ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

MATH FOR AI

LINEAR REGRESSION

NHÓM 5

Giáo viên	Cấn Trần Thành Trung
	Nguyễn Ngọc Toàn
23122003	Nguyễn Văn Linh
23122022	Trần Hoàng Gia Bảo
23122026	Trần Chấn Hiệp
23122040	Nguyễn Thị Mỹ Kim

TP. HỒ CHÍ MINH, THÁNG 3/2025

Mục lục

Tóm tắt

Dữ liệu dạng bảng dự đoán giá xe hơi cũ bao gồm 19 thuộc tính (features). Việc xử lý dữ liệu nhiều chiều (high-dimensionality) khá thách thức, đặc biệt khi có thể tồn tại nhiễu (outliers) và các thuộc tính chưa được chuẩn hóa dưới dạng số (numeric labels). Trong bài báo cáo này, nhóm sử dụng hồi quy tuyến tính (linear regression) để giả định mối quan hệ tuyến tính giữa biến phụ thuộc (`Price`) và các biến độc lập (features), qua đó giải thích ảnh hưởng của từng feature lên `Price` thông qua trọng số (coefficient). Tuy nhiên, qua thực nghiệm, ta nhận thấy `Price` có mối quan hệ phi tuyến với các feature thay vì tuyến tính như giả định ban đầu. Để giải quyết vấn đề này, hồi quy tuyến tính có thể được điều chỉnh bằng cách xử lý các mối quan hệ phi tuyến thông qua biến đổi đặc trưng (feature transformation). Qua quan sát và thực nghiệm, nhóm chọn các feature ảnh hưởng lớn sau khi biến đổi đặc trưng để đưa vào hồi quy đa thức (polynomial regression) mà không gây overfitting.

1 Giới thiệu

1.1 Dữ liệu dự đoán giá xe hơi cũ

Tập dữ liệu được sử dụng trong nghiên cứu này có định dạng CSV và bao gồm 19 thuộc tính (features), phục vụ cho bài toán dự đoán giá xe hơi cũ. Dữ liệu được tổ chức dưới dạng bảng, trong đó mỗi dòng đại diện cho một mẫu dữ liệu (sample), và mỗi cột tương ứng với một thuộc tính mô tả xe hơi. Các thuộc tính trong tập dữ liệu có thể được phân thành bốn nhóm chính như sau:

1. **Chi tiết kỹ thuật (Vehicle Specifications)**
   - **Make**: Thương hiệu xe.
   - **Model**: Dòng xe.
   - **Year**: Năm sản xuất.
   - **Transmission**: Loại hộp số.
   - **Length, Width, Height**: Kích thước xe (dài, rộng, cao).
   - **Color**: Màu xe.
2. **Động cơ và hiệu suất (Engine and Performance)**
   - **Engine**: Dung tích động cơ.
   - **Fuel Type**: Loại nhiên liệu sử dụng.
   - **Fuel Tank Capacity**: Dung tích bình nhiên liệu.
   - **Max Power**: Công suất tối đa.
   - **Max Torque**: Mô-men xoắn cực đại.
   - **DriveTrain**: Hệ thống truyền động.
   - **Seating Capacity**: Số lượng chỗ ngồi.
3. **Tình trạng và lịch sử sở hữu (Condition and Ownership History)**
   - **Owner**: Số đời chủ sở hữu của xe.
   - **Kilometer**: Quãng đường đã di chuyển (tính theo kilomet).

4. **Yếu tố thị trường (Market Factors)**
   - **Location**: Địa điểm bán xe.
   - **Seller Type**: Loại người bán

1.2 Linear Regression

Bảng 1: **Ký hiệu trong hồi quy tuyến tính**

Ký hiệu	Diễn giải
n	Số lượng mẫu dữ liệu (samples)
m	Số lượng đặc trưng (features)
$X \in \mathbb{R}^{n \times m}$	Ma trận đặc trưng (dataset) với n mẫu và m đặc trưng
$Y \in \mathbb{R}^n$	Vector giá trị mục tiêu (label)
$W \in \mathbb{R}^m$	Vector trọng số (weights)
$T \in \mathbb{R}^n$	Vector biểu diễn đặc trưng T của tập dữ liệu, với n mẫu dữ liệu
$b \in \mathbb{R}$	Hệ số điều chỉnh (bias)
$\hat{Y} \in \mathbb{R}^n$	Vector giá trị dự đoán (predictions)
$f : \mathbb{R}^{n \times m} \to \mathbb{R}^n$	Hàm hồi quy tuyến tính ánh xạ tập dữ liệu X thành vector dự đoán $\hat{Y} = f(X) = XW + b$
$f : \mathbb{R}^m \to \mathbb{R}$	Hàm hồi quy ánh xạ một mẫu dữ liệu x_i thành giá trị dự đoán $\hat{y}_i = f(x_i) = x_i W + b$

1.2.1 Biểu diễn toán học

Hồi quy tuyến tính là một phương pháp học có giám sát (supervised learning) được sử dụng để dự đoán giá trị y thông qua một tổ hợp tuyến tính của các đặc trưng (features). Mô hình có dạng:

$$f : \mathbb{R}^m \to \mathbb{R}, \quad \hat{y} = f(x) = xW + b = x_1 w_1 + x_2 w_2 + \cdots + x_m w_m + b$$

Khi đó $x = [x_1, x_2, \ldots, x_m]$ là một vector hàng đại diện cho một điểm dữ liệu với m đặc trưng, $W = [w_1, w_2, \ldots, w_m]^T$ là vector cột chứa trọng số của các đặc trưng và b là hệ số điều chỉnh (bias).

Để tổng quát hóa cho một tập dữ liệu gồm n mẫu, ta sử dụng vector cột $Y = [y_1, y_2, \ldots, y_n]^T \in \mathbb{R}^n$ với dữ liệu X biểu diễn dưới dạng ma trận:

$$\mathbf{X} := \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

Khi đó, mô hình có dạng:

$$f : \mathbb{R}^{n \times m} \to \mathbb{R}^n, \quad \hat{Y} = f(X) = XW + b$$

1.2.2 Hàm mất mát - MSE

Công thức tổng quát cho hàm MSE (Mean Squared Error):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i \cdot W + b - y_i)^2$$

Trong đó:
N : số dữ liệu dùng để huấn luyện mô hình
x_i: hàng dữ liệu thứ i
W: là vector trọng số
b: bias
y_i: là giá trị đúng của cột dữ liệu thứ i

Áp vào dữ liệu của chúng ta, để tính toán thuận tiện hơn, ta đã chuyển ma trận dữ liệu đầu vào thành ma trận X kích thước $n \times m$ (với n là số lượng dữ liệu, m là số thuộc tính). Khi đó, $XW + b$ sẽ cho ta một vector dọc là các kết quả mà mô hình đoán, Y là vector dọc chứa các giá trị đúng của dữ liệu.

Thư viện Numpy giúp chúng ta xử lý chúng một cách dễ dàng qua các tính chất như broadcast, elementwise, ...

Mục tiêu là ta cần tối thiểu giá trị của hàm MSE thông qua việc cập nhật W và b. Từ đây, ta quy về bài toán tìm cực tiểu của hàm nhiều biến, trong đó các biến là $w_1, w_2, ...$ của W và b.

Việc tìm cực trị chính xác theo Toán học là rất khó khăn và không hợp lý khi số lượng thuộc tính thay đổi. Vì vậy, ta sử dụng một phương pháp ước lượng cực tiểu là **Gradient Descent**.

1.2.3 Hàm mất mát - MAE

Công thức tổng quát cho hàm MAE (Mean Absolute Error):

$$\frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| = \frac{1}{N}\sum_{i=1}^{N}|x_i \cdot W + b - y_i|$$

Trong đó:
N : số dữ liệu dùng để huấn luyện mô hình
x_i: hàng dữ liệu thứ i
W: là vector trọng số
b: bias
y_i: là giá trị đúng của cột dữ liệu thứ i
Tương tự như hàm MSE đây là một hàm khác cũng được dùng để tính toán độ chính xác cho mô hình.

