TRƯỜNG ĐẠI HỌC ĐIỆN LỰC
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN
MÔN NHẬP MÔN HỌC MÁY
ĐỀ TÀI: DỰ ĐOÁN GIÁ NHÀ BẰNG LINE REGRESSION

Sinh viên thực hiện : **NGÔ ANH ĐỨC**
NGUYỄN TRUNG QUÂN
Giảng viên hướng dẫn : **Ths.ĐÀO NAM ANH**

Ngành : **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành : **HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ**

Lớp : **D15HTTMDT2**

Khóa : **2020-2024**

Hà Nội, tháng 12 năm 2022

PHIẾU CHẤM ĐIỂM

STT	Họ và tên sinh viên	Nội dung thực hiện	Điểm	Chữ ký
1	NGÔ ANH ĐỨC			
2	NGUYỄN TRUNG QUÂN			

Họ và tên giảng viên	Chữ ký	Ghi chú
Giảng viên chấm 1:		
Giảng viên chấm 2:		

MỤC LỤC

MỤC LỤC HÌNH ẢNH

LỜI MỞ ĐẦU

Công nghệ ngày càng phổ biến và không ai có thể phủ nhận được tầm quan trọng và những hiệu quả mà nó đem lại cho cuộc sống chúng ta. Bất kỳ trong lĩnh vực nào, sự góp mặt của trí tuệ nhân tạo sẽ giúp con người làm việc và hoàn thành tốt công việc hơn. Và gần đây, một thuật ngữ "machine learning" rất được nhiều người quan tâm.Thay vì phải code phần mềm với cách thức thủ công theo một bộ hướng dẫn cụ thể nhằm hoàn thành một nhiệm vụ đề ra thì máy sẽ tự "học hỏi" bằng cách sử dụng một lượng lớn dữ liệu cùng những thuật toán cho phép nó thực hiện các tác vụ.

Đây là một lĩnh vực khoa học tuy không mới, nhưng cho thấy lĩnh vực trí tuệ nhân tạo đang ngày càng phát triển và có thể tiến xa hơn trong tương lai. Đồng thời, thời điểm này nó được xem là một lĩnh vực "nóng" và dành rất nhiều mối quan tâm để phát triển nó một cách mạnh mẽ, bùng nổ hơn.

Hiện nay, việc quan tâm machine learning càng ngày càng tăng lên là vì nhờ có machine learning giúp gia tăng dung lượng lưu trữ các loại dữ liệu sẵn, việc xử lý tính toán có chi phí thấp và hiệu quả hơn rất nhiều.

Những điều trên được hiểu là nó có thể thực hiện tự động, nhanh chóng để tạo ra những mô hình cho phép phân tích các dữ liệu có quy mô lớn hơn và phức tạp hơn đồng thời đưa ra những kết quả một cách nhanh và chính xác hơn.

Chính sự hiệu quả trong công việc và các lợi ích vượt bậc mà nó đem lại cho chúng ta khiến machine learning ngày càng được chú trọng và quan tâm nhiều hơn. Vì vậy chúng em quyết định chọn đề tài: “Dự đoán giá nhà bằng Linear-Regression” để làm báo cáo.

Chúng em xin chân thành gửi lời cảm ơn tới các thầy cô giáo trong Trường Đại học Điện Lực nói chung và các thầy cô giáo trong Khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức cũng như kinh nghiệm quý báu trong suốt quá trình học. Đặc biệt, em gửi lời cảm ơn đến thầy Đào Nam Anh đã tận tình theo sát giúp đỡ, trực tiếp chỉ bảo, hướng dẫn trong suốt quá trình nghiên cứu và học tập của chúng em.

