**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÀI TẬP LỚN**

**MÔN HỌC MÁY**

**ĐỀ TÀI:**

**ỨNG DỤNG PHƯƠNG PHÁP HỒI QUY TUYẾN TÍNH DỰ ĐOÁN GIÁ Ô TÔ Ở ẤN ĐỘ**

**Nhóm sinh viên thực hiện:**

1. Hoàng Thị Mai - 1851061729

2. Nguyễn Thị Hồng Hải - 1851061761

**Giảng viên hướng dẫn:**

Thầy Nguyễn Thanh Tùng

HÀ NỘI, NĂM 2021

**MỤC LỤC**

**PHẦN I: TỔNG QUAN**

[**1. Giới thiệu về học máy**](#_ta1z0t10347l) **4**

[**2. Giới thiệu về thuật toán Linear regression**](#_u63vezbn7xgo) **5**

[2.1 Khái niệm](#_lhw3a0q13049) 5

[2.2 Thuật toán](#_yzbdqp9ay5bt) 5

**PHẦN II: THỰC NGHIỆM**

[**2.1. Mô tả đề tài:**](#_gv2njlms089) **9**

[2.1.1 Tên đề tài](#_jxfl7icccgtz) 9

[2.1.2 Thuật toán đề tài](#_jv1x9vde7z32) 10

[**2.2.Mô tả tập dữ liệu của bài toán**](#_mkj68w1rngl) **11**

[**2.3. Chia tập dữ liệu**](#_14mv9g84293d) **11**

[**2.4. Mô tả ma trận dữ liệu huấn luyện (xtrain, ytrain) và ma trận dữ liệu kiểm tra (xtest, ytest)**](#_xj22ofcvilmv) **12**

[2.4.1 Tập dữ liệu huấn luyện(Training)](#_iixmb2ujj3ou) 12

[2.4.2 Tập dữ liệu kiểm tra(Testing)](#_wmtbyfb5yc9w) 12

**2.5. Dùng phương pháp Hồi quy tuyến tính để xây dựng mô hình cho bài toán với tập dữ liệu huấn luyện.** [**13**](#_o1uv6ybrcm1x)

**PHẦN III: ĐÁNH GIÁ**

[**3.1 Mô tả kết quả dự đoán**](#_injr9gxl3vw2) **13**

[**3.2 Mô tả biểu đồ đánh giá mô hình**](#_bo6ketla99ou) **16**

[**3.3 Đánh giá chất lượng mô hình**](#_3uavoi76hess) **16**

[**KẾT LUẬN**](#_je3hcsd7l80n) **17**

[**TÀI LIỆU THAM KHẢO**](#_lnxbz9) **17**

**LỜI NÓI ĐẦU**

Những năm gần đây, trí tuệ nhân tạo (artificial intelligence–AI) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1–động cơ hơi nước, 2–năng lượng điện, 3–công nghệ thông tin). Trí tuệ nhân tạo đã và đang trở thành thành phần cốt lõi trong các hệ thống công nghệ cao. Nó đã len lỏi vào hầu hết các lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook; trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, hệ thống dịch đa ngôn ngữ Google Translate, máy chơi cờ vây AlphaGo và gần đây là AlphaGo Zero của Google DeepMind, v.v., chỉ là một vài ứng dụng nổi bật trong vô vàn những ứng dụng của trí tuệ nhân tạo.

Học máy (machine learning–ML) là một tập con của trí tuệ nhân tạo. Nó là một lĩnh vực nhỏ trong khoa học máy tính, có khả năng tự học hỏi dựa trên dữ liệu được đưa vào mà không cần phải được lập trình cụ thể (Machine Learning is the subfiled of computer science, that “gives computers the ability to learn without being explicitly programmed” –Wikipedia).

Những năm gần đây, sự phát triển của các hệ thống tính toán cùng với lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn đã giúp machine learning tiến thêm một bước dài. Một lĩnh vực mới được ra đời được gọi là học sâu (deep learning–DL). Deep learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào mười năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, chuyển đổi ngôn ngữ, hay thậm chí cả sáng tác văn thơ hay âm nhạc.

**PHẦN 1: TỔNG QUAN**

# **Giới thiệu về học máy**

* 1. Khái niệm học máy

Học máy (Machine Learning) là một công nghệ từ lĩnh vực trí tuệ nhân tạo. Các thuật toán học máy là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian.