1.2.4 Hàm R^2

Công thức tổng quát cho hàm R^2:

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

Trong đó:
N : số lượng dữ liệu dùng để huấn luyện mô hình

y_i : là giá trị đúng của dữ liệu thứ i
$\hat{y}_i$: là giá trị mà mô hình dự đoán cho dữ liệu thứ i
$\overline{y}$: là giá trị trung bình của tập giá trị đúng của N dữ liệu

R^2 : đo lường tỉ lệ phương sai của y được giải thích bởi mô hình
$R^2 = 1$: mô hình giải thích 100% phương sai của y, đầu vào và ra có mối quan hệ tuyến tính hoàn hảo
$R^2 = 0$: không có sự tương quan giữa dự đoán và thực tế

1.2.5 Kết luận

Các hàm MSE, MAE và R^2 đều tập trung vào tính độ phân tán dữ liệu của mô hình, các chỉ số này phản ánh mức độ hiệu quả của mô hình được huấn luyện thông qua việc so sánh lấy tổng của bình phương chênh lệch giữa điểm dự đoán và thực tế (ở tính phương sai thì ta coi như kết quả của các lần dự đoán luôn là mean của tập dữ liệu)

1.2.6 Gradient Descent

- Trước tiên, ta có công thức tổng quát cho Gradient Descent là:

$$X_{t+1} = X_t - \eta \nabla_f$$

  Trong đó:
  X_t : vị trí thứ t
  η : learning rate
  ∇_f : vector gradient của f tại điểm X_t

- Giải thích cho thuật toán trên, ta có tính chất sau:
  $D_u f = \nabla_f \cdot u = |\nabla_f| \cdot |u|.cos(\nabla_f, u) \geq -|\nabla_f| \cdot |u| = -|\nabla_f|$
  Từ đây ta thấy nếu vector u có hướng trùng với vector $-\nabla_f$ thì hàm f sẽ giảm do đạo hàm theo hướng âm nếu bước nhảy là nhỏ thích hợp. Chính vì lý do trên, ta sẽ ước lượng điểm cực tiểu cho hàm thông qua từng bước nhảy liên tiếp nhau và theo công thức Gradient Descent thì $X_{t+1} - X_t = \eta.(-\nabla_f)$ có nghĩa là điểm thứ $t+1$ được chọn theo hướng của $-\nabla_f$ và có độ lớn là $\eta|\nabla_f|$, từ đó có thể tìm được ước lượng điểm cực trị thích hợp.

- Nếu learing rate η quá lớn thì ta sẽ không tìm được do có thể bước quá lớn không còn đảm bảo nhận xét điểm mới làm giảm giá trị hàm f, ngược lại nếu quá nhỏ thì lại mất quá nhiều bước nhảy để tìm được ước lượng. Chính vì vậy, việc chọn learning rate phù hợp là rất quan trọng trong việc huấn luyện mô hình.

- Tóm lại, ta sẽ dùng Gradient Descent lên các hàm mất mát (Loss Function) với các biến là $w_1, w_2, ...$ của W và b để ước lượng điểm cực tiểu nhằm tối ưu hóa mô hình.

1.2.7 Cài đặt Gradient Descent

- Ta đã biết, dữ liệu đầu vào của ta là ma trận $X \in M_{N \times m}(\mathbb{R})$, $W \in M_{m \times 1}(\mathbb{R})$ là vector cột chứa m trọng số $w_1, ..., w_m$ và b là bias; (N: là số mẫu dữ liệu dùng để huấn luyện, và m: là số thuộc tính mà chúng ta chọn để hình thành mô hình)

- Khi đó, ta gọi $\hat{Y} = X \cdot W + b \in M_{N\times 1}(\mathbb{R})$ là vector cột mà mô hình ta dự đoán cho N mẫu dữ liệu (ở đây ta sử dụng tính **broadcast**, tức là $A_{n\times m} + C = A_{ij} + C$, hay ta cộng thêm C vào mỗi phần tử của ma trận/vector); $Y \in M_{N\times 1}(\mathbb{R})$ là kết quả đúng của N mẫu ta chọn.

- Trong đồ án này, ta chỉ dùng hàm MSE làm hàm mất mát. Dưới đây là công thức của MSE được cài đặt:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i \cdot W + b - y_i)^2$$

với $x_i \in M_{1\times m}(\mathbb{R})$ là vector dòng của input mẫu thứ i. Từ đây, ta có:

$$\frac{\partial(MSE)}{\partial w_i} = \frac{\partial(\frac{1}{N}\sum_{j=1}^{N}(x_{j1}w_1 + ... + x_{ji}w_i + ... + x_{jm}w_m + b - y_j)^2)}{\partial w_i} = \frac{2}{N}\sum_{j=1}^{N}(x_j \cdot W + b - y_j)x_{ji}$$

$$\frac{\partial(MSE)}{\partial b} = \frac{\partial(\frac{1}{N}\sum_{j=1}^{N}(x_{j1}w_1 + ... + x_{ji}w_i + ... + x_{jm}w_m + b - y_j)^2)}{\partial b} = \frac{2}{N}\sum_{j=1}^{N}(x_j \cdot W + b - y_j)$$

Vậy vector

$$\nabla(MSE) = \begin{bmatrix} \frac{\partial(MSE)}{\partial w_1} \\ \vdots \\ \frac{\partial(MSE)}{\partial w_m} \\ \frac{\partial(MSE)}{\partial b} \end{bmatrix} = \frac{2}{N}\begin{bmatrix} \sum_{j=1}^{N}(x_j \cdot W + b - y_j)x_{j1} \\ \vdots \\ \sum_{j=1}^{N}(x_j \cdot W + b - y_j)x_{jm} \\ \sum_{j=1}^{N}(x_j \cdot W + b - y_j) \end{bmatrix} \quad (\Delta)$$

Ta lại có:

$$\frac{2}{N}X^T \cdot (XW + b - Y) = \frac{2}{N}\begin{bmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{Nm} \end{bmatrix}^T \cdot \begin{bmatrix} x_1 \cdot W + b - y_1 \\ \vdots \\ x_N \cdot W + b - y_N \end{bmatrix}$$

$$= \frac{2}{N}\begin{bmatrix} \sum_{j=1}^{N}(x_j \cdot W + b - y_j) \cdot x_{j1} \\ \vdots \\ \sum_{j=1}^{N}(x_j \cdot W + b - y_j) \cdot x_{jm} \end{bmatrix}$$

Thay vào (Δ) ta được:

$$\nabla(MSE) = \frac{2}{N}\begin{bmatrix} X^T \cdot (XW + b - Y) \\ \sum_{j=1}^{N}(x_j \cdot W + b - y_j) \end{bmatrix}$$

Tới đây ta áp dụng **Gradient Descent** vào hàm MSE với bộ các biến là $w_1, ..., w_m, b$ ta có ngay:

$$U = \begin{bmatrix} w_1 \\ \vdots \\ w_m \\ b \end{bmatrix} = \begin{bmatrix} W \\ b \end{bmatrix}; U_{t+1} = U_t - \eta.\nabla(MSE) = U_t - \eta.\frac{2}{N}\begin{bmatrix} X^T \cdot (XW_t + b_t - Y) \\ \sum_{j=1}^{N}(x_j \cdot W_t + b_t - y_j) \end{bmatrix}.$$

Ở đây do tính chất của vector nên ta sẽ tách U thành W và b và cập nhật song song 2 biến này. Từ đây, ta được công thức cuối cùng là:

$$W_{t+1} = W_t - \eta.\frac{2}{N}X^T \cdot (XW_t + b_t - Y)$$

$$b_{t+1} = b_t - \eta.\frac{2}{N}\sum_{j=1}^{N}(x_j \cdot W_t + b_t - y_j)$$

- Tới đây, ta đã hoàn tất phần xử lí **Gradient Descent** cho hàm **MSE**. Đây chính là chứng minh tường minh cho phần code thuật toán **Gradient Descent** của đồ án này.

