# CHƯƠNG 2: BÀI TOÁN HỌC MÁY

## 2.1. Bài toán hồi quy dự đoán giá xe ô tô Toyota Corolla cũ :

### 2.1.1. Mô tả bài toán :

Trong xu thế phát triển mạnh mẽ của xã hội như hiện nay, có rất nhiều người lựa chọn mua xe ô tô để phục vụ di chuyển, vừa để khẳng định đẳng cấp, vị thế của mình. Thế nhưng việc quyết định nên mua xe cũ hay xe mới lại trở thành một bài toán khiến cho nhiều người phải đau đầu suy nghĩ. Vấn đề được – mất trong việc mua xe ô tô cũ có lẽ đã trở thành vấn đề được rất nhiều người quan tâm. Bỏ qua những điểm hạn chế khi mua xe ô tô cũ so với ô tô mới như là các vấn đề bảo hành, nội thất, tiện nghi, ta có thể kể ra được một số ưu điểm khi mua ô tô cũ. Đầu tiên là việc tiết kiệm chi phí. Có thể nói, một trong những lợi thế lớn nhất khi mua ô tô cũ đó là tiết kiếm được một khoản chi phí tương đối lớn. Một ưu điểm nữa là ta có rất nhiều mẫu mã và chủng loại để lựa chọn ô tô sao cho phù hợp với túi tiền của bản thân và cho gia đình.

Trước những vấn đề đó, trong báo cáo học tập này, em đã sử dụng thuật toán **Hồi quy tuyến tính (Linear Regression)** để giải quyết bài toán “hồi quy dự đoán giá bán xe ô tô Toyota Corolla cũ” thông qua các thuộc tính cơ bản của xe ô tô.

### 2.1.2. Thu thập và mô tả dữ liệu

Trong bài toán dự đoán giá bán xe ô tô cũ này, em đã tìm hiểu và thu thập đươc tập dữ liệu các thuộc tính của xe ô tô cũ mà ở đây cụ thể là của nhãn hiệu Toyota Corolla. Tập dữ liệu nói trên được chúng em thu thập thông qua trang web **https://github.com/datailluminations/PredictingToyotaPricesBlog.** Tập dữ liệu bao gồm 1437 mẫu với 10 thuộc tính của xe ô tô.

### 2.1.3. Mô tả thuộc tính

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Price | Giá bán |
| 2 | Age | Tuổi thọ |
| 3 | KM | Số lượng km xe đã đi |
| 4 | HP | Horse Power: Mã lực |
| 5 | Met color | Màu kim loại |
| 6 | Automatic | Xe tự động |
| 7 | CC | Dung tích động cơ |
| 8 | Doors | Số cửa |
| 9 | Weight | Trọng lượng |
| 10 | FuelType | Loại nhiên liệu |

Bảng 2.1. Các thuộc tính của tập dữ liệu

Với 10 thuộc tính của xe trên đây, để phù hợp với yêu cầu là dự đoán giá bán xe ô tô cũ, chúng em chọn thuộc tính **Price** ở đây làm biến mục tiêu cần tìm, các thuộc tính còn lại làm biến giải thích.

### 2.1.4. Số lượng mẫu

Tập dữ liệu của chúng em bao gồm 1437 dòng tương ứng với số lượng mẫu. Dưới đây là một số mẫu có trong tập dữ liệu:

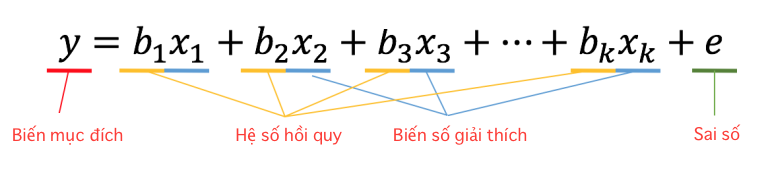
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Price** | **Age** | **KM** | **HP** | **MetColor** | **Automatic** | **CC** | **Doors** | **Weight** | **FuelType** |
| 13500 | 23 | 46986 | 90 | 1 | 0 | 2000 | 3 | 1165 | Diesel |
| 13750 | 23 | 72937 | 90 | 1 | 0 | 2000 | 3 | 1165 | Diesel |
| 13950 | 24 | 41711 | 90 | 1 | 0 | 2000 | 3 | 1165 | Diesel |
| 14950 | 26 | 48000 | 90 | 0 | 0 | 2000 | 3 | 1165 | Diesel |
| 13750 | 30 | 38500 | 90 | 0 | 0 | 2000 | 3 | 1170 | Diesel |
| 12950 | 32 | 61000 | 90 | 0 | 0 | 2000 | 3 | 1170 | Diesel |
| 16900 | 27 | 94612 | 90 | 1 | 0 | 2000 | 3 | 1245 | Diesel |

Bảng 2.2. Mô tả chi tiết tập dữ liệu

### 2.1.5. Xây dựng mô hình thuật toán

Học có giám sát (Supervised Learning) được chia ra làm 2 dạng lớn là Hồi quy (Regression) và Phân loại (Classification) dựa trên tập dữ liệu mẫu - tập huấn luyện (training data). Ở đây chúng em sẽ tìm hiểu về Hồi quy mà cụ thể là Hồi quy tuyến tính (Linear Regression).

**Khái niệm:** Mô hình hồi quy tuyến tính (Linear Regression) được biểu diễn dưới dạng công thức toán học, là loại mô hình dự đoán giá trị của biến mục tiêu dựa trên giá trị của biến giải thích.



Hình 2.1. Mô hình thuật toán Hồi quy tuyến tính

Ví dụ: dự đoán giá nhà ở Hà Nội dựa vào thông tin về diện tích, vị trí, năm xây dựng của ngôi nhà thì biến mục tiêu ở đây sẽ là giá nhà và **x = (x1,x2,x3)** với **x1**là diện tích, **x2** là vị trí và **x3** là năm xây dựng.

Có hai dạng hồi quy tuyến tính là đơn biến và đa biến. Hồi quy tuyến tính biến đơn là một kỹ thuật được sử dụng để mô hình hóa mối quan hệ giữa một biến độc lập đầu vào (biến giải thích) và biến phụ thuộc đầu ra bằng mô hình tuyến tính, ví dụ như một đường. Trường hợp tổng quát hơn là Hồi quy tuyến tính đa biến trong đó một mô hình được tạo cho mối quan hệ giữa nhiều biến đầu vào độc lập (biến giải thích) và biến phụ thuộc đầu ra. Mô hình vẫn tuyến tính trong đó đầu ra là sự kết hợp tuyến tính của các biến đầu vào.

Có một trường hợp tổng quát thứ ba được gọi là Hồi quy đa thức trong đó mô hình bây giờ trở thành tổ hợp phi tuyến tính của các biến giải thích, có thể có các biến số mũ, sin và cos, v.v. Tuy nhiên, điều này đòi hỏi kiến thức về cách dữ liệu liên quan đến đầu ra. Các mô hình hồi quy có thể được đào tạo bằng cách sử dụng thuật toán giảm dần độ dốc Stochastic Gradient Descent (SGD).

Ưu điểm và nhược điểm của thuật toán Hồi quy tuyến tính:

**Ưu điểm:**

* Mô hình đơn giản, dễ hiểu
* Dễ diễn giải hệ số hồi quy
* Nhận được kết quả khả thi khi dữ liệu nhỏ
* Nhiều cải tiến/mở rộng

**Nhược điểm:**

* Mô hình đơn giản nên khó dự đoán chính xác với dữ liệu có miền giá trị rộng
* Nhạy cảm với dữ liệu ngoại lai – do sử dụng phương pháp bình phương nhỏ nhất

Như đã trên trình bày ở trên, mô hình đơn giản nhất được sử dụng trong thuật toán Hồi quy tuyến tính có dạng:

Áp dụng vào bài toán này ta có thể thấy được **y** (biến mục tiêu) cần tìm ở đây là **Price** (giá nhà), còn **x1** là Age, **x2**là KM, **x3** là FuelType,… ở đây là độ lệch chuẩn, ,… là các tham số của mô hình. Các tham số này sẽ đánh giá mức độ quan trọng của các thuộc tính đến kết quả cuối cùng.