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. Các giải thuật học máy

Supervised Learning (Học có giám sát): là nhóm phổ biến nhất trong các thuật toán học máy, thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (data, label) đã biết từ trước. Hay nhiệm vụ của học có giám sát là dự đoán đầu ra mong muốn dựa vào giá trị đầu vào. Một tập hợp biến đầu vào X={x1, x2, ..., xN} và một tập hợp nhãn tương ứng Y={y1, y2,...,yN} . Các cặp dữ liệu biết trước (xi, yi) ∈X×Y được gọi là tập training data Từ tập training data, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y:

yi≈f( xi), ∀i=1,2,...,N

Unsupervised Learning (Học không giám sát) là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó. Phương pháp này thường được sử dụng để tìm cấu trúc của tập dữ liệu. Tuy nhiên lại không có phương pháp đánh giá được cấu trúc tìm ra được là đúng hay sai. Ví dụ như phân cụm dữ liệu, chiết xuất thành phần chính của một chất nào đó

# Một số giải thuật học máy khác: Học máy bán giám sát(semi--supervised learning), Học tăng cường (reinforcement learning), Các hệ thống khuyến nghị (recommender systems), etc.

# **2. Phương pháp học máy sử dụng**

## 2.1 Phương pháp sử dụng

Sử dụng phương pháp hồi quy tuyến tính (Linear Regression) để giải quyết bài toán.

## 2.2 Phương pháp hồi quy tuyến tính(Linear Regression)

## Hồi quy tuyến tính (linear regression) là một thuật toán hồi quy mà đầu ra là một hàm số tuyến tính của đầu vào. Đây là thuật toán đơn giản nhất trong nhóm các thuật toán học có giám sát.

Tổng quát, nếu mỗi điểm dữ liệu được mô tả bởi một vector đặc trưng d chiều

x ∈ R d , hàm dự đoán đầu ra được viết dưới dạng

##### **y = w1x1 + w2x2 + · · · + wdxd = x T w.**

## 2.3 Thuật toán

**Sai số dự đoán**

Sau khi xây dựng được mô hình dự đoán đầu ra , ta cần tìm một phép đánh giá phù hợp với bài toán. Với bài toán hồi quy nói chung, ta mong muốn sự sai khác e giữa đầu ra thực sự y và đầu ra dự đoán yˆ là nhỏ nhất:



Ở đây, bình phương được lấy vì e = y − yˆ có thể là một số âm. Việc sai số là nhỏ nhất có thể được mô tả bằng cách lấy trị tuyệt đối |e| = |y − yˆ|. Tuy nhiên, cách làm này ít được sử dụng vì hàm trị tuyệt đối không khả vi tại gốc toạ độ, không thuận tiện cho việc tối ưu. Hệ số 1/ 2 sẽ bị triệt tiêu khi lấy đạo hàm của e theo tham số mô hình w

**Hệ số điều chỉnh**

Điều tương tự xảy ra với tất cả các cặp dữ liệu (xi , yi), i = 1, 2, . . . , N, với N là số lượng dữ liệu trong tập huấn luyện. Việc tìm mô hình tốt nhất tương đương với việc tìm w để hàm số sau đạt giá trị nhỏ nhất:

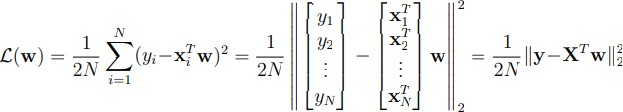


Hàm số L(w) chính là hàm mất mát của mô hình hồi quy tuyến tính với tham số θ = w. Ta luôn mong muốn sự mất mát là nhỏ nhất, điều này có thể đạt được bằng cách tối thiểu hàm mất mát theo w:

##### **w ∗ = argmin w L(w).**

w∗ \_ nghiệm cần tìm của bài toán. Đôi khi dấu ∗ được bỏ đi và nghiệm có thể được viết gọn lại thành **w = argmin w L(w).**

Trước khi xây dựng nghiệm cho bài toán tối ưu hàm mất mát, ta thấy rằng hàm số này có thể được viết gọn lại dưới dạng ma trận, vector, và norm như sau:



với y = [y1, y2, . . . , yN ] T , X = [x1, x2, . . . , xN ]. Như vậy L(w) là một hàm số liên quan tới bình phương của l2 norm.

##### **Nghiệm của hồi quy tuyến tính**

Nhận thấy rằng hàm mất mát L(w) có gradient tại mọi w. Giá trị tối ưu của w có thể tìm được thông qua việc giải phương trình đạo hàm của L(w) theo w bằng không. Gradient của hàm số này tương đối đơn giản:

Phương trình gradient bằng không:



##### **Hệ số điều chỉnh**

Hàm dự đoán đầu ra của hồi quy tuyến tính thường có thêm một hệ số điều chỉnh (bias) b:



Nếu b = 0, đường thẳng/mặt phẳng f(x)luôn đi qua gốc toạ độ. Việc thêm hệ số b khiến mô hình linh hoạt hơn. Hệ số điều chỉnh này cũng là một tham số mô hình. Để ý thấy rằng, nếu coi mỗi điểm dữ liệu có thêm một đặc trưng x0 = 1, ta có



Nếu đặt X¯ = [x¯1, x¯2, . . . , x¯N ], ta có nghiệm của bài toán tối thiểu hàm mất mát



Kỹ thuật thêm một đặc trưng x0 = 1 vào vector đặc trưng và ghép hệ số điều chỉnh b vào vector trọng số w như trên còn được gọi là thủ thuật gộp hệ số điều chỉnh (bias trick). Chúng ta sẽ gặp lại kỹ thuật đó nhiều lần trong cuốn sách này.