2 Phương pháp

2.1 Chuẩn hóa Z-score (standard data)

Chuẩn hóa Z-score biến đổi dữ liệu về cùng một thang đo, giúp các đặc trưng có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1 để giảm sự chênh lệch giữa các biến.
Công thức chuẩn hóa Z-score

$$Z = \frac{X - \mu}{\sigma}$$

Khi đó, $X \in \mathbb{R}^{n\times m}$ là dữ liệu gốc cần chuẩn hóa, $\mu \in \mathbb{R}^m$ là vector giá trị trung bình và $\sigma \in \mathbb{R}^m$ là độ lệch chuẩn của từng đặc trưng trong dữ liệu X.

- $Z = 0$: Giá trị dữ liệu bằng trung bình.
- $Z > 0$: Giá trị dữ liệu lớn hơn trung bình.
- $Z < 0$: Giá trị dữ liệu nhỏ hơn trung bình.

2.2 Tính hệ số tương quan (correlation)

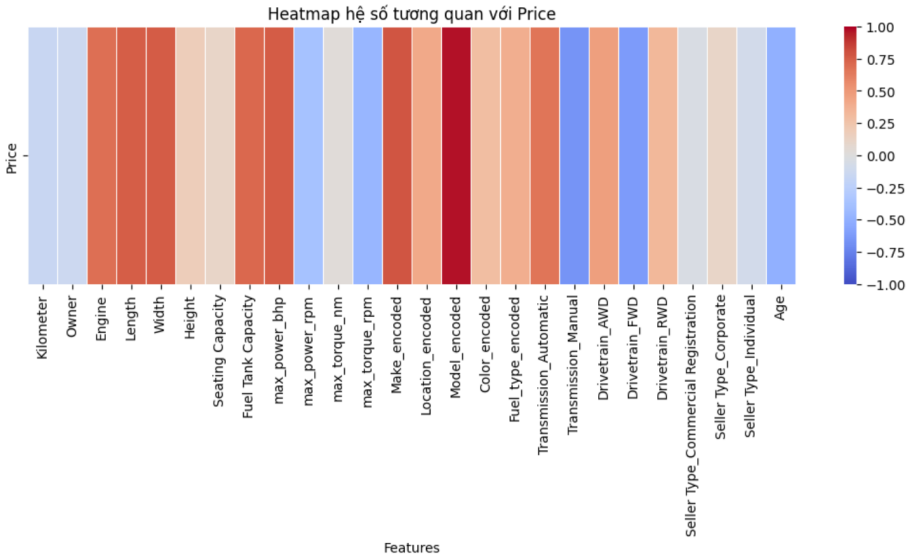
Tính hệ số tương quan r là phương pháp thống kê đo độ mạnh và hướng của mối quan hệ giữa hai biến (giả sử vector feature T và vector Y):

$$r(T,Y) = \frac{\sum(t_i - \bar{T})(y_i - \bar{Y})}{\sqrt{\sum(t_i - \bar{T})^2 \sum(y_i - \bar{Y})^2}},\ r \in [-1,1]$$

Khi đó, $T = [t_1, \ldots, t_n]^T \in \mathbb{R}^n$ là vector cột biểu diễn feature T và $\bar{T}, \bar{Y}$ là giá trị trung bình của vector T và Y.
Độ mạnh: khi $r(x,y) \to 0$ thì mối quan hệ giữa hai biến x, y này càng yếu.
Hướng: được thể hiện qua dấu của r:

- $r(x,y) > 0$: x tăng y tăng
- $r(x,y) < 0$: x tăng y giảm hoặc y tăng x giảm
- $r(x,y) = 0$: x và y không có sự tương quan.

Ta sử dụng hệ số tương quan để loại bỏ các feature không quá quan trọng, giúp cho việc chọn các feature đưa vào mô hình hiệu quả hơn.
Quan sát hình 1, các đặc trưng có thang màu trong khoảng từ $[-0.3, 0.3]$ là các đặc trưng có mối quan hệ yếu với vector Y nên ta có thể lựa chọn loại bỏ các cột này.



Hình 1: Hệ số tương quan giữa Y và các cột feature T

2.3 Biến đổi logarithm (Logarithmic Transformations)

Biến đổi logarithm là biến đổi đơn điệu (monotone transformation), nghĩa là phép biến đổi này giữ nguyên thứ tự của các giá trị x đưa vào hàm f. Giả sử hàm logarithm $f(x) = \log_b(x)$ với $b > 1$ là một hàm đơn điệu tăng vì:

- Đạo hàm luôn dương, $f(x)$ tăng khi x tăng.

$$\nabla_x f = \frac{1}{x \ln b} > 0, \ \forall x > 0$$

- Dữ liệu vẫn được giữ nguyên trật tự sau khi biến đổi

$$x_1 > x_2 > 0 \rightarrow \log_b(x_1) > \log_b(x_1)$$

Hàm logarithm trong mô hình tuyến tính

Trong các mô hình tuyến tính cơ bản, có thể sử dụng một số hàm logarithm cơ số $e \approx 2.718$ kết hợp như sau (các diễn giải bên dưới mặc định log là log cơ số e):

Bảng 2: log-transformation

Dạng	**Diễn giải**
linear	$\hat{Y} = XW + b$
linear-log	$\hat{Y} = \log(X_i)W + b$
log-linear	$\log(\hat{Y}) = XW + b$

Mô hình linear

Trong mô hình: $Y = XW + b$, khi thay đổi X_i một đơn vị, giá trị Y thay đổi một giá trị là W_i đơn vị.

$$Y + W_i \leftrightarrow X + 1$$

Mô hình linear-log

Trong mô hình linear-log: $\hat{Y} = \log(X_i)W + b$, khi thay đổi $\log(X_i)$ một đơn vị, giá trị Y thay đổi một giá trị là W_i đơn vị. Ta xem xét biểu thức sau:

$$\log(X) + 1 = \log(eX)$$

Khi cộng 1 cho $\log X$ mang ý nghĩa là nhân bản thân giá trị X với cơ số e.
Điều này cũng tương tụ với mô hình linear-log, muốn thay đổi Y một đơn vị thì ta phải thay đổi X giá trị là $e \approx 1.72828$ lần.

$$Y + 1 \leftrightarrow eX$$

Mô hình log-linear

Trong mô hình log-linear: $\log(\hat{Y}) = XW + b$, khi thay đổi X một đơn vị, giá trị $\log(Y)$ thay đổi một giá trị là W_i đơn vị. Điều này còn có nghĩa, khi X tăng một đơn vị, giá trị Y tăng e^W lần.

$$Y.e^W \leftrightarrow X + 1$$

Đánh giá log-transformation lên Price

Biến đổi logarithm trong mô hình hồi qui tuyến tính xử lí mối quan hệ không tuyến tính (non-linear) giữa biến phụ thuộc (vector Y) và biến độc lập (ma trận X) mà vẫn giữ nguyên tính thứ tự của X và Y. Trong bài toán này, quan sát biểu đồ histogram của vector Y khi chưa biến đổi logarithm (hình 2). Dữ liệu cột `Price` phân tán lệch trái (right-skewed), nghĩa là đa số các sample trong dữ liệu bunched ở những giá trị thấp.

Nếu sử dụng biến đổi logarithm cho cột Price (hình 3, rõ ràng các giá trị trong cột này trông phân phối chuẩn hơn.