### 2.1.6. Huấn luyện mô hình

Sau khi có mô hình hồi quy, ta thực hiện đến bước huấn luyện chúng. Trước khi bắt đầu huấn luyện, ta chia tập dữ liệu thành hai phần: một phần sử dụng cho việc huấn luyện (Training) và phần còn lại sử dụng trong việc kiểm thử mô hình (Test). Huấn luyện mô hình là việc tìm ra các tham số tối ưu nhất thông qua bộ dữ liệu huấn luyện. Để làm được điều này chúng ta phái xác định được thước đo để biết được mô hình tốt hay không tốt (tham số có tối ưu hay không).

Một trong những phương pháp được sử dụng để ước lượng tham số là dùng biểu thức Root Mean Square Error (RMSE). Mục tiêu của việc huấn luyện đó là giảm thiểu giá trị của Mean Square Error (MSE).

MSE của mô hình hồi quy tuyến tính trên tập huấn luyện X được tính như sau:

Để tìm giá trị θ để phương trình MSE(***θ***) đạt cực tiểu có thể tìm được nhờ công thức sau:

Trong đó:

* là giá trị của ***θ*** *mà tại đó MSE(****θ)*** *đạt cực tiểu*
* là vector giá trị cần tìm bao gồm

### 2.1.7. Áp dụng kỹ thuật, thuật toán vào bài toán

Sau khi huấn luyện mô hình, ta sử dụng ngôn ngữ lập trình Python trong ứng dụng PyCharm IDE để thực hiện khởi tạo mô hình trên máy tính:

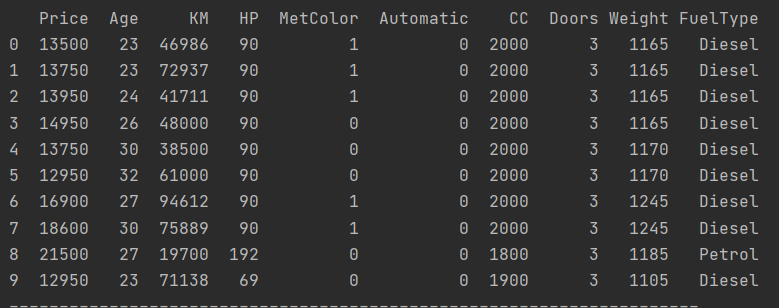
**Bước 1:** Khởi tạo thư viện

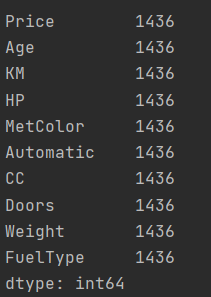
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.metrics import mean\_squared\_error  
from sklearn.model\_selection import cross\_val\_score

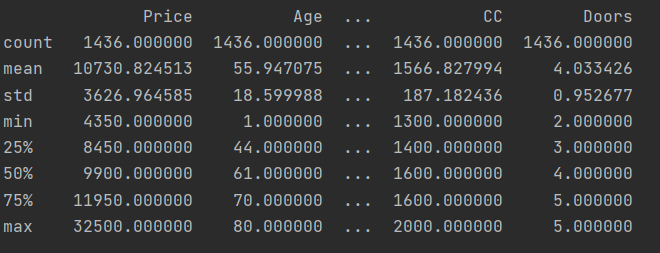
**Bước 2:** Xử lý dữ liệu

**Đọc file dữ liệu đã tải về**

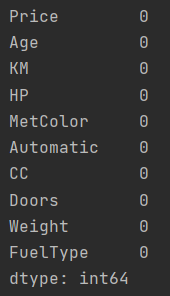
dataset = pd.read\_csv('database/ToyotaCorolla.csv')  
print(dataset.head(10))  
print("---------------------------------------------------------------------")