**PHẦN 2: THỰC NGHIỆM**

# **2.1. Mô tả đề tài:**

## 2.1.1 Tên đề tài

## Ứng dụng phương pháp hồi quy tuyến tính dự đoán giá Ô tô ở Ân Độ

+ Input: gồm

* DOORNUMBER: SỐ CHỖ
* CARLENGTH: CHIỀU DÀI(inch)
* CARWIDTH: CHIỀU RỘNG (inch)
* CARHEIGHT: CHIỀU CAO (inch)
* CYLINDERNUMBER: SỐ XILANH (int)
* ENGINESIZE: KÍCH THƯỚC ĐỘNG CƠ (cc)
* COMPRESSINRATIO: TỈ LỆ NÉN (vc)
* HORSEPOWER: MÃ LỰC (hp)

+ Output: dự đoán PRICE: GIÁ (dola) xe của mỗi Ô tô.

## 2.1.2 Thuật toán đề tài

* Nghiên cứu về phương pháp học máy có giám sát và cụ thể là thuật toán hồi quy tuyến tính với đề tài “Ứng dụng phương pháp hồi quy tuyến tính dự đoán giá Ô tô ở Ấn Độ”.
* Hàm số cơ bản nhất mô tả mối quan hệ giữa giá nhà và 3 đại lượng đầu vào là:

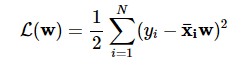
y ≈ f(x) = y’

f(x) = w1x1 + w2x2 + w3x3 + w0

* Với y là giá trị thực, y’ là giá trị dự đoán, w1,2,3 là hằng số,x1,2,3 là các giá trị của một điểm dữ liệu đầu vào, w0 là bias (kiểu hệ số tự do).
* Bài toán chúng ta trở thành đi tìm các hệ số tối ưu { w1, w2 ,w3 ,w0 }
* Với e là sự sai khác giữa giá trị thực y và giá trị dự đoán y’(sai số dự đoán), để thuận tiện cho việc tính toán, ta xét :

c

* Vớif và e 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.
* Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm w để hàm số sau đạt giá trị nhỏ nhất:

(Hàm mất mát)

* Sau các biến đổi đại số, ta tìm được w thỏa mãn yêu cầu là:

g

* Từ đó có thể xây dựng code dự đoán được giá bán xe Oto dựa trên các đặc điểm đã cho.

# **2.2.Mô tả tập dữ liệu của bài toán**

* Dữ liệu bài toán gồm:
* Ma trận x là vector đầu vào tập hợp các điểm dữ liệu, mối điểm dữ liệu mô tả giá của xe Oto(đơn vị:USD) là ma trận y (giá trị thực của đầu ra) gồm có 8 thông tin dữ liệu:
* DOORNUMBER: SỐ CHỖ
* CARLENGTH: CHIỀU DÀI(inch)
* CARWIDTH: CHIỀU RỘNG (inch)
* CARHEIGHT: CHIỀU CAO (inch)
* CYLINDERNUMBER: SỐ XILANH (int)
* ENGINESIZE: KÍCH THƯỚC ĐỘNG CƠ (cc)
* COMPRESSINRATIO: TỈ LỆ NÉN (vc)
* HORSEPOWER: MÃ LỰC (hp)
* Bài toán có 200 điểm dữ liệu.
* Dữ liệu bài toán

Dataset: <https://www.kaggle.com> .

# **2.3. Chia tập dữ liệu**

* Tập dữ liệu có 200 điểm dữ liệu chia: 140 điểm dữ liệu dùng để huấn luyện mô hình, 60 điểm dữ liệu dùng để kiểm tra sự phù hợp của mô hình.
* Dữ liệu tập huấn luyện (Training)
* Ma trận xtrain là vector đầu vào tập hợp các điểm dữ liệu, mối điểm dữ liệu mô tả giá của xe Oto là ma trận ytrain (giá trị thực của đầu ra) gồm có 8 thông tin dữ liệu: DOORNUMBER, CARLENGTH, CARWIDTH, CARHEIGHT, CYLINDERNUMBER, ENGINESIZE, COMPRESSINRATIO, HORSEPOWER.
* Tập dữ liệu huấn luyện gồm 140 điểm dữ liệu.
* Dữ liệu tập kiểm tra (Testing)
* Ma trận xtest là vector đầu vào tập hợp các điểm dữ liệu, mối điểm dữ liệu mô tả giá của xe Oto là ma trận ytest (giá trị thực của đầu ra) gồm có 8 thông tin dữ liệu: DOORNUMBER, CARLENGTH, CARWIDTH, CARHEIGHT, CYLINDERNUMBER, ENGINESIZE, COMPRESSINRATIO, HORSEPOWER.
* Tập dữ liệu huấn luyện gồm 60 điểm dữ liệu.