CHƯƠNG 1: GIỚI THIỆU VỀ HỌC MÁY VÀ THUẬT TOÁN HỒI QUY TUYẾN TÍNH TRONG HỌC MÁY

1.1. Giới thiệu về học máy

1.1.1 Khái niệm về học máy

Học máy (Machine learning) là một lĩnh vực con của Trí tuệ nhân tạo (Artificial Intelligence) sử dụng các thuật toán cho phép máy tính có thể học từ dữ liệu để thực hiện các công việc thay vì được lập trình một cách rõ ràng, cung cấp cho hệ thống khả năng tự động học hỏi và cải thiện hiệu suất, độ chính xác dựa trên những kinh nghiệm từ dữ liệu đầu vào. Học máy tập trung vào việc phát triển các phần mềm, chương trình máy tính có thể truy cập vào dữ liệu và tận dụng nguồn dữ liệu đó để tự học.

Học máy vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Các mô hình học máy yêu cầu lượng dữ liệu đủ lớn để "huấn luyện" và đánh giá mô hình. Trước đây, các thuật toán học máy thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hóa các mối quan hệ giữa các dữ liệu. Sự tăng

trưởng trong dữ liệu lớn (big data) đã cung cấp các thuật toán học máy với đủ dữ liệu để cải thiện độ chính xác của mô hình và dự đoán.

1.1.2 Ứng dụng của học máy

Nhiều hoạt động hàng ngày của chúng ta được trợ giúp bởi các thuật toán machine learning, bao gồm:

Trong y tế: xác định bệnh lý của người bệnh mới dựa trên dữ liệu lịch sử của các bệnh nhân có cùng bệnh lý có cùng các đặc điểm đã được chữa khỏi trước đây, hay xác định loại thuốc phù hợp

- Trong lĩnh vực ngân hàng: xác định khả năng khách hàng chậm trả các khoản vay hoặc rủi ro tín dụng do nợ xấu dựa trên phân tích Credit score; xác định xem liệu các giao dịch có hành vi phạm tội, lừa đảo hay không.

- •Trong giáo dục: phân loại các học sinh theo hoàn cảnh, học lực để xem xem cần hỗ trợ gì cho những học sinh ví dụ như hoàn cảnh sống khó khăn nhưng học lực lại tốt.

- Trong thương mại điện tử: phân loại khách hàng theo sở thích cụ thể để hỗ trợ personalized marketing hay xây dựng hệ thống khuyến nghị, dựa trên dữ liệu từ website, social media.

1.2. Thuật toán hồi quy tuyến tính trong học máy

1.2.1. Khái niệm

Hồi quy tuyến tính la một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Hồi quy tuyến tính là một trong hai dạng lớn của học có giám sát (supervised learning) dựa trên tập dữ liệu mẫu.

Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...

1.2.2. Dạng của Linear Regression

Hồi quy tuyến tính có phương trình đạng :

$F(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$ (1)

Trong đó, w_1, w_2, w_n, w_0 là các hằng số, w_0 còn được gọi là bias hay sai số. Mối quan hệ giữa y f(x) bên trên là một mối quan hệ tuyến tính (linear). Bài

toán chúng ta đang làm là một bài toán thuộc loại regression. Bài toán đi tìm các hệ số tối ưu $\{ w_1, w_2, w_n, w_0 \}$ chính vì vậy được gọi là bài toán Linear Regression (Hồi quy tuyến tính).

Trong phương trình (1) nếu chúng ta đặt $\mathbf{w} = [w_0, w_1, w_2, w_n]^T$ là một vecter (cột) hệ số cần phải tối ưu và = $[1, x_1, x_2, x_n]$ (đọc là x bar trong tiếng Anh) là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:

y (trong đó là một vecter hàng)

1.2.3. Hàm mất mát

Máy học từ giá trị trung bình của một hàm mất mát. Đây là một phương pháp đánh giá độ hiệu quả của một thuật toán nào đó trên bộ dữ liệu cho trước. Nếu kết quả dự đoán chênh lệch quá nhiều so với kết quả thực tế, hàm mất mát sẽ là một số rất lớn. Điều tương tự xảy ra với tất cả các cặp (x_i, y_i), i = 1, 2, 3, …, N với N là số lượng dữ liệu quan sát được. Để hàm mất mát nhỏ nhất khi đó tổng sai số là nhỏ nhất tương đương với việc tìm **w** để hàm số sau đạt giá trị nhỏ nhất:

J(w) = (2)

Hàm số J(w) được gọi là hàm mất mát (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất,

điều đó đồng nghĩa với việc tìm vector hệ số **w** sao cho giá trị của hàm mất mát này càng nhỏ càng tốt.

Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát (2). Đặt là một vector cột chứa tất cả các output của training data; = là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu.

1.2.4. Tìm nghiệm của mô hình hồi quy tuyến tính

Để tìm nghiệm cho một bài toán tối ưu chúng ta thường giải phương trình đạo hàm J(w) = bằng 0.

Đạo hàm theo **w** của hàm mất mát là:

$$\frac{\partial \mathbf{J}(\mathbf{w})}{\partial \mathbf{w}} = \bar{\mathbf{X}}^T(\bar{\mathbf{X}}\mathbf{w} - \mathbf{y})$$

Phương trình đạo hàm bằng 0 tương đương với:

Đặt và b ( ) khi đó ta có :

A.W = b

(với I là ma trận đơn vị)

là nghiệm của mô hình hồi quy tuyến tính.

Trên thực tế A có thể không khả nghịch nên ta sẽ dùng ma trận giả nghịch đảo nên ta có W = hay W = Đây chính là nghiệm tổng quát của hồi quy tuyến tính.

1.2.5. Mức độ lỗi của mô hình hồi quy tuyến tính

Ta có công thức tính mức độ lỗi của mô hình như sau:

MSE =

Với mức độ lỗi của mô hình cho ta biết mức độ học của mô hình.

CHƯƠNG 2 ỨNG DỤNG THUẬT TOÁN

2.1 Giới thiệu bài toán

Lấy bối cảnh ở nước Mỹ, chúng ta sẽ đóng vai trò như một đại lý nhà nước để dự đoán giá nhà cho các khu vực. Với tập dữ liệu đã được chuẩn bị sẵn, nhiệm vụ bây giờ đó là sử dụng mô hình hồi quy tuyến tính để có thể ước tính ngôi nhà sẽ được bán với giá bao nhiêu.

Tệp dữ liệu là file mở rộng CSV. Trong tệp dữ liệu này có 7 cột và 5000 hàng:

- Avg. Area Income: Thu nhập trung bình tại khu vực ngôi nhà đã bán
- Avg. Area House Age: Trung bình tuổi của một ngôi nhà đã bán
- Avg. Area Number of Rooms: Trung bình diện tích các phòng
- Avg. Area Number of Bedrooms: Trung bình diện tích phòng ngủ
- Area Population: Dân số tại khu vực bán nhà
- Price: Giá ngôi nhà đã bán
- Address: Địa chỉ ngôi nhà bán

2.2 Giải quyết bài toán

2.2.1 Phân tích dữ liệu dataset

Chúng ta sẽ sử dụng 1 số thư viện cần thiết để xử lí và phân tích trực quan dữ liệu:

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

%matplotlib inline
```

Hình 2. 1 import thư viện

Tiếp theo chúng ta sẽ load tệp dữ liệu có đuôi .csv:

```
HouseDF = pd.read_csv('USA_Housing.csv')
```

```
HouseDF.head()
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanielltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

Hình 2.2 load dữ liệu từ file csv

Như đã nói ở trên dataset này gồm có các thông tin cơ bản đó là :

- Avg. Area Income: Thu nhập trung bình tại khu vực ngôi nhà đã bán
- Avg. Area House Age: Trung bình tuổi của một ngôi nhà đã bán
- Avg. Area Number of Rooms: Trung bình diện tích các phòng
- Avg. Area Number of Bedrooms: Trung bình diện tích phòng ngủ
- Area Population: Dân số tại khu vực bán nhà
- Price: Giá ngôi nhà đã bán
- Address: Địa chỉ ngôi nhà bán