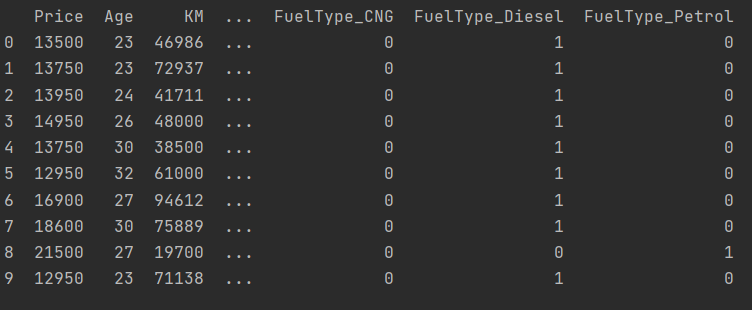
  
print(dataset.count())  
print("---------------------------------------------------------------------")

  
print(dataset.describe())  
print("---------------------------------------------------------------------")

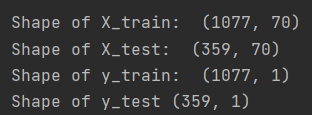


print(dataset.isnull().sum())  
print("---------------------------------------------------------------------")

  
  
# dataset = pd.get\_dummies(dataset, columns=['FuelType'])  
dataset = pd.get\_dummies(dataset)  
print(dataset.head(10))

  
X = dataset.drop(['Price'], axis = 1).values  
y = dataset.iloc[:, 0].values.reshape(-1,1)  
  
print("---------------------------------------------------------------------")

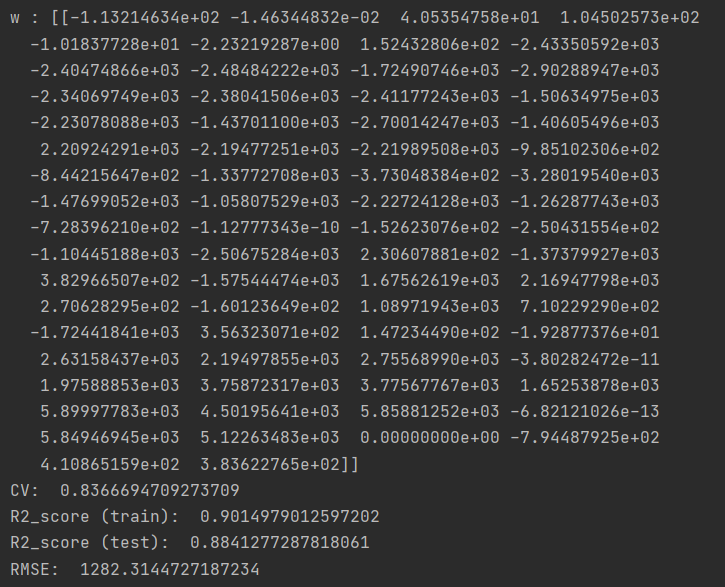
**Chia tập dữ liệu thành tập Train và tập Test**  
  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 42)  
print("Shape of X\_train: ", X\_train.shape)  
print("Shape of X\_test: ", X\_test.shape)  
print("Shape of y\_train: ", y\_train.shape)  
print("Shape of y\_test", y\_test.shape)  
print("---------------------------------------------------------------------")