# **2.4. Mô tả ma trận dữ liệu huấn luyện (xtrain, ytrain) và ma trận dữ liệu kiểm tra (xtest, ytest)**

## 2.4.1 Tập dữ liệu huấn luyện(Training)

* Link: [data\_train\_oto](https://docs.google.com/spreadsheets/d/1gNEPsL5WdLOsnyPpoy_S7ZpZ2fZ69gz-Nazl1plmIbs/edit#gid=0)
* Demo 10 điểm dữ liệu trong 140 điểm dữ liệu huấn luyện

|  |  |
| --- | --- |
| Tập giá trị xtrain | Tập giá trị ytrain |
|  |  |

## 2.4.2 Tập dữ liệu kiểm tra(Testing)

* Link: [data\_test\_oto](https://docs.google.com/spreadsheets/d/1aPlLr142kz78oMuxX10GjhIRFwvxQUXcqGLS7md6DCw/edit#gid=0)
* Demo 10 điểm dữ liệu trong 60 điểm dữ liệu kiểm tra

|  |  |
| --- | --- |
| Tập giá trị xtest | Tập giá trị ytest |
|  | |  | | --- | | PRICE | | 55.7 | | 55.7 | | 55.7 | | 56.2 | | 57.5 | | 56.2 | | 57.5 | | 55.5 | | 55.5 | | 55.5 | |

# **2.5. Dùng phương pháp Hồi quy tuyến tính để xây dựng mô hình cho bài toán với tập dữ liệu huấn luyện.**

Code chương trình python

from sklearn import linear\_model  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import mean\_squared\_error  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
#lấy và xử lý data  
data = pd.read\_csv("Data\_Car.csv")  
print(data)  
X = np.array(data[['DOORNUMBER','CARLENGTH','CARWIDTH','CARHEIGHT','CYLINDERNUMBER','ENGINESIZE','COMPRESSINRATIO','HORSEPOWER']].values)  
Y = np.array(data['PRICE']).T  
#chia data theo tỷ lệ  
xtrain, xtest, ytrain, ytest = train\_test\_split(X,Y,test\_size=0.3, shuffle = False) # chia data theo tỷ lệ  
  
print("Train: \n")  
print(xtrain,'\n',ytrain)  
print("Test: \n")  
print(xtest,'\n',ytest)  
  
  
#use hồi quy  
regr1 = linear\_model.LinearRegression() #Bình phương nhỏ nhất thông thường Hồi quy tuyến tính.  
regr1.fit(xtrain,ytrain) #use hồi quy  
  
  
  
  
w\_0=regr1.intercept\_ # số hạng đập lập trong mô hình tuyến tính  
print('w\_0 = ',w\_0,'\n')  
w\_1=regr1.coef\_ # hệ số ước lượng  
print('w\_1 = ',w\_1,'\n')  
  
  
#thực hiện test  
ypre = []  
  
for i in range(len(ytest)):  
 kqDuDoan = np.dot(w\_1, xtest[i]) + w\_0 #tính kqddoan  
 ypre.append(kqDuDoan)  
 print('Với xtest là ' + str(xtest[i]) + ' thì kết quả dự đoán là '+ str(round(kqDuDoan,2)) + ' kết quả thực tế là '+ str(ytest[i]) + '\n')  
 print('Sai lệch : ',round(abs(ytest[i] - round(kqDuDoan,2)),2),'\n')  
  
  
  
  
  
# pre = regr1.predict(xtest)  
# #print('Đánh giá sai số ',round(np.sqrt(mean\_squared\_error(ytest, pre)), 3))  
  
#vẽ biểu đồ  
plt.style.use('dark\_background')  
plt.title("Đồ thị biểu diễn giá xe")  
plt.ylabel("Giá xe")  
# # plt.scatter(ytest,pre)  
# # plt.scatter(ytest,ytest,'yellow')  
plt.plot(ytest,'red')  
plt.plot(ypre,'blue')  
plt.show()