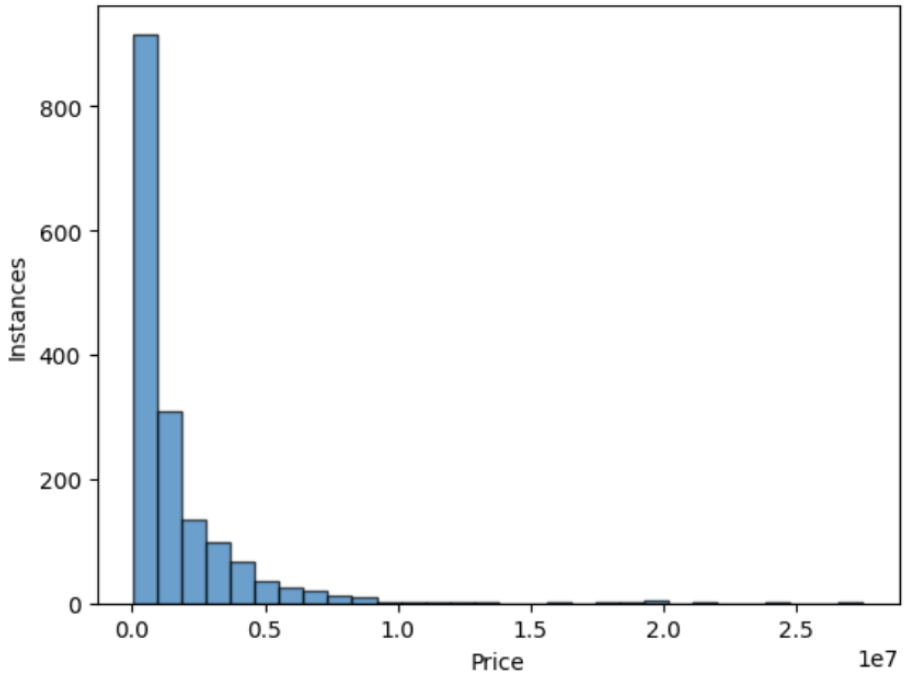
2.4 Hồi qui đơn thức và hồi qui đa thức

2.4.1 Hồi qui đơn thức

- Công thức của hồi quy đơn thức:

$$y = \beta_0 + \beta_1 x_1$$

- Đây là mô hình đơn giản nhất cho mô hình hồi qui.
- Công thức cho mô hình quá đơn giản nên không thích hợp cho các bài toán mà đầu vào và đầu ra các có đặc điểm không tuyến tính. Vậy nên trong đồ án này mô hình này chỉ đóng vai trò là nền tảng để sử dụng cho mô hình hồi quy đa thức.



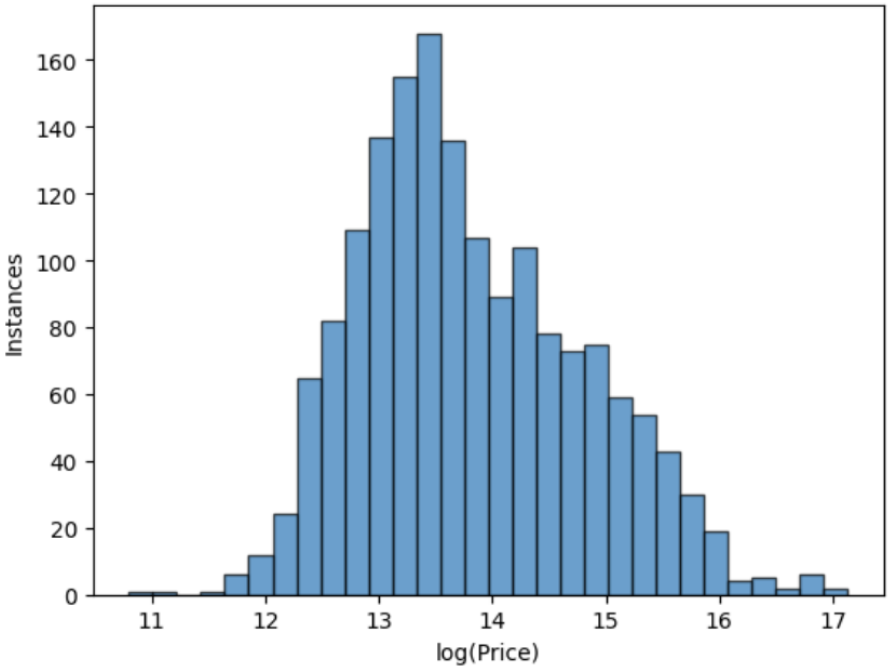
Hình 2: Biểu đồ histogram cho cột Price

2.4.2 Hồi qui đa thức

- Công thức cho hồi quy đa thức:

$$y = \beta_0 + \sum_{k=1}^{K} \beta_k \cdot \prod_{m=1}^{M} x_m^{a_{km}}$$

- Trong đồ án này, các mô hình đều sử dụng dựa trên công thức trên. Đây là một mô hình cho đầy đủ các yếu tố giúp ta xây dựng một mô hình hồi qui mà ở đó, dữ liệu đầu ra có thể không tuyến tính mạnh với nhiều đặc điểm dữ liệu đầu vào. Đặc biệt, mô hình hồi quy đa thức rất hữu ích khi các mối quan hệ không phải là tuyến tính đơn giản mà có sự phức tạp như đường cong, các mối quan hệ bậc cao hoặc các tương tác giữa các biến.
- Bên cạnh công thức trên, khi ta muốn một biến của ta cần lấy theo hàm log hay lại là e^x thì ta vẫn có thể làm được thông qua việc tạo một feature mới từ feature ban đầu của ta và áp dụng nó cho mô hình hồi quy đa thức. Đây được gọi là kĩ thuật **Feature Transformation**



Hình 3: Biểu đồ histogram cho cột log(Price)

3 Thực nghiệm

3.1 Dataset

3.1.1 Training Dataset

Dataset cho dữ liệu dự đoán giá xe cũ là dữ liệu dạng bảng, với 1647 mẫu và 19 features. Đây là một tập dữ liệu nhiều chiều, yêu cầu các kĩ thuật xử lý khi các feature của thuộc kiểu dữ liệu hỗn hợp (chuỗi ký tự và số), và chứa các giá trị bị thiếu (NaN). Đồng thời, dữ liệu cũng chứa các điểm dữ liệu gây nhiễu hoặc các feature không đóng góp đáng kể đến hiệu suất mô hình dự đoán (được đánh giá trên MSE và R-squared score). Trong dữ liệu cũng tồn tại các feature bị lệch (skewed) không tuân theo phân phối chuẩn, ảnh hưởng đến tính tuyến tính của mô hình. Dựa vào dataset này, chúng ta sẽ đánh giá và lựa chọn ra mô hình phù hợp, nâng cao hiệu suất dự đoán giá xe.

3.1.2 Data preparation

Xử lý dữ liệu chuỗi kết hợp số (String + Int Data Processing)

1. Feature `Engine`: Loại bỏ đơn vị `cc` và chuyển đổi giá trị thành kiểu `float`.
2. Feature `Max Power`:: Lấy thông tin công suất cực đại (bhp) và vòng quay (rpm) từ chuỗi `string` và chuyển đổi chúng thành kiểu `int`, tạo thành 2 feature mới là `max_power_bhp`, `max_power_rpm`.

3. Feature `Max Torque`: Tương tụ, lấy thông tin mô-men xoắn cực đại (Nm) và vòng quay (rpm) từ chuỗi `string` và chuyển đổi chúng thành kiểu `int`, tạo thành 2 feature mới là `max_torque_nm, max_torque_rpm`.

Xử lý dữ liệu chuỗi (String Data Processing)

Dữ liệu chuỗi trong tập dữ liệu cần được chuyển đổi thành dạng số để có thể sử dụng trong các mô hình học máy. Các phương pháp chính được áp dụng bao gồm:

1. Mã hóa thứ tự (Ordinal Encoding): chuyển đổi các biến phân loại (categorical variables) thành giá trị số theo thứ tự có sẵn. Khi đó, mỗi unique values trong feature sẽ được mapping với một con số (có thứ tự). Với feature *Owner*, sử dụng mã hóa thứ tự để ánh xạ số lần chủ sở hữu xe thành giá trị số như trong bảng 3.

Bảng 3: Ordinal Encode cho feature 'Owner'

Chủ sở hữu	**Label**
First	1
Second	2
Third	3
Fourth	4
4 or More	5
UnRegistered Car	0

2. Mã hóa dựa trên giá trị trung bình (Target Encoding): Ý tưởng của việc sử dụng phương pháp này đến từ việc sử dụng Ordinal Encoding, làm sao để mã hóa các giá trị dữ liệu có mức độ quan trọng khác nhau đối với giá xe mà không cần phải encode bằng tay? Sử dụng **giá tiền trung bình (`Price`) của từng loại xe** để sắp xếp thứ tự là một phương pháp cơ bản để giải quyết các feature có nhiều (hoặc quá nhiều) unique values trong feature đó. Như vậy, với feature `Make`, hãng có giá xe trung bình cao hơn sẽ được gán nhãn cao hơn nhờ có giá trị trung bình cao hơn. (Giả sử xe *Porche* sẽ có giá trung bình cao *Toyota*). Các feature khác như `Location, Color, Fuel Type, Model` cũng được mã hóa tương tự bằng giá trị trung bình của giá xe.