Để có thể xem tổng quát về dataset này ta có thể sử dụng

```
HouseDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Avg. Area Income             5000 non-null   float64
 1   Avg. Area House Age          5000 non-null   float64
 2   Avg. Area Number of Rooms    5000 non-null   float64
 3   Avg. Area Number of Bedrooms 5000 non-null   float64
 4   Area Population              5000 non-null   float64
 5   Price                        5000 non-null   float64
 6   Address                      5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

Hình 2.3 tổng quát về dataset

Dựa trên dữ liệu này ta có 5000 dòng tương ứng với 5000 ngôi nhà đã được bán. Lượng dữ liệu này đủ để có thể xây dựng một mô hình học máy

Ngoài ra ta còn có thể thống kê dữ liệu này bằng describe():

```
HouseDF.describe()
```

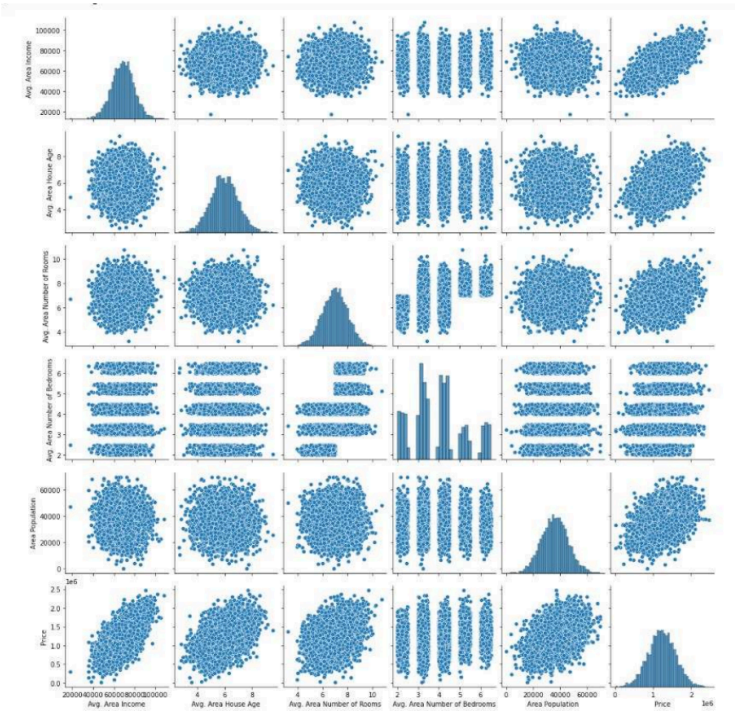
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

Hình 2.4 thống kê dữ liệu này bằng describe()

Nhờ hàm describe() giúp trả về dataframe mới với số hàng được hiển thị ra các thông số như số hàng, giá trị trung bình, độ lệch chuẩn, min, max, tỉ lệ phần tram của các cột.

Phân bố bằng Seaborn ta có thể thấy được tương quan về dữ liệu

```
sns.pairplot(HouseDF)
```



Hình 2.5 Phân bố bằng Seaborn

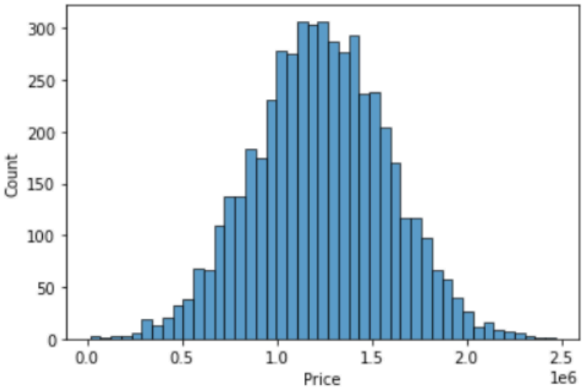
Về tương quan giữa các cột, ta thấy Cột Price có kiểu phân tán theo mô hình tuyến tính, dựa trên thông tin này, ta xây dựng mô hình máy học hồi quy

tuyến tính để dự đoán nó dựa trên giá trị các cột khác, trừ cột địa chỉ (Address) ngôi nhà.

Bây giờ chúng ta sẽ sử dụng histplot() để vẽ biểu đồ giá nhà :