**PHẦN 3: ĐÁNH GIÁ**

# **3.1 Mô tả kết quả dự đoán**

# Demo 20 kết quả dự đoán

# Với xtest là [ 2. 171.7 65.5 55.7 4. 109. 9. 85. ] thì kết quả dự đoán là 10120.51 kết quả thực tế là 7975.0

# Sai lệch : 2145.51

# Với xtest là [ 4. 171.7 65.5 55.7 4. 97. 23. 52. ] thì kết quả dự đoán là 9457.74 kết quả thực tế là 7995.0

# Sai lệch : 1462.74

# Với xtest là [ 4. 188.8 67.2 56.2 4. 141. 9.5 114. ] thì kết quả dự đoán là 14975.07 kết quả thực tế là 15985.0

# Sai lệch : 1009.93

# Với xtest là [ 4. 188.8 67.2 57.5 4. 141. 9.5 114. ] thì kết quả dự đoán là 15252.51 kết quả thực tế là 16515.0

# Sai lệch : 1262.49

# Với xtest là [ 4. 188.8 67.2 56.2 4. 130. 7.5 162. ] thì kết quả dự đoán là 16911.94 kết quả thực tế là 18420.0

# Sai lệch : 1508.06

# Với xtest là [ 4. 188.8 67.2 57.5 4. 130. 7.5 162. ] thì kết quả dự đoán là 17189.38 kết quả thực tế là 18950.0

# Sai lệch : 1760.62

# Với xtest là [ 4. 188.8 68.9 55.5 4. 141. 9.5 114. ] thì kết quả dự đoán là 15662.79 kết quả thực tế là 16845.0

# Sai lệch : 1182.21

# Với xtest là [ 4. 188.8 68.8 55.5 4. 141. 8.7 160. ] thì kết quả dự đoán là 18659.31 kết quả thực tế là 19045.0

# Sai lệch : 385.69

# Với xtest là [ 4. 188.8 68.9 55.5 6. 173. 8.8 134. ] thì kết quả dự đoán là 20273.89 kết quả thực tế là 21485.0

# Sai lệch : 1211.11

# Với xtest là [ 4. 188.8 68.9 55.5 6. 145. 23. 106. ] thì kết quả dự đoán là 19249.39 kết quả thực tế là 22470.0

# Sai lệch : 3220.61

# Với xtest là [ 4. 188.8 68.9 55.5 4. 141. 9.5 114. ] thì kết quả dự đoán là 15662.79 kết quả thực tế là 22625.0

# Sai lệch : 6962.21

# Với xtest là [ 4. 183.1 66.9 55.1 4. 109. 9. 88. ] thì kết quả dự đoán là 9980.03 kết quả thực tế là 12290.0

# Sai lệch : 2309.97

# Với xtest là [ 4. 188.8 67.2 56.2 4. 141. 9.5 114. ] thì kết quả dự đoán là 14975.07 kết quả thực tế là 12940.0

# Sai lệch : 2035.07

# Với xtest là [ 4. 188.8 67.2 57.5 4. 141. 9.5 114. ] thì kết quả dự đoán là 15252.51 kết quả thực tế là 13415.0

# Sai lệch : 1837.51

# Với xtest là [ 2. 171.2 65.5 52.4 6. 152. 9. 154. ] thì kết quả dự đoán là 18617.62 kết quả thực tế là 16500.0

# Sai lệch : 2117.62

# Với xtest là [ 4. 176.6 66.2 54.3 4. 109. 10. 102. ] thì kết quả dự đoán là 10845.3 kết quả thực tế là 13950.0

# Sai lệch : 3104.7

# Với xtest là [ 4. 176.6 66.4 54.3 5. 136. 8. 115. ] thì kết quả dự đoán là 14028.44 kết quả thực tế là 17450.0

# Sai lệch : 3421.56

# Với xtest là [ 2. 177.3 66.3 53.1 5. 136. 8.5 110. ] thì kết quả dự đoán là 14112.76 kết quả thực tế là 15250.0

# Sai lệch : 1137.24

# Với xtest là [ 4. 192.7 71.4 55.7 5. 136. 8.5 110. ] thì kết quả dự đoán là 16169.87 kết quả thực tế là 17710.0

# Sai lệch : 1540.13

# Với xtest là [ 4. 