Bảng 4: Pseudocode cho Target Encoding

Target Encoding
Input: Cột feature **T** kiểu string **Output**: Cột feature **T** được label **G** ← **Group** *data* by feature **T** **M** ← **Mean** `Price` **foreach** T-type ∈ *G* Return **Sorted(M)** in **Ascending** order

3. Mã hóa One-Hot Encoding: được sử dụng đối với các thuộc tính không có thứ tự rõ ràng. Khi đó, mỗi unique values trong feature sẽ được biến thành một cột nhị phân. Với feature `Drivetrain`, sử dụng One-Hot Encoding như bảng 5. Các feauture như `Transmission, Seller Type` cũng được One-Hot Encoding.

Bảng 5: Feature Drivetrain trước và sau khi dùng One-Hot Encoding

Drivetrain
AWD
FWD
RWD
AWD
FWD

AWD	RWD	FWD
1	0	0
0	0	1
0	1	0
1	0	0
0	0	1

Chuyển đổi dữ liệu năm sản xuất thành tuổi của xe

Giá xe thường giảm theo tuổi xe (`Age`), nên mô hình có thể học tốt hơn mối quan hệ này nếu sử dụng Age. Trong bài toán này, tùy vào góc độ mà ta lựa chọn việc giữ lại `Year` hay `Age` thì tốt cho mô hình hơn. Tuy nhiên, nếu giữ nguyên Year, mô hình có thể hiểu sai rằng xe sản xuất năm 2010 và 2020 có mức chênh lệch lớn về giá trị, trong khi thực tế, khoảng cách thời gian (tức tuổi xe) mới là yếu tố quan trọng hơn.
Để tính toán tuổi của xe, ta sử dụng công thức:

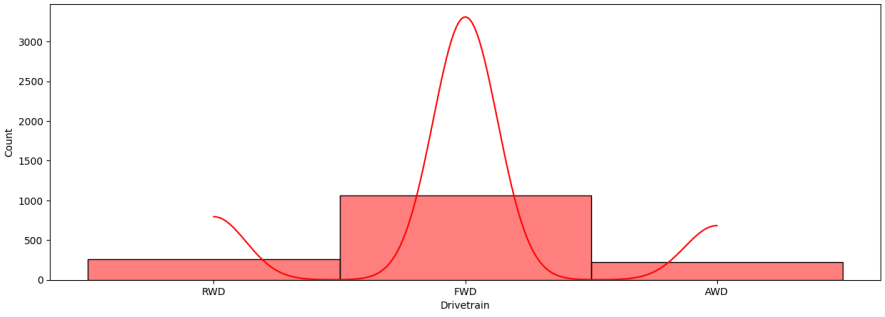
$$\text{Age} = \max(\text{Year}) - \text{Year} + 1$$

Ví dụ, nếu $\max(\text{Year}) = 2022$ và năm sản xuất của xe là 2020, thì tuổi của xe sẽ là 3. Khi cùng loại xe, xe có tuổi 5 sẽ có giá thành thấp hơn xe có tuổi 3.

Xử lí các dữ liệu NaN

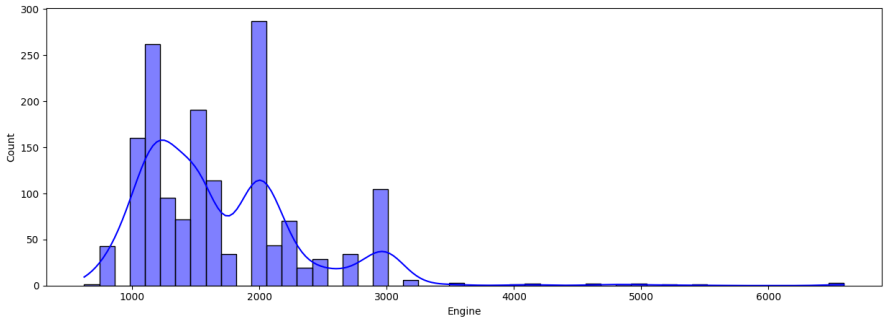
Khi dữ liệu có giá trị bị thiếu (`NaN`), có ba phương pháp phổ biến để điền khuyết:

- **Mode (Giá trị phổ biến nhất)**: Được sử dụng cho dữ liệu phân loại (categorical data). Mode là giá trị xuất hiện nhiều nhất trong cột dữ liệu. Phương pháp này hữu ích khi cột có một số lượng nhỏ giá trị duy nhất, chẳng hạn như feature `Drivetrain, Seating Capacity`



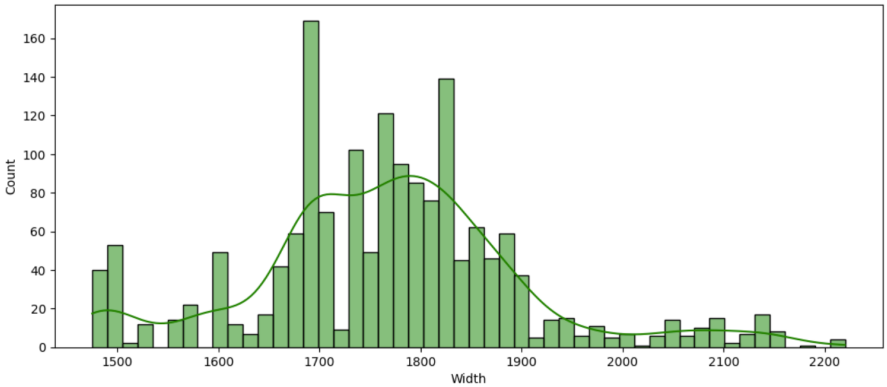
Hình 4: Dạng biểu đồ cần dùng mode để điền khuyết

- **Median (Trung vị)**: Được sử dụng cho dữ liệu số nhưng có phân phối lệch (skewed data). Median là giá trị trung gian khi sắp xếp dữ liệu theo thứ tự tăng dần. Nó ít bị ảnh hưởng bởi các giá trị ngoại lai (outliers) nên phù hợp với dữ liệu như dung tích động cơ `Engine` hoặc mô-men xoắn `max_torque_nm`



Hình 5: Dạng biểu đồ cần dùng median để điền khuyết

- **Mean (Trung bình)**: Được sử dụng cho dữ liệu số có phân phối chuẩn (normal distribution). Mean là giá trị trung bình cộng của tất cả các điểm dữ liệu trong cột. Phương pháp này phù hợp với các thông số như `Fuel Tank Capacity`, `max_power_bhp`, `max_power_rpm`, `max_torque_rpm`, `Width`, `Height`, `Length`