```
sns.histplot(HouseDF['Price'])
```

```
<AxesSubplot:xlabel='Price', ylabel='Count'>
```



Hình 2.6 sẽ sử dụng histplot() để vẽ biểu đồ giá nhà

Ta thấy giá các ngôi nhà đã bán thường tập trung ở mức giá 0.5 đến 2.0, và nhiều nhất là 0.8 đến 1.7

Chúng ta sẽ sử dụng bản đồ nhiệt để kiểm tra độ tương quan giữa các cột:

```
sns.heatmap(HouseDF.corr(), annot=True)
```

```
<AxesSubplot:>
```

Hình 2.7 bản đồ nhiệt để kiểm tra độ tương quan giữa các cột

Qua đó, ta phân tích được các cột có giá trị tương quan như như thế nào với nhau. Về cơ bản, cột giá (price) có chút tương quan với các cột còn lại nhiều nhất, chứng tỏ các yếu tố đó có tác động ít nhiều lên giá nhà.

2.2.2 Xây dựng mô hình dự đoán bằng thư viện Scikit – Learn

2.2.2.1 Phân tách dữ liệu thành train và test

Bây giờ chúng ta hãy bắt đầu đào tạo mô hình hồi quy. Trước tiên, chúng ta sẽ cần tách dữ liệu của mình thành một mảng X chứa các tính năng cần đào

tạo (các biến độc lập) và một mảng y với biến mục tiêu (biến phụ thuộc), trong trường hợp này là cột Giá. Chúng ta sẽ loại bỏ cột Địa chỉ vì nó chỉ có thông tin văn bản mà mô hình hồi quy tuyến tính không thể sử dụng

```
X = HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
             'Avg. Area Number of Bedrooms', 'Area Population']]

y = HouseDF['Price']
```

Hình 2.8 tách dữ liệu thành một mảng X chứa các tính năng cần đào tạo và một mảng y với biến mục tiêu

Giờ ta đã có hai biến x, y theo yêu cầu của mô hình, hai biến này dựa trên dữ liệu là dataset ta có được để đào tạo mô hình. Giờ là ta tách các biến trên thành giá trị train và test, hai giá trị này chúng ta sẽ luôn gặp và sử dụng trong qua trình xây dựng mô hình máy học.

Đầu tiên, ta từ thư viện Scikti – Learn model_selection ta import train_test_split, phương thức này giúp ta tạo mô hình hồi quy

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
```

Hình 2.9 tạo mô hình hồi quy

Sau đó ta tạo 4 biến, gồm X_train, y_train và X_test, y_test. Với đối số truyền vào là giá trị X, y ta đã lấy từ dữ liệu bên trên, test_size trả về cho ta phần trăm dữ liệu được chia, ví dụ 0.4 tương ứng với dữ liệu được chia thành 40% giá trị là test, còn lại là dữ liệu train. random_state bằng một số tương ứng nào đó để đảm bảo mỗi lần ta chạy lại mô hình, giá trị phân tách ngẫu nhiên nhận được là giống nhau, bạn có thể cho số nào bất kỳ.

2.2.2.2 Tạo model và training Linear Regression

Từ thư viện Scikit – Learn , linear_model import module LinearRegression

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

Hình 2.10 Tạo model và training Linear Regression

Tiến hành train dữ liệu bằng phương thức fit()

```
lm.fit(X_train,y_train)
```

Hình 2.11 train dữ liệu bằng phương thức fit()

Nếu kết quả trả về là một hàm LinearRegression(), chứng tỏ mô hình đã train xong

2.2.3 Hệ số coeff

Để đánh giá sực tác động của các tính năng (các biến độc lập) lên kết quả đầu ra (biến phụ thuộc), ta sử dụng hệ số Coeff. Hệ số này cho ta biết khi giá trị biến độc lập thay đổi 1 đơn vị, thì giá trị đầu ra sẽ thay đổi như thế nào.

```
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
```

