Capacity'].str.extract('(^\d\*)').astype(float)

df['SSD Capacity'] = df['SSD Capacity'].replace(1, 1024)

df['SSD Capacity'] = df['SSD Capacity'].replace(2, 2048)

df['Bluetooth'] =df['Bluetooth'].str.extract('(\d+\.\d+)').astype(float)

df['Screen Size'] =df['Screen Size'].str.extract('(\d+\.\d+)').astype(float)

df['Screen Resolution'] =df['Screen Resolution'].str.extract('(^\d\*)').astype(float)

df['Clock Speed'] =df['Clock Speed'].str.extract('(\d+\.\d+)').astype(float)

df['Weight'] = pd.to\_numeric(df['Weight'].astype(str).str[:-2], errors='coerce')

* Tách cột *RAM Type* thành các cột

Cột *LPDDR:* các loại ram LPDDR

df['LPDDR']=df['RAM Type'].str.extract('(LPDDR\d?X|LPDDR\d)',*expand*=True)

df['LPDDR']=df['LPDDR'].str.replace('(X)','.5',*regex*=True)

df['LPDDR']=df['LPDDR'].str.replace('(LPDDR)','',*regex*=True)

df['RAM Type']=df['RAM Type'].str.replace('(LPDDR)','LP',*regex*=True)

df['LPDDR'].fillna(0,*inplace*=True)

df['LPDDR'] = df['LPDDR'].astype(float)

df['LPDDR'].unique()

Cột *DDR:* các loại ram DDR

df['DDR']=df['RAM Type'].str.extract('(DDR\d)',*expand*=True)

df['DDR']=df['DDR'].str.replace('(DDR)','',*regex*=True)

df['DDR'].fillna(0,*inplace*=True)

df['DDR'] = df['DDR'].astype(int)

print(df['DDR'].unique())

* Tách cột *Processor Name* thành các cột

Cột *Ryzen*: thế hệ Ryzen

df['Ryzen']=df['Processor Name'].str.extract('(Ryzen\s\d)',*expand*=True)

df['Ryzen']=df['Ryzen'].str.replace('(Ryzen\s)','',*regex*=True)

df['Ryzen'].fillna(0,*inplace*=True)

df['Ryzen'] = df['Ryzen'].astype(int)

df['Ryzen'].unique()

Cột *Intel*: thế hệ Intel

df['Intel']=df['Processor Name'].str.extract('(Core\si\d)',*expand*=True)

df['Intel']=df['Intel'].str.replace('(Core\si)','',*regex*=True)

df['Intel'].fillna(0,*inplace*=True)

df['Intel'] = df['Intel'].astype(int)

df['Intel'].unique()

Cột *Pentium*: vi Pentium

df['Pentium']=df['Processor Name'].str.extract('(Pentium)',*expand*=True)

df['Pentium']=df['Pentium'].str.replace('(Pentium)','1',*regex*=True)

df['Pentium'].fillna(0,*inplace*=True)

df['Pentium'] = df['Pentium'].astype(int)

df['Pentium'].unique()

Cột *Atom*: vi Atom

df['Atom']=df['Processor Name'].str.extract('(Atom)',*expand*=True)

df['Atom']=df['Atom'].str.replace('(Atom)','1',*regex*=True)

df['Atom'].fillna(0,*inplace*=True)

df['Atom'] = df['Atom'].astype(int)

df['Atom'].unique()

Cột *Celeron:* vi Celeron

df['Celeron']=df['Processor Name'].str.extract('(Celeron)',*expand*=True)

df['Celeron']=df['Celeron'].str.replace('(Celeron)','1',*regex*=True)

df['Celeron'].fillna(0,*inplace*=True)

df['Celeron'] = df['Celeron'].astype(int)

df['Celeron'].unique()

Cột M1: vi M1

df['M1']=df['Processor Name'].str.extract('(M1\sPro|M1)',*expand*=True)

df['M1']=df['M1'].str.replace('(M1\sPro)','2',*regex*=True)

df['M1']=df['M1'].str.replace('(M1)','1',*regex*=True)

df['M1'].fillna(0,*inplace*=True)

df['M1'] = df['M1'].astype(int)

df['M1'].unique()