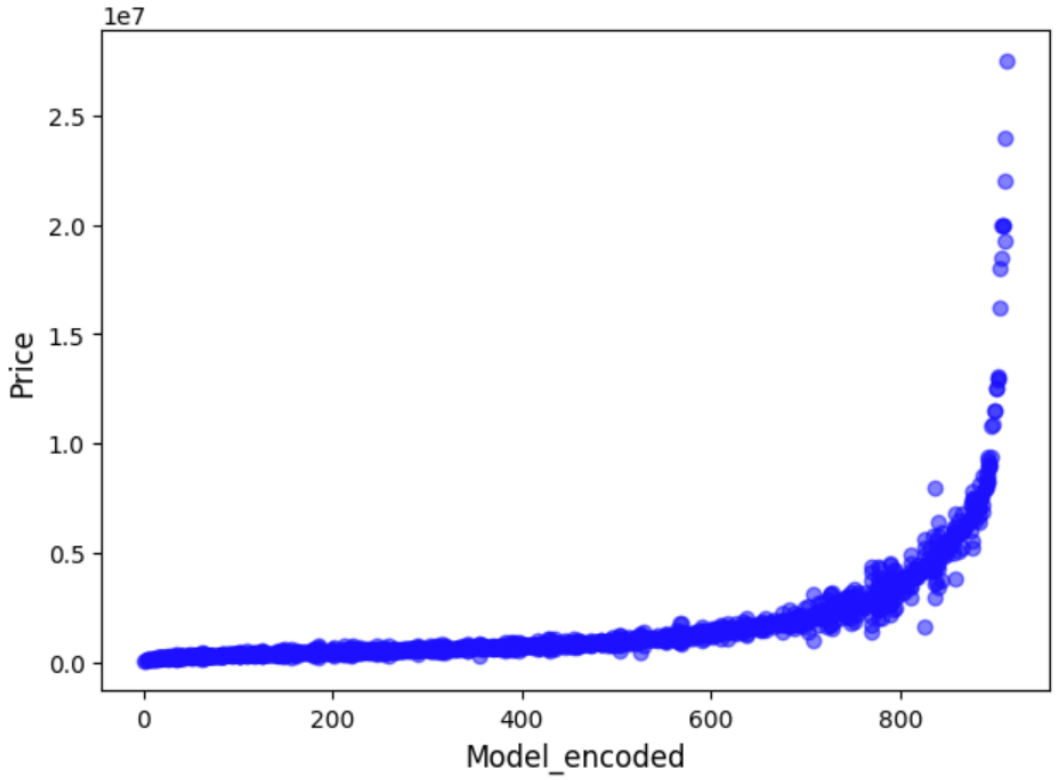
Hình 6: Dạng biểu đồ cần dùng mean để điền khuyết

3.1.3 Data Leakage

Data leakage xảy ra khi mô hình học quá nhiều từ các thông tin có sẵn trong dữ liệu huấn luyện, dẫn đến việc mô hình đạt hiệu suất rất cao trong quá trình huấn luyện, nhưng lại thất bại khi gặp dữ liệu mới (test set) hoặc trong thực tế. Điều này khiến mô hình bị overfitting và không thể tổng quát tốt. Trong trường hợp này, khi nhóm sử dụng mã hóa dựa trên giá trị trung bình (target encoding) cho feature `Model` và tạo ra cột mới là `Model_encoded`, feature này đã gây ra hiện tượng data leakage.

Ban đầu, sự tương quan giữa `Model_encoded` và `Price` chưa qua biến đổi logarithm là 0.6752. Tuy nhiên, sau khi áp dụng phép biến đổi logarithm vào `Price`, hệ số tương quan này tăng vọt lên 0.9637. Quan sát hình 7, ta thấy rằng `Model_encoded` gần như tạo ra một đường cong hoàn hảo mô tả một hàm toán học (chẳng hạn như dạng $a.e^{f(x)}$) để `Model_encoded` khớp với các giá trị của `Price`, điều này rõ ràng chỉ ra rằng feature này chứa thông tin quá mức về target, gây ra data leakage.

Thêm vào đó, có đến 912 giá trị unique trong `Model_encoded`, nhưng dữ liệu huấn luyện chỉ có 1647 mẫu. Khi một feature có quá nhiều giá trị unique trong khi số lượng mẫu huấn luyện lại



Hình 7: Biểu diễn feature `Model_encoded` theo `Price`

không đủ lớn để đại diện cho tất cả các giá trị này, mô hình dễ dàng "học thuộc" các mối quan hệ chi tiết giữa feature và target, dẫn đến overfitting và giảm khả năng tổng quát.

Với mối quan hệ tương quan lớn giữa `Model_encoded` và `Price`, cùng với số lượng giá trị unique lớn nhưng số mẫu huấn luyện lại quá ít, nhóm đã quyết định loại bỏ feature này khỏi mô hình. Mặc dù feature này có thể cải thiện độ chính xác trong quá trình huấn luyện, nhưng sẽ không thể giúp mô hình tổng quát tốt khi áp dụng trên dữ liệu mới, và quan trọng hơn là nó có thể gây ra overfitting nghiêm trọng.

3.1.4 Train Set, Validation Set

Sau khi xử lý dữ liệu, trước khi đưa vào mô hình nhóm sẽ chuẩn hóa Z-score cho tập dữ liệu. Sau đó, nhóm chia dữ liệu trong dữ liệu thành hai tập:

- **Tập huấn luyện (Train set)**: Được sử dụng để **huấn luyện mô hình**, tìm kiếm các tham số tối ưu (hệ số hồi quy).
- **Tập kiểm định (Validation set)**: Được sử dụng để **đánh giá mô hình trong quá trình huấn luyện**, giúp điều chỉnh siêu tham số nhằm tránh overfitting.

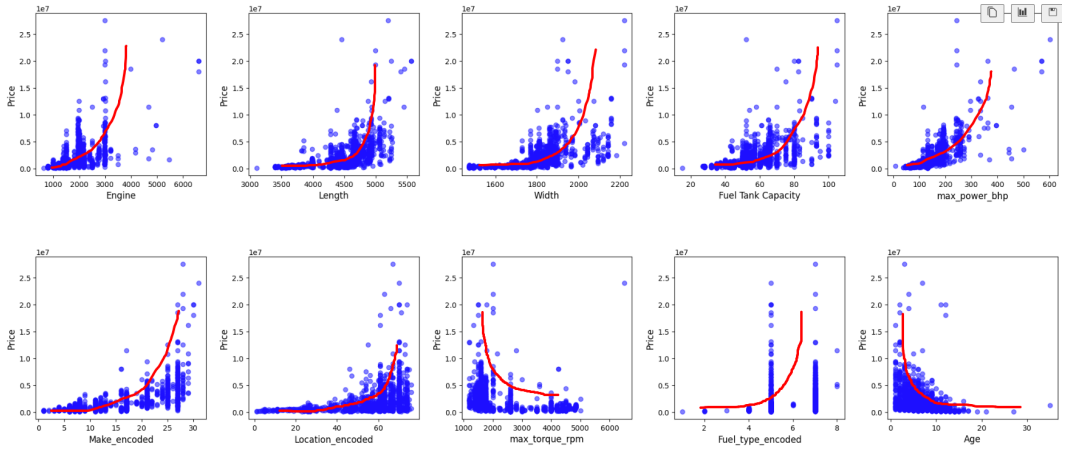
Mục đích tập Validation

Hồi quy tuyến tính có ít siêu tham số hơn các mô hình phức tạp như mạng neurons, nhưng vẫn cần tập validation để:

1. **Đánh giá hiệu suất mô hình** trước khi thử nghiệm trên dữ liệu thực.
2. **Tối ưu hóa siêu tham số**: Chọn đặc trưng tối ưu (Feature Selection) nhờ tập validation giúp kiểm tra xem việc loại bỏ đặc trưng có cải thiện độ chính xác hay không, hoặc khi tăng chiều dữ liệu lên khi sử dụng hồi qui đa thức (polinomial regression) có bị overfit không.