192.7 71.4 55.7 5. 136. 8.5 110. ] thì kết quả dự đoán là 16169.87 kết quả thực tế là 18920.0

# Sai lệch : 2750.13

# Với xtest là [ 4. 192.7 71.4 55.9 5. 131. 8.3 140. ] thì kết quả dự đoán là 17838.32 kết quả thực tế là 23875.0

# Sai lệch : 6036.68

# Với xtest là [ 2. 176.2 65.6 52. 4. 146. 9.3 116. ] thì kết quả dự đoán là 14763.09 kết quả thực tế là 9989.0

# Sai lệch : 4774.09

# Với xtest là [ 4. 175.4 62.5 54.1 4. 110. 9. 86. ] thì kết quả dự đoán là 7731.9 kết quả thực tế là 10295.0

# Sai lệch : 2563.1

# Với xtest là [ 4. 175.4 65.2 54.1 4. 110. 9. 101. ] thì kết quả dự đoán là 10117.02 kết quả thực tế là 12945.0

# Sai lệch : 2827.98

# Với xtest là [ 2. 169.1 66. 51. 4. 110. 9.1 100. ] thì kết quả dự đoán là 10593.19 kết quả thực tế là 10345.0

# Sai lệch : 248.19

# Với xtest là [ 4. 170.7 61.8 53.5 4. 111. 8.5 78. ] thì kết quả dự đoán là 6777.14 kết quả thực tế là 6785.0

# Sai lệch : 7.86

# Với xtest là [ 2. 155.9 63.6 52. 4. 90. 9.6 70. ] thì kết quả dự đoán là 6200.44 kết quả thực tế là 8916.5

# Sai lệch : 2716.06

# Với xtest là [ 4. 155.9 63.6 52. 4. 90. 9.6 70. ] thì kết quả dự đoán là 5561.67 kết quả thực tế là 8916.5

# Sai lệch : 3354.83

# Với xtest là [ 2. 172.6 65.2 51.4 4. 119. 9.2 90. ] thì kết quả dự đoán là 10308.25 kết quả thực tế là 11048.0

# Sai lệch : 739.75

# Với xtest là [ 4. 199.6 69.6 52.8 6. 258. 8.1 176. ] thì kết quả dự đoán là 30009.32 kết quả thực tế là 32250.0

# Sai lệch : 2240.68

# Với xtest là [ 4. 199.6 69.6 52.8 6. 258. 8.1 176. ] thì kết quả dự đoán là 30009.32 kết quả thực tế là 31500.0

# Sai lệch : 1490.68

# Với xtest là [ 4. 177.8 66.5 55.5 4. 122. 8.6 84. ] thì kết quả dự đoán là 10756.39 kết quả thực tế là 8495.0

# Sai lệch : 2261.39

# Với xtest là [ 2. 177.8 66.5 53.7 4. 122. 8.6 84. ] thì kết quả dự đoán là 11011.01 kết quả thực tế là 10595.0

# Sai lệch : 416.01

# Với xtest là [ 4. 177.8 66.5 55.5 4. 122. 8.6 84. ] thì kết quả dự đoán là 10756.39 kết quả thực tế là 10245.0

# Sai lệch : 511.39

# Với xtest là [ 4. 177.8 66.5 55.5 4. 122. 22.7 64. ] thì kết quả dự đoán là 12721.28 kết quả thực tế là 10795.0

# Sai lệch : 1926.28

# Với xtest là [ 4. 177.8 66.5 55.5 4. 122. 8.6 84. ] thì kết quả dự đoán là 10756.39 kết quả thực tế là 11245.0

# Sai lệch : 488.61

# Với xtest là [ 4. 175. 66.1 54.4 4. 140. 8. 120. ] thì kết quả dự đoán là 14357.09 kết quả thực tế là 18280.0

# Sai lệch : 3922.91

# Với xtest là [ 4. 175. 66.1 54.4 4. 134. 22. 72. ] thì kết quả dự đoán là 13802.58 kết quả thực tế là 18344.0

# Sai lệch : 4541.42

# Với xtest là [ 4. 190.9 70.3 56.5 5. 183. 21.5 123. ] thì kết quả dự đoán là 23979.36 kết quả thực tế là 25552.0

# Sai lệch : 1572.64

# Với xtest là [ 4. 