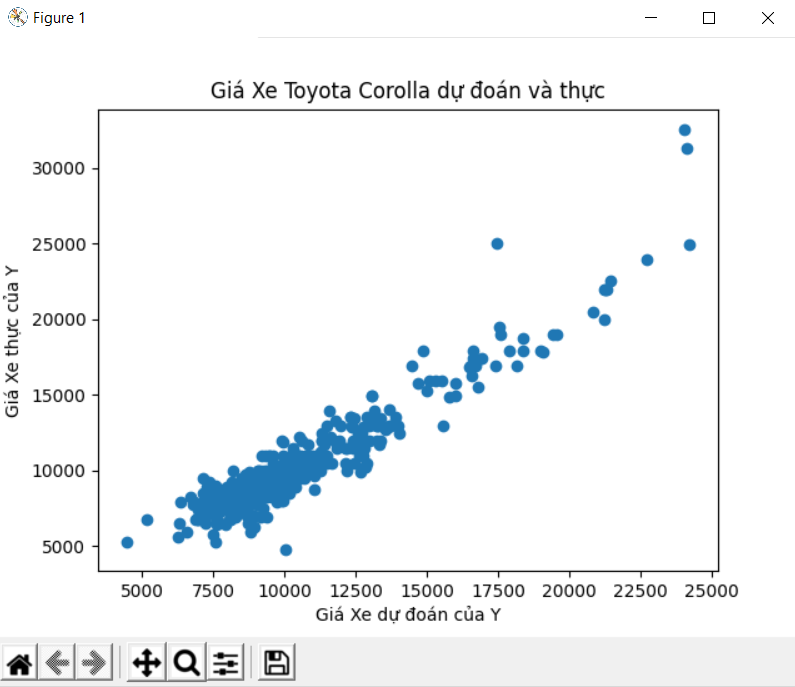


**Bước 3:** Áp dụng thuật toán Linear Regression giải quyết bài toán

#Linear Regression  
from sklearn.linear\_model import LinearRegression  
regressor\_linear = LinearRegression()  
regressor\_linear.fit(X\_train, y\_train)  
  
from sklearn.metrics import r2\_score  
  
# Predicting Cross Validation Score the Test set results  
#Dự đoán kết quả giao của tập Test  
cv\_linear = cross\_val\_score(estimator = regressor\_linear, X = X\_train, y = y\_train, cv = 10)  
  
# Predicting R2 Score the Train set results  
y\_pred\_linear\_train = regressor\_linear.predict(X\_train)  
r2\_score\_linear\_train = r2\_score(y\_train, y\_pred\_linear\_train)  
  
# Predicting R2 Score the Test set results  
y\_pred\_linear\_test = regressor\_linear.predict(X\_test)  
r2\_score\_linear\_test = r2\_score(y\_test, y\_pred\_linear\_test)  
  
# Predicting RMSE the Test set results  
rmse\_linear = (np.sqrt(mean\_squared\_error(y\_test, y\_pred\_linear\_test)))

print("w :",regressor\_linear.coef\_)#Sai so  
print("CV: ", cv\_linear.mean())  
print('R2\_score (train): ', r2\_score\_linear\_train)  
print('R2\_score (test): ', r2\_score\_linear\_test)  
print("RMSE: ", rmse\_linear)#Trung binh sai so

  
  
plt.scatter(y\_pred\_linear\_test,y\_test)  
plt.xlabel("Giá Xe dự đoán của Y")  
plt.ylabel("Giá Xe thực của Y")  
plt.title("Giá Xe Toyota Corolla dự đoán và thực")  
plt.show()



### 2.1.7. Đánh giá mô hình

Sau khi việc huấn luyện hoàn tất, ta cần xem mô hình có tốt không bằng cách dùng bước Đánh giá. Việc đánh giá cho phép chúng ta kiểm tra mô hình với những dữ liệu chưa từng được dùng cho huấn luyện.

CV: 0.8366694709273709

R2\_score (train): 0.9014979012597202

R2\_score (test): 0.8841277287818061

RMSE: 1282.3144727187234