3.2 Mô hình 1

Quan sát



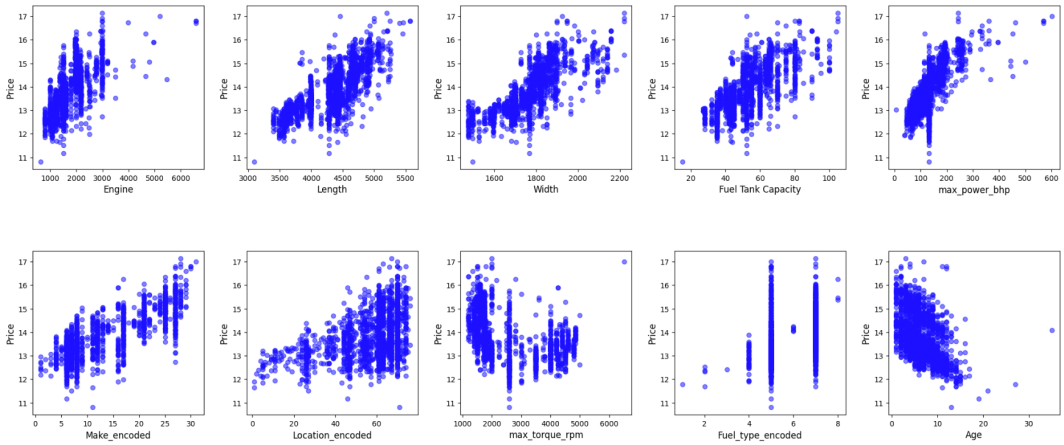
Hình 8: Biểu diễn một số feature theo Price

Quan sát hình 8, các biểu đồ scatter plot thể hiện mối quan hệ giữa `Price` và các đặc trưng như `Engine`, `Length`, `Width` , v.v. Tuy nhiên, dữ liệu có sự phân bố lệch, với giá trị `Price` trải rộng trên một thang đo lớn. Hơn nữa, hầu hết các đặc trưng thể hiện mối quan hệ phi tuyến tính với `Price`, đặc biệt khi giá trị của đặc trưng tăng hoặc giảm mạnh ở các khoảng giá trị cao/thấp. Điều này gợi ý rằng việc áp dụng biến đổi logarithm có thể giúp chuẩn hóa phân phối và làm cho quan hệ giữa các biến trở nên tuyến tính hơn.

Như thể hiện trong bảng 2, với các biến đổi logarithm, mô hình log-linear có thể phù hợp hơn cho bài toán này. Khi giá trị X tăng một đơn vị, giá trị Y sẽ thay đổi theo hệ số e^W, nghĩa là khi các đặc trưng thay đổi lớn hơn, `Price` cũng có xu hướng biến động mạnh hơn. Sau khi áp dụng biến đổi logarithm cho `Price` (hình 9), mối quan hệ giữa $\log(Price)$ và các đặc trưng trở nên tuyến tính hơn, giúp cải thiện khả năng dự đoán của mô hình hồi quy.

Mô hình hồi qui log-linear

$$\log(Price) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta x_{12} + \epsilon \tag{1}$$



Hình 9: Biểu diễn một số đặc trưng theo log(Price)

Với x_i là các đặc trưng được chọn gồm feature `Engine`, `Length`, `Width`, `Fuel_Tank_Capacity`, `max_power_bhp`, `max_torque_rpm`, `Make_encoded`, `Transmission_Automatic`, `Transmission_Manual`, `Drivetrain_AWD`, `Drivetrain_FWD`, `Age`.

Feature Selection

Để lựa chọn đặc trưng, mô hình 1 chỉ sử dụng các feature có giá trị tuyệt đối của **hệ số tương quan** với Price lớn hơn 0.3. Khi tính toán tương quan giữa **Price** (chưa được biến đổi logarithm) với các feature khác, một số các giá trị có hệ số thấp gồm:

- **max_power_rpm**: -0.2117
- **Location_encoded**: 0.2815
- **Fuel_type_encoded**: 0.1961
- **Drivetrain_RWD**: 0.2537

Do các giá trị này đều thấp hơn 0.3, các đặc trưng trên **bị loại bỏ khỏi mô hình**.

Tuy nhiên, sau khi áp dụng **biến đổi logarithm** lên **Price**, hệ số tương quan của các feature này **tăng lên và lớn hơn 0.3**, cho phép chúng được **giữ lại trong mô hình**. Điều này cho thấy rằng việc biến đổi logarithm giúp cải thiện mối quan hệ tương quan giữa các feature và Price, giúp tăng khả năng dự đoán của mô hình.

Ngược lại, một số đặc trưng như `Kilometer, Owner, Height, Seating Capacity, max_torque_nm, Color_encoded, Seller Type` vẫn duy trì độ tương quan thấp ngay cả sau khi thực hiện biến đổi logarithm. Do đó, chúng không được đưa vào mô hình.

Kết quả thực nghiệm

Ta chia tập train và tập validation để đánh giá mô hình (tỉ lệ 80:20) để đánh giá mô hình.

Metric	Model	Train	Val	Total-Train
MSE	Linear	0.3472	0.2139	0.3206
	Log-Linear	0.0733	0.0626	0.0712
R-Squared	Linear	0.6659	0.7453	0.6793
	Log-Linear	0.9245	0.9337	0.9263

Bảng 6: So sánh Mean Squared Error và R-Squared Score cho mô hình linear và log-linear

Đánh giá

Như trình bày trong Bảng 6, bài toán này đánh giá hiệu suất của hai mô hình Linear và Log-Linear dựa trên Mean Squared Error (MSE) và R-Squared Score.

Kết quả thực nghiệm cho thấy cho thấy sử dụng mô hình Log-Linear có thể cải thiện đáng kể độ chính xác của mô hình so với mô hình Linear. Cụ thể, Log-Linear giúp giảm MSE từ 0.3206 xuống 0.0712, tương ứng với mức cải thiện khoảng 77.8%. Điều này có nghĩa là mô hình Log-Linear có khả năng dự đoán tốt hơn và ít sai số hơn.

Đối với hệ số xác định (R-Squared Score), mô hình Log-Linear đạt giá trị 0.9263, cao hơn đáng kể so với 0.6793 của mô hình Linear. Vậy Log-Linear có thể giải thích phương sai của dữ liệu tốt hơn, giúp tăng độ tin cậy của dự đoán.

3.3 Mô hình 2

Mô hình hồi quy tuyến tính được xây dựng như sau:

$$\sqrt{Price} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \beta_{n+1} x_{n+1}^2 + \beta_{n+2} x_{n+2}^2 + \beta_{n+3} x_{n+3} x_{n+4} + \epsilon \tag{2}$$

Trong đó:

- $\sqrt{Price}$ là biến mục tiêu đã được biến đổi để giảm skew và ảnh hưởng của ngoại lai.
- x_i là các biến độc lập đã chọn từ tập dữ liệu ban đầu.
- β_i là các hệ số hồi quy cần được huấn luyện.
- ϵ là sai số ngẫu nhiên.

Mở rộng không gian đặc trưng

- $Engine^2$, Age^2: Hai biến này được thêm vào nhằm mô hình hóa các mối quan hệ phi tuyến giữa đặc trưng với giá xe. Cụ thể:
  - Giá xe không tăng tuyến tính theo dung tích động cơ. Xe có động cơ lớn thường có giá cao hơn, nhưng sau một mức nhất định, mức tăng giá sẽ không còn đáng kể. Việc bình phương `Engine` giúp mô hình hóa hiện tượng này.
  - Tuổi xe thường có quan hệ nghịch biến với giá, nhưng mối quan hệ này không tuyến tính. Xe mới có xu hướng mất giá nhanh hơn trong vài năm đầu, trong khi xe đã cũ thì tốc độ mất giá chậm lại. Việc thêm Age^2 hỗ trợ mô hình trong việc học được xu hướng này.

Biến x_i	**Định nghĩa**
x_1	`Engine` – Dung tích động cơ
x_2	`Length` – Chiều dài xe
x_3	`Width` – Chiều rộng xe
x_4	`Fuel_Tank_Capacity` – Dung tích bình nhiên liệu
x_5	`max_power_bhp` – Công suất cực đại (mã lực)
x_6	`Make_encoded` – Hãng sản xuất (mã hóa)
x_7	`Transmission_Automatic` – Biến nhị phân: xe số tự động
x_8	`Transmission_Manual` – Biến nhị phân: xe số tay
x_9	`Drivetrain_AWD` – Hệ dẫn động 4 bánh toàn thời gian
x_{10}	`Drivetrain_FWD` – Hệ dẫn động cầu trước
x_{11}	`Age` – Tuổi của xe
x_{12}	$Engine^2$ – Bình phương động cơ
x_{13}	Age^2 – Bình phương tuổi xe
x_{14}	`Engine * Length` – Biến tương tác giữa động cơ và chiều dài xe

Bảng 7: Danh sách các biến đầu vào trong mô hình

- `Engine * Length`: Việc kết hợp giữa dung tích động cơ và chiều dài xe tạo ra một đặc trưng tương tác có khả năng dự đoán cao hơn so với từng biến riêng lẻ. Trên thực tế, các xe có động cơ lớn thường đi kèm với thân xe dài (xe SUV, xe sang), và yếu tố này có ảnh hưởng đáng kể đến giá thành.