190.9 70.3 58.7 5. 183. 21.5 123. ] thì kết quả dự đoán là 24448.88 kết quả thực tế là 24252.0

# Sai lệch : 196.88

# Với xtest là [ 2. 187.5 70.3 54.9 5. 183. 21.5 123. ] thì kết quả dự đoán là 24358.42 kết quả thực tế là 24176.0

# Sai lệch : 182.42

# Với xtest là [ 2. 165.3 63.8 54.5 4. 103. 21.9 55. ] thì kết quả dự đoán là 9630.3 kết quả thực tế là 7099.0

# Sai lệch : 2531.3

# Với xtest là [ 2. 165.3 63.8 54.5 4. 97. 9.4 69. ] thì kết quả dự đoán là 7100.78 kết quả thực tế là 6649.0

# Sai lệch : 451.78

# Với xtest là [ 4. 165.3 63.8 54.5 4. 97. 9.4 69. ] thì kết quả dự đoán là 6462.01 kết quả thực tế là 6849.0

# Sai lệch : 386.99

# Với xtest là [ 4. 170.2 63.8 53.5 4. 97. 9.4 69. ] thì kết quả dự đoán là 6130.75 kết quả thực tế là 7349.0

# Sai lệch : 1218.25

# Với xtest là [ 2. 165.3 63.8 54.5 4. 97. 9.4 69. ] thì kết quả dự đoán là 7100.78 kết quả thực tế là 7299.0

# Sai lệch : 198.22

# Với xtest là [ 2. 165.6 63.8 53.3 4. 97. 9.4 69. ] thì kết quả dự đoán là 6837.47 kết quả thực tế là 7799.0

# Sai lệch : 961.53

# Với xtest là [ 4. 165.3 63.8 54.5 4. 97. 9.4 69. ] thì kết quả dự đoán là 6462.01 kết quả thực tế là 7499.0

# Sai lệch : 1036.99

# Với xtest là [ 4. 170.2 63.8 53.5 4. 97. 9.4 69. ] thì kết quả dự đoán là 6130.75 kết quả thực tế là 7999.0

# Sai lệch : 1868.25

# Với xtest là [ 2. 162.4 63.8 53.3 4. 97. 9.4 69. ] thì kết quả dự đoán là 6914.43 kết quả thực tế là 8249.0

# Sai lệch : 1334.57

# Với xtest là [ 4. 173.4 65.2 54.7 4. 120. 8.5 97. ] thì kết quả dự đoán là 10767.22 kết quả thực tế là 8949.0

# Sai lệch : 1818.22

# Với xtest là [ 4. 198.9 68.4 58.7 4. 152. 21. 95. ] thì kết quả dự đoán là 18232.66 kết quả thực tế là 13860.0

# Sai lệch : 4372.66

# Với xtest là [ 4. 186.7 68.4 56.7 4. 120. 8.4 95. ] thì kết quả dự đoán là 12285.29 kết quả thực tế là 15580.0

# Sai lệch : 3294.71

# Với xtest là [ 4. 186.7 68.4 56.7 4. 152. 21. 95. ] thì kết quả dự đoán là 18099.23 kết quả thực tế là 16900.0

# Sai lệch : 1199.23

# Với xtest là [ 4. 198.9 68.4 56.7 4. 120. 8.4 95. ] thì kết quả dự đoán là 11991.88 kết quả thực tế là 16695.0

# Sai lệch : 4703.12

# Với xtest là [ 4. 198.9 68.4 58.7 4. 152. 21. 95. ] thì kết quả dự đoán là 18232.66 kết quả thực tế là 17075.0

# Sai lệch : 1157.66

# Với xtest là [ 4. 186.7 68.4 56.7 4. 120. 8.4 97. ] thì kết quả dự đoán là 12426.03 kết quả thực tế là 16630.0

# Sai lệch : 4203.97

# Với xtest là [ 4. 186.7 68.4 56.7 4. 152. 21. 95. ] thì kết quả dự đoán là 18099.23 kết quả thực tế là 17950.0

# Sai lệch : 149.23

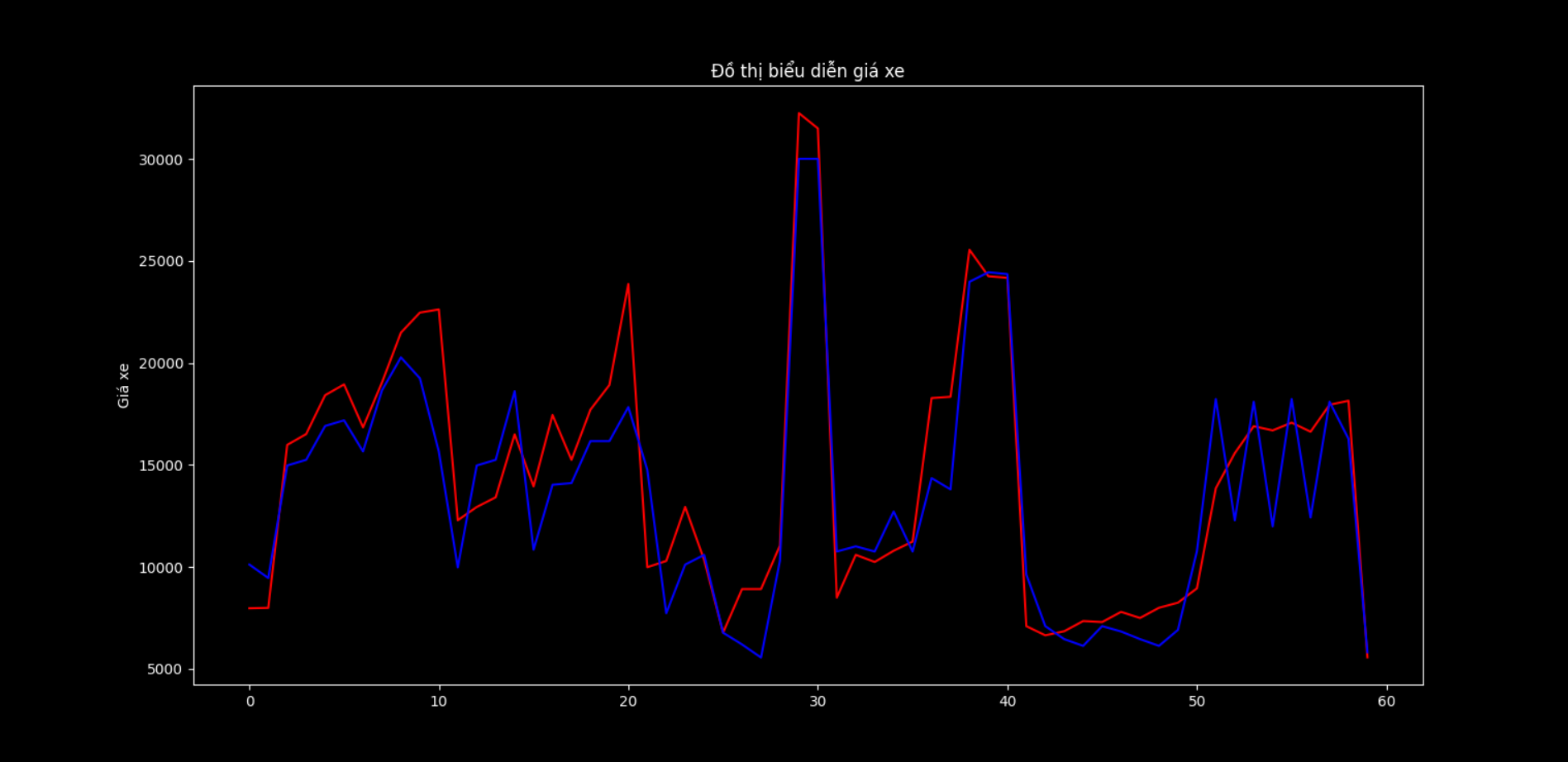
# Với xtest là [ 4. 186.7 68.3 56. 4. 134. 7. 142. ] thì kết quả dự đoán là 16284.45 kết quả thực tế là 18150.0