	Coefficient
Avg. Area Income	21.528276
Avg. Area House Age	164883.282027
Avg. Area Number of Rooms	122368.678027
Avg. Area Number of Bedrooms	2233.801864
Area Population	15.150420

Hình 2.12 hệ số Coeff

Diễn giải các hệ số trên:

Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area Income thì sẽ tăng $ 21, 52 trong giá nhà

Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area House Age thì sẽ tăng $164883.28 trong giá nhà

Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area Number of Rooms thì sẽ tăng $122368.67 trong giá nhà

Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area Number of Bedrooms thì sẽ tăng $2233.80 trong giá nhà

Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Area Population thì sẽ tăng $15.15 trong giá nhà

2.2.4 Dự đoán và đánh giá mô hình

Để dự đoán và kiểm tra mô hình, ta sử dụng dữ liệu test bên trên mà ta đã tách ra. Trong đó, X_test là các tính năng mà mô hình chưa biết, y_test là kết quả biết trước để ta so sánh với kết quả dự đoán từ X_test.

Lấy kết quả dự đoán từ X_test, ta dùng phương thức predict() truyền đối số X_test vào

```
predictions = lm.predict(X_test)
print(predictions)
```

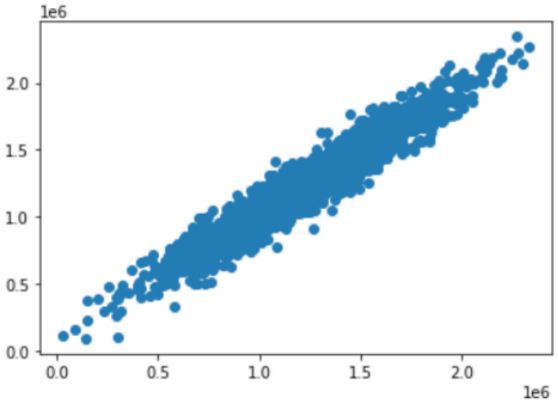
```
[1260960.70567627  827588.75560334 1742421.2425434  ...  372191.40626923
 1365217.15140897 1914519.5417887 ]
```

Hình 2.13 dùng phương thức predict() truyền đối số X_test

Kết quả dự đoán bên trên là một mảng trong numpy chứa kết quả dự đoán từ giá trị X_test, để kiểm tra kết quả dự đoán (predictions) và kết quả ban đầu (y_test) xem mô hình ta có thể trực quan quan sát bằng biểu đồ phân tán (scatter) truyền vào 2 giá trị trên vào để quan sát:

```
plt.scatter(y_test,predictions)
```

```
<matplotlib.collections.PathCollection at 0x20ca2980a90>
```

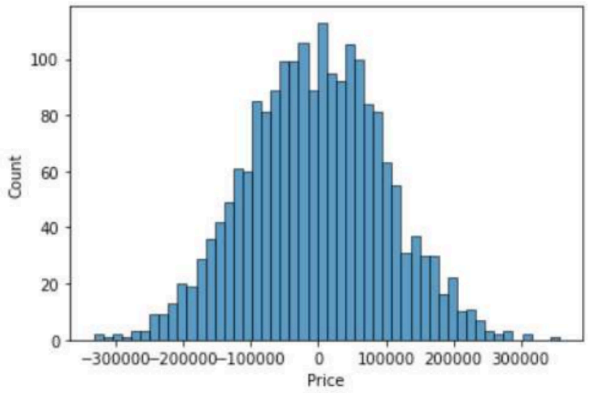


Hình 2.14 biểu đồ phân tán

Trong biểu đồ phân tán ở trên, chúng ta thấy dữ liệu có dạng đường, có nghĩa là mô hình của chúng ta đã dự đoán tốt.