Cột *Athlon:* vi Athlon

df['Athlon']=df['Processor Name'].str.extract('(Athlon)',*expand*=True)

df['Athlon']=df['Athlon'].str.replace('(Athlon)','1',*regex*=True)

df['Athlon'].fillna(0,*inplace*=True)

df['Athlon'] = df['Athlon'].astype(int)

print(df['Athlon'].unique())

Cột *MediaTek:* vi MediaTek

df['MediaTek']=df['Processor Name'].str.extract('(MediaTek)',*expand*=True)

df['MediaTek']=df['MediaTek'].str.replace('(MediaTek)','1',*regex*=True)

df['MediaTek'].fillna(0,*inplace*=True)

df['MediaTek'] = df['MediaTek'].astype(int)

print(df['MediaTek'].unique())

Cột *Core:* số nhân của vi xử lý

df['Core']=df['Processor Name'].str.extract('(Ryzen\s\d\sHexa\sCore|Ryzen\s\d\sDual\sCore|Ryzen\s\d\sQuad\sCore|Ryzen\s\d\sOcta\sCore|Celeron\sDual\sCore|Athlon\sDual\sCore|Pentium\sQuad\sCore|Dual\sCore\si\d|Quad\sCore\si\d|Hexa\sCore\si\d|Octa\sCore\si\d)'

                                            ,*expand*=True)

df['Core']=df['Core'].str.replace('(Ryzen\s\d\sDual\sCore|Athlon\sDual\sCore|Celeron\sDual\sCore)','2',*regex*=True)

df['Core']=df['Core'].str.replace('(Quad\sCore\si\d|Pentium\sQuad\sCore|Ryzen\s\d\sQuad\sCore)','4',*regex*=True)

df['Core']=df['Core'].str.replace('(Ryzen\s\d\sHexa\sCore|Hexa\sCore\si\d)','6',*regex*=True)

df['Core']=df['Core'].str.replace('(Octa\sCore\si\d|Ryzen\s\d\sOcta\sCore)','8',*regex*=True)

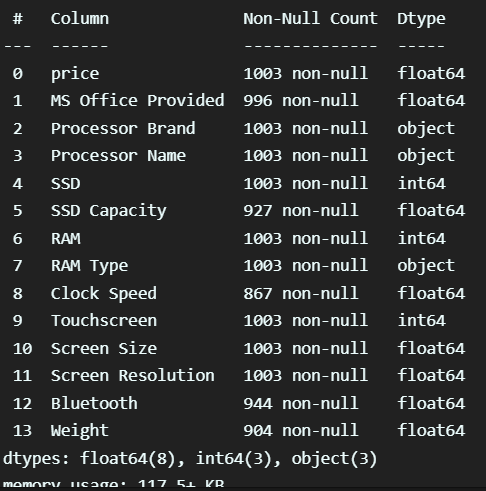
df['Core'].fillna(0,*inplace*=True)

df['Core'] = df['Core'].astype(int)

df['Core'].unique()

## 3.2 Xử lý dữ liệu trống

* Hiển thị các trường có dữ liệu trống



* Xử lý dữ liệu trống bằng phương pháp thay thế bằng các giá trị random

def Missing\_values(col):

    random\_samples = df[col].dropna().sample(n=df[col].isnull().sum(),random\_state=0)

    random\_samples

    random\_samples.index = df[df[col].isnull()].index

    df[col]=df[col]

    df.loc[df[col].isnull(), col]=random\_samples

Missing\_values('Bluetooth')

Missing\_values('Weight')

Missing\_values('Clock Speed')

Missing\_values('MS Office Provided')

* Riêng cột SSD Capacity dữ liệu trống là do Laptop không có SSD nên đưa dữ liệu trống về bằng 0

df['SSD Capacity']=df['SSD Capacity'].fillna(0)

# 4. Mô hình hóa dữ liệu

## 4.1. Lựa chọn mô hình

Hai mô hình được sử dụng là Linear Regression và XGBoost Regression là 2 mô hình phổ biến và đơn giản, phù hợp với mục tiêu bài toán.