# Sai lệch : 1865.55

# Với xtest là [ 2. 157.3 63.8 50.8 4. 90. 9.4 68. ] thì kết quả dự đoán là 5820.58 kết quả thực tế là 5572.0

# Sai lệch : 248.58

# **3.2 Mô tả biểu đồ đánh giá mô hình**



Chú thích:

Giá trị thực tế: Đường màu đỏ

Giá trị dự đoán: Đường màu xanh

**3.3 Đánh giá chất lượng mô hình**

* Từ mô tả kết quả từ tập dữ liệu kiểm tra ta thấy kết quả dự đoán của mô hình chênh lệch sai số không nhiều so với kết quả thực tế,
* Từ đồ thị trên ta thấy các giá trị dự đoán tập trung gần với giá trị thực tế có sự phân bổ đều.

=> Chất lượng mô hình được xây dựng từ tập dữ liệu huấn luyện(training) là phù hợp và hoạt động tốt.

**KẾT LUẬN**

Mục tiêu của hồi quy là xác định các giá trị của các tham số để tối thiểu hóa tổng các giá trị còn lại bình phương (dự đoán chênh lệch between và quan sát) cho tập hợp các quan sát. Do hồi quy tuyến tính bị hạn chế trong việc kết hợp các hàm tuyến tính (đường thẳng / mặt phẳng) với dữ liệu, nên nó không phù hợp với dữ liệu trong thế giới thực vì các kỹ thuật tổng quát hơn như mạng thần kinh có thể mô hình hóa các hàm phi tuyến tính. Nhưng hồi quy tuyến tính có một số lợi thế thú vị:

* Hồi quy tuyến tính là phương pháp được sử dụng rộng rãi nhất, và nó được hiểu rõ.
* Đào tạo một mô hình hồi quy tuyến tính thường nhanh hơn nhiều so với các phương pháp như mạng lưới thần kinh.
* Các mô hình hồi quy tuyến tính rất đơn giản và yêu cầu bộ nhớ tối thiểu để thực hiện, vì vậy chúng hoạt động tốt trên các bộ điều khiển nhúng có không gian bộ nhớ hạn chế.
* Bằng cách kiểm tra độ lớn và dấu hiệu của các hệ số hồi quy, bạn có thể suy ra cách biến dự báo ảnh hưởng đến kết quả mục tiêu.
* Đây là một trong những thuật toán đơn giản nhất và có sẵn trong một số gói, ngay cả Microsoft Excel.

Thu hoạch của nhóm: Trau dồi thêm nhiều kiến thức về môn học “Học máy”. Qua bài tập lớn, các thành viên đã phần nào hiểu được những khái niệm và những công việc cơ bản nhất để xây dựng một mô hình học máy dựa trên các thuật toán và áp dụng được những lý thuyết đã học vào trong đề tài thực tế.

Và chúng em xin được cảm ơn thầy Nguyễn Thanh Tùng đã tận tình giảng dạy những kiến thức về machine learning. Trong một khoảng thời gian nghiên cứu và thực hiện đề tài, nhóm đã hoàn thành được đề tài: *Ứng dụng phương pháp hồi quy tuyến tính dự đoán giá Ô tô ở Ấn Độ.* Do kiến thức chúng em còn chưa thật vững chắc nên vẫn còn những khuyết điểm trong đề tài này. Mong thầy thông cảm và góp ý cho nhóm để đề tài của chúng em được hoàn thiện hơn và rút ra kinh nghiệm để thức hiện những đề tài sau này.

**TÀI LIỆU THAM KHẢO**

* Tham khảo slide “Học máy” của thầy Nguyễn Thanh Tùng
* Tài liệu file: <https://github.com/tiepvupsu/ebookMLCB> của thầy Vũ Hữu Tiệp
* Dataset:[https://www.kaggle.com/](https://www.kaggle.com/goyalshalini93/car-price-prediction-linear-regression-rfe/data?fbclid=IwAR0x2sXrpPkU5PBKFhvSrWrJL344T-Kwyx5Qtunm_z4TOkW3xHsPJ7Slg3s)