Để có thể nhìn 1 cách trực quan hơn độ chênh lệch ta dùng vẽ đồ thị bằng histplot() trong Seaborn và tìm hiểu sự phân phối của độ lệch này

```
sns.histplot((y_test-predictions),bins=50);
```



Hình 2.15 sự chênh lệch giữa giá dự đoán và giá trị thực tế ban đầu

Nhìn vào biểu đồ trên, bạn thấy giá trị chênh lệch giữa giá dự đoán (predictions) và giá trị thực tế ban đầu (y_test), phân bố tập trung ở 0 và trên dưới 100.000 USD, chứng tỏ mô hình của chúng ta có độ chính xác tương đối

cao và hợp lý khi kết quả dự đoán và kết quả ban đầu có sự chênh lệch thấp và phần lớn dao động trong khoảng + (-) 10%.

Nhưng nhìn vào biểu đồ trên, nó không cho ta biết được các giá trị cụ thể mà chỉ dựa trên phán đoán trực quan tổng thể. Bây giờ, ta hãy tìm số liệu cụ thể để có cái nhìn chính xác hơn dựa trên số liệu phân tích. Ta dựa vào chỉ số đánh giá hồi quy

Chúng ta có ba chỉ số đánh giá hồi quy để có số liệu chính xác:

Mean Absolute Error (MAE): MAE là một phương pháp đo lường sự khác biệt (độ chênh lệch giá trị) giữa hai biến liên tục. Giả sử rằng X và Y là hai biến liên tục thể hiện kết quả dự đoán của mô hình và kết quả thực tế, đây là chỉ số dễ hiểu nhất, vì đó là giá trị chênh lệch trung bình và được xác định bằng công thức:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Hình 2.16 Mean Absolute Error

Mean Squared Error (MSE): là giá trị trung bình của bình phương sai số (Hàm mất mát), là sự khác biệt giữa các giá trị được mô hình dự đoán và gía

trị thực. MSE cũng được gọi là một hàm rủi ro, tương ứng với giá trị kỳ vọng của sự mất mát sai số bình phương hoặc mất mát bậc hai chỉ số này phổ biến hơn chỉ số MAE bên trên, được xác định bằng công thức:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Hình 2.17 Mean Squared Error

Root Mean Squared Error (RMSE): là căn bậc hai của giá trị trung bình của các sai số bình phương (MSE). Thông thường, ta thường dùng chỉ số này để xác định giá trị chênh lệch trung bình giữa giá dự đoán và giá trị test ban đầu, được xác định bằng công thức:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Hình 2.18 Root Mean Squared Error

Đầu tiên, chúng ta sẽ import metrics từ sklearn và lấy tất cả các chỉ số hồi quy đã nói ở trên

```
from sklearn import metrics
```

```
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 82288.22251914947
MSE: 10460958907.209059
RMSE: 102278.82922290936
```

Hình 2.19 import metrics từ sklearn và lấy tất cả các chỉ số hồi quy

Ta thấy, sử dụng chỉ số RMSE, cho thấy giá trị chênh lệch trung bình của giá dự đoán từ mô hình và giá trị thực tế là 102.278 USD.

KẾT LUẬN

Kết quả đạt được: chúng em đã cài đặt được thuật toán và sử dụng dụng thư viện scikit-learn trong quá trình học tập. Nhưng bên cạnh đó thuật toán vẫn còn những ưu nhược điểm như:

- Ưu điểm: Nhanh chóng để mô hình hóa và đặc biệt hữu ích khi mối quan hệ được mô hình hóa không quá phức tạp và nếu bạn không có nhiều dữ liệu. Hồi quy tuyến tính là đơn giản để hiểu, nó rất có giá trị cho các quyết định kinh doanh.

- Nhược điểm: Đối với dữ liệu phi tuyến tính, hồi quy đa thức có thể khá khó khăn để thiết kế, vì người ta phải có một số thông tin về cấu trúc của dữ liệu và mối quan hệ giữa các biến tính năng.

Do thời gian và kiến thức có hạn nên báo cáo chúng em vẫn còn nhiều sai sót, rất mong các thầy cô góp ý giúp chúng em hoàn thiện bài hơn nữa.