**Mô hình Linear Regression**: Hồi quy tuyến tính là thuật toán tìm ra phương trình tuyến tính dựa trên tập dữ liệu quan hệ giữa X (dữ liệu đầu vào) và Y (dữ liệu đầu ra). Thuật toán này thích hợp để dự đoán các giá trị đầu ra là các đại lượng liên tục như doanh số hay giá cả.

**Siêu tham số của mô hình Linear Regression**: mô hình Linear Regression được sử dụng từ thư viện Sklearn là mô hình đơn giản, vì vậy không có siêu tham số.

**Mô hình XGBoost Regression**: 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.

* XGBoost có thể được sử dụng để giải quyết được tất cả các vấn đề từ hồi quy (regression), phân loại (classification), ranking và giải quyết các vấn đề do người dùng tự định nghĩa.

**Siêu tham số của mô hình XGBoost Regression:** Các thông số tổng thể đã được các tác giả của XGBoost chia thành 3 loại:

1. **General Parameters::**  Hướng dẫn hoạt động tổng thể
2. **Booster Parameters:**  Hướng dẫn tăng cường riêng lẻ (cây / hồi quy) ở mỗi bước
3. **Learning Task Parameters:**  Hướng dẫn tối ưu hóa được thực hiện

**Booster Parameters**

• learning\_rate: độ co rút kích thước bước được sử dụng để ngăn chặn việc trang bị quá mức. Phạm vi là [0,1]

• max\_depth: xác định độ sâu của mỗi cây được phép phát triển trong bất kỳ vòng thúc đẩy nào.

• subsample: tỷ lệ mẫu được sử dụng trên mỗi cây. Giá trị thấp có thể dẫn đến trang bị thấp.

• colsample\_bytree: phần trăm tính năng được sử dụng trên mỗi cây. Giá trị cao có thể dẫn đến trang bị quá nhiều.

• n\_estimators: số cây muốn dựng.

• objective: xác định hàm mất mát được sử dụng như reg:linearcho các bài toán hồi quy, reg:logisticcho các bài toán phân loại chỉ có quyết định, binary:logistic cho các bài toán phân loại với xác suất.

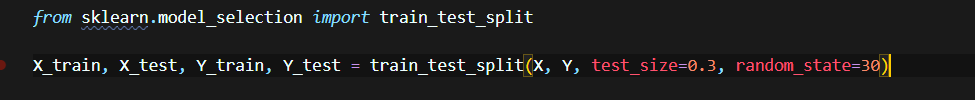
• gamma: kiểm soát xem một nút nhất định có bị tách hay không dựa trên mức giảm tổn thất dự kiến sau khi tách. Giá trị cao hơn dẫn đến ít phân tách hơn. Chỉ hỗ trợ cho người học dựa trên cây.

• alpha: Sự điều hòa L1 trên trọng lượng lá. Giá trị lớn dẫn đến chính quy hóa nhiều hơn.

• lambda: Sự điều hòa L2 trên trọng lượng lá và mượt mà hơn sự điều hòa L1

## 4.2. Chia dataset

Dataset được chia thành 2 tập train và test bằng hàm train\_test\_split với tỉ lệ tương ứng 7:3

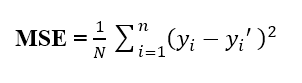


## 4.3. Các metrics đánh giá mô hình

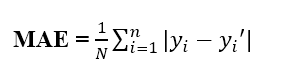
Mô hình được đánh giá bằng 2 metric: MSE(Mean Square Error), MAE(Mean Absolute Error), RMSE

**MSE (Mean Square Error)** có lẽ là một metric phổ biến nhất trong các bài toán hồi quy. Về cơ bản, nó tính trung bình của bình phương sai số giữa giá trị thực tế và giá trị dự đoán

Giả sử ta có một bài toán mà chắc hẳn ai đọc về Machine Learning cũng từng đọc qua, chính là bài toán dự đoán giá nhà. Coi giá trị thực tế của nhà thứ i là yi, còn giá trị dự đoán của căn nhà đó là yi’. Vậy, MSE có thể được tính như sau:

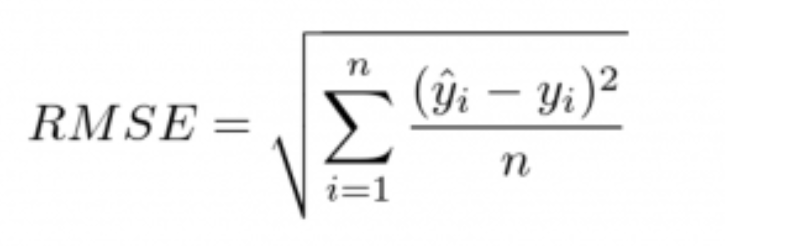


**MAE (Mean Absolute Error)** là 1 metric đánh giá mô hình bằng cách tính trung bình giá trị tuyệt đối sai số giữa giá trị thực tế và giá trị dự đoán. Công thức MAE được định nghĩa như sau:



MAE được biết đến là mạnh mẽ hơn đối với các yếu tố ngoại lai (outliers) so với MSE. Lý do chính bởi vì MSE sử dụng bình phương lỗi, các ngoại lai (những samples mà có lỗi cao hơn hẳn các samples khác) sẽ được chú ý và chiếm ưu thế hơn (do tính bình phương) trong việc đánh giá và điều này tác động đến các thông số của mô hình.

**Root Mean Squared Error (RMSE)** cũng là độ đo để đánh giá các mô hình hồi quy. RMSE được định nghĩa là căn bậc 2 trung bình tổng bình phương sai số giữa đầu ra dự đoán và kết quả thực:

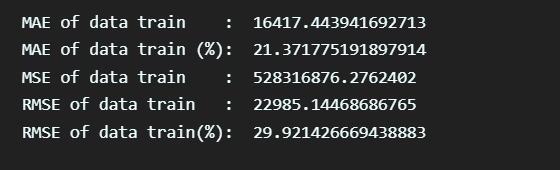


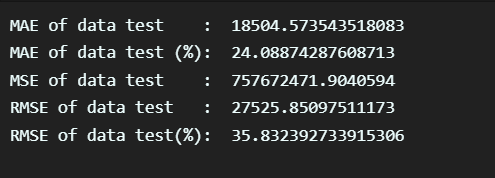
## 4.4. Đánh giá hiệu suất 2 mô hình

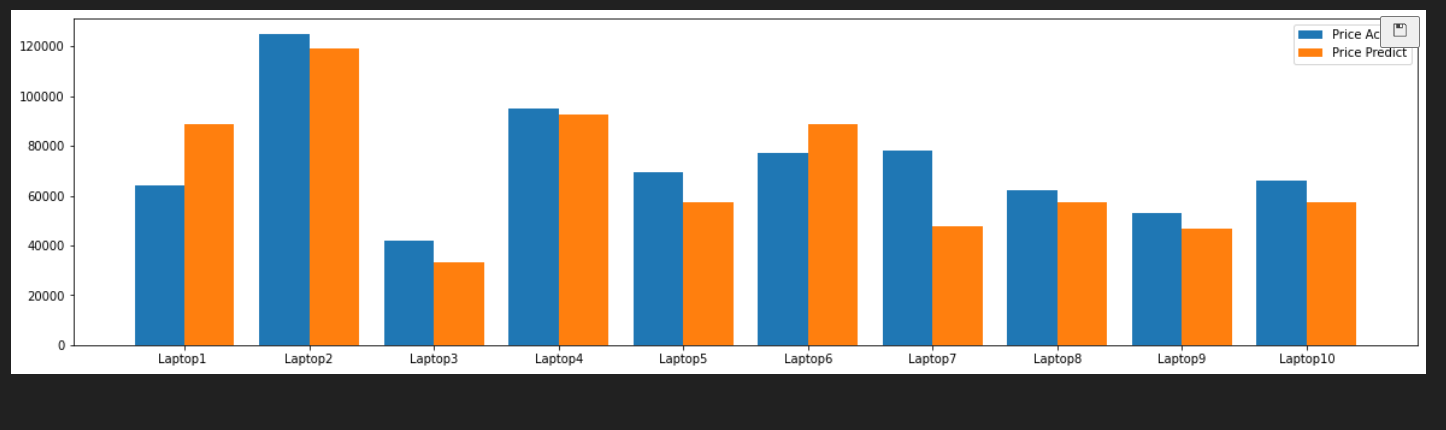
**Linear Regression**

Mô hình Linear Regression dự đoán kết quả trên tập dữ liệu Train và Test cho kết quả tương đối gần đúng, vẫn còn nhiều trường hợp kết quả dự đoán chênh lệch cao so với kết quả thực tế.

Các giá trị MSE, MAE, RMSE cho thấy độ lệch kết quả còn cao







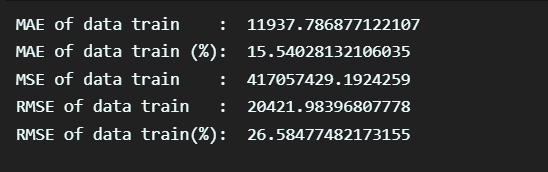
Hình 4: Hình vẽ minh hoạ giá thực tế và dự đoán của 10 mẫu khi dùng mô hình Linear Regression

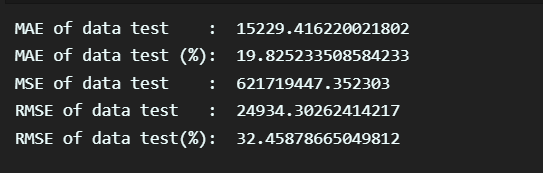
Hình vẽ cho thấy giá Laptop dự đoán từ Model và giá Laptop thực tế tương đối gần đúng

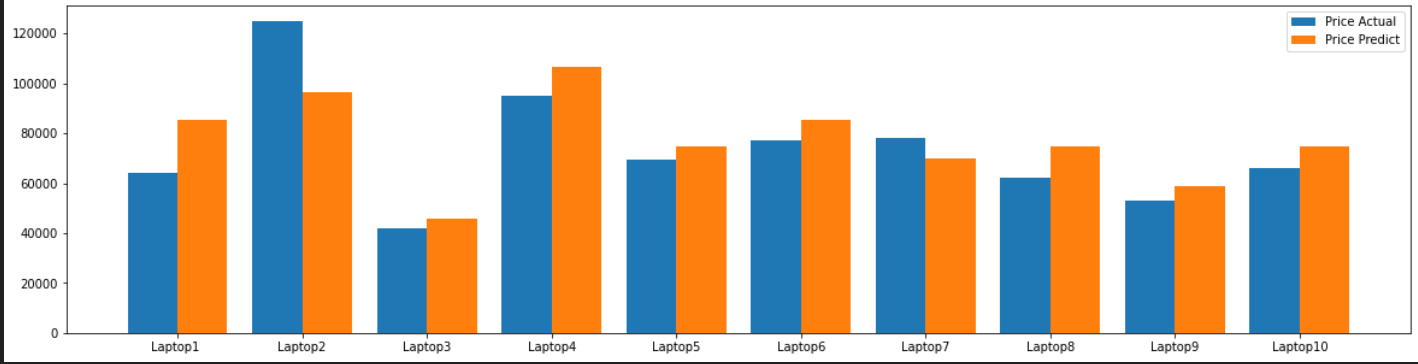
**XGBoost**

Mô hình XGBoost dự đoán kết quả trên tập dữ liệu Train và Test cho kết quả tương đối gần đúng

Các giá trị MSE, MAE, RMSE cho thấy độ lệch kết quả ít



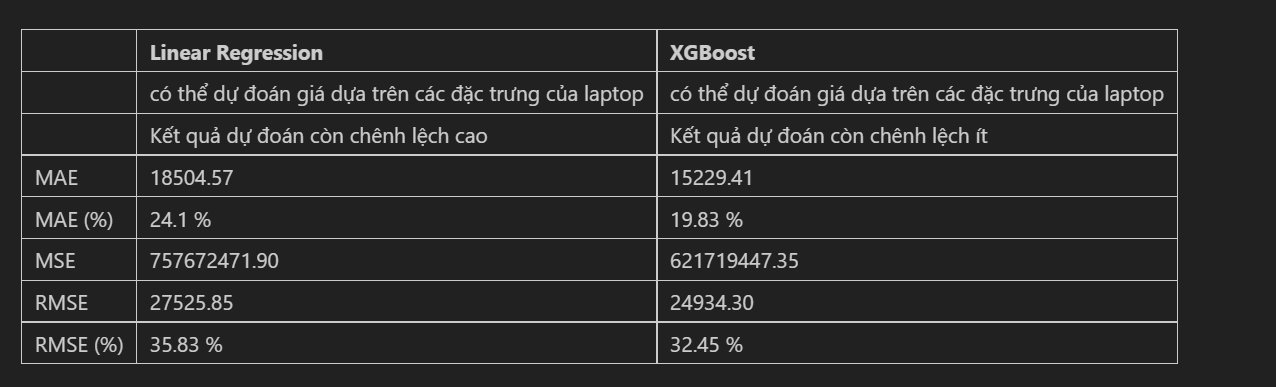




Hình 5: Hình vẽ minh hoạ giá thực tế và dự đoán của 10 mẫu khi dùng mô hình XGBoost

Hình vẽ cho thấy giá Laptop dự đoán từ Model và giá Laptop thực tế tương đối gần đúng

So sánh kết quả khi thực thi trên 2 Model



# 5. Kết luận

- Ta có thể dựa vào những thông số cửa laptop như RAM,SSD,CPU, … của laptop để dự đoán được giá của laptop với sai số khoảng 24.1% khi dùng mô hình Linear Regression, khoảng 19.83% khi dùng mô hình XGBoost.

- Mô hình XGBoost cho ra kết quả dự đoán chính xác hơn khoảng 5% so với Linear Regression .

- Hướng phát triển: tiếp tục thử các mô hình khác để tìm ra mô hình phù hợp hơn, tìm và thu thập nhiều thuộc tính của video để tạo các biến và quan hệ mới có tương quan cao với lượt xem.

# 6. Tài liệu tham khảo

[1] Gradient Boosting - Tất tần tật về thuật toán mạnh mẽ nhất trong Machine Learning

<https://viblo.asia/p/gradient-boosting-tat-tan-tat-ve-thuat-toan-manh-me-nhat-trong-machine-learning-YWOZrN7vZQ0>

[2] Using XGBoost in Python Tutorial

<https://www.datacamp.com/tutorial/xgboost-in-python>

[3] Thinh, Nguyen Toan. “Đánh Giá Model Trong Machine Learing.” Viblo, Sun\* AI Research Team, 27 June 2022, <https://viblo.asia/p/danh-gia-model-trong-machine-learing-RnB5pAq7KPG#_23-root-mean-square-error-14>.

[4] DrDang5. “Hệ SỐ Tương Quan R2 LÀ Gì.” Boxhoidap, Boxhoidap, 16 Oct. 2021, <https://boxhoidap.com/he-so-tuong-quan-r2-la-gi>.

[5] “Feature Scaling and Normalization.” Giới thiệu về Feature Engineering.

[Machine Learning cơ bản (machinelearningcoban.com)](https://machinelearningcoban.com/general/2017/02/06/featureengineering/#feature-scaling-and-normalization)