Xây dựng mô hình hồi quy tuyến tính

- Chuẩn hóa đầu vào bằng chuẩn z-score để đảm bảo các đặc trưng có cùng thang đo.
- Huấn luyện mô hình bằng phương pháp Gradient Descent với số vòng lặp lớn (10000 epochs).
- Đánh giá mô hình bằng các chỉ số:
  - R^2: hệ số xác định.
  - MSE: trung bình bình phương sai số.

Kết quả thực nghiệm

Ta chia tập train và tập validation theo tỉ lệ 80:20 để đánh giá hiệu suất mô hình.

Metric	Train	Val
MSE	0.1378	0.0950
R-Squared	0.8660	0.8928

Bảng 8: Kết quả huấn luyện và đánh giá mô hình hồi quy

Đánh giá

Như trình bày trong Bảng 8, mô hình hồi quy sử dụng biến phụ thuộc là căn bậc hai của `Price` cho kết quả khá tốt.

Cụ thể, Mean Squared Error (MSE) trên tập huấn luyện là 0.1378 và trên tập kiểm tra là 0.0950, cho thấy sai số thấp và khả năng tổng quát hóa tốt. Đồng thời, hệ số xác định R-Squared đạt 0.8660 trên tập train và 0.8928 trên tập validation, thể hiện rằng mô hình có khả năng giải thích tốt phương sai của dữ liệu.

Việc sử dụng căn bậc hai của `Price` giúp làm giảm độ lệch của phân phối dữ liệu, từ đó cải thiện hiệu quả dự đoán và tăng độ ổn định cho mô hình.

3.4 Mô hình 3

Mô hình hồi quy

Mô hình hồi quy tuyến tính được xây dựng như sau:

$$Price = \beta_0+\beta_1(x_1*x_2)+\beta_2x_3+\beta_3x_4\beta_4x_5+\beta_5x_6\beta_6x_7+\beta_7x_8+\beta_8x_9+\beta_9x_10+\beta_10x_11+\beta_11x_12\epsilon \quad (3)$$

Trong đó:

- `Price` là biến phụ thuộc (biến cần dự đoán).
- x_i là các biến độc lập (bao gồm cả biến bậc 1 và biến tương tác).
- β_i là các hệ số hồi quy cần ước lượng.
- ϵ là sai số ngẫu nhiên.

Biến x_i	**Định nghĩa**
$x_1 * x_2$	`max_power_bhp * Make_encoded` Biến tương tác giữa mã lực cực đại và mã hóa hãng sản xuất.
x_3	`Engine` Dung tích động cơ.
x_4	`Fuel_Tank_Capacity` Dung tích bình xăng.
x_5	`Width` Chiều rộng của xe.
x_6	`Length` Chiều dài của xe.
x_7	`Transmission_Automatic` Hộp số tự động.
x_8	`Drivetrain_AWD` Hệ dẫn động 4 bánh toàn thời gian.
x_9	`Age` Tuổi của xe.
x_{10}	`max_torque_rpm` công suất.
x_{11}	`Transmission_Manual` Biến nhị phân: xe số tay.

Biến x_i	Định nghĩa
x_{12}	Drivetrain_FWD .

Quan sát về phương pháp

Mở rộng không gian đặc trưng

- Tạo các biến bậc cao hơn từ các thuộc tính gốc:
  - **Bậc 2**: Bình phương từng biến gốc x_i^2.
  - **Bậc 3**: Lập phương từng biến gốc x_i^3.
- Tạo các biến tương tác bằng cách nhân chéo các thuộc tính với nhau: $x_i x_j$.

Lựa chọn biến đầu vào cho mô hình

- Tính hệ số tương quan giữa từng biến mở rộng với biến mục tiêu.
- Chọn ra các biến có hệ số tương quan cao nhất để đưa vào mô hình, loại bỏ các biến có tương quan thấp hoặc không có ý nghĩa thống kê.

Xây dựng mô hình hồi quy tuyến tính

- Xây dựng mô hình hồi quy với các biến đã chọn.
- Ước lượng hệ số hồi quy bằng phương pháp bình phương nhỏ nhất (OLS).
- Đánh giá chất lượng mô hình bằng các chỉ số như R^2, MSE, MAE rồi từ đó loại bỏ các biến khong hiệu quả.

Kết quả thực nghiệm

Các thuộc tính đang có trong mô hình	MSE	R^2
$x_1 * x_2$	0.2220	0.7357
$x_1 * x_2$, x_3	0.2201	0.7380
$x_1 * x_2$, x_3, x_4	0.2202	0.7379
$x_1 * x_2$, x_3, x_4, x_5	0.2209	0.7370
$x_1 * x_2$, x_3, x_4, x_5, x_6	0.2211	0.7367
$x_1 * x_2$, x_3, x_4, x_5, x_6, x_7	0.2209	0.7370
$x_1 * x_2$, x_3, x_4, x_5, x_6, x_7, x_8	0.2192	0.7391
$x_1 * x_2$, x_3, x_4, x_5, x_6, x_7, x_8, x_9	0.1834	0.7817
$x_1 * x_2$, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}	0.1795	0.7863
$x_1 * x_2$, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}	0.1795	0.7863
$x_1 * x_2$, x_3, x_4, x_5, x_6, , x_7, x_8, x_9, x_{10}, x_{11}, x_{12}	0.1743	0.7925

Bảng 10: Kết quả MSE và R^2 cho mô hình sau khi thêm lần lượt các thuộc tính được chọn

Mô hình tối ưu Theo bảng kết quả thực nghiệm, mô hình tối ưu là mô hình bao gồm tất cả các biến (x_1 đến x_{12}), vì nó có:

- MSE (Sai số bình phương trung bình) thấp nhất: 0,1743
- Giá trị R^2 cao nhất: 0,7925

Ý nghĩa của một số biến quan trọng đến mô hình trong thực tế

1. $x_1 * x_2$ **(Tương tác giữa mã lực và hãng sản xuất):**
   - Đây là biến quan trọng nhất, chỉ riêng biến này đã giải thích được 73,57
   - Trong thực tế, điều này phản ánh rằng mức độ ảnh hưởng của công suất động cơ đến giá xe khác nhau tùy thuộc vào thương hiệu. Ví dụ, tăng 50 mã lực ở một xe Mercedes có thể làm tăng giá nhiều hơn so với tăng 50 mã lực ở một xe Kia
   - Phản ánh sự khác biệt về định vị thương hiệu và phân khúc thị trường
2. x_9 **(Tuổi của xe):**
   - Có tác động rất lớn đến mô hình khi R^2 tăng từ 0,7391 lên 0,7817 khi thêm biến này
   - Trong thực tế, xe càng cũ thì giá trị càng giảm do khấu hao và hao mòn tự nhiên
   - Tốc độ mất giá khác nhau tùy loại xe và phân khúc thị trường
3. x_3 **(Dung tích động cơ):**
   - Là chỉ số kỹ thuật quan trọng ảnh hưởng đến hiệu suất và giá thành xe
   - Động cơ lớn hơn thường đi kèm với công suất cao hơn và chi phí vận hành lớn hơn
   - Trong thực tế, dung tích động cơ còn ảnh hưởng đến mức thuế, phí đăng ký ở nhiều quốc gia
4. x_8 **(Hệ dẫn động 4 bánh toàn thời gian - AWD):**
   - Góp phần cải thiện R^2 lên 0,7391 từ 0,7370
   - Trong thực tế, xe có hệ dẫn động AWD thường đắt hơn vì cung cấp khả năng vận hành tốt hơn trong điều kiện địa hình và thời tiết khắc nghiệt
   - Là tính năng cao cấp được ưa chuộng trong phân khúc SUV và xe thể thao
5. x_{12} **(Hệ dẫn động cầu trước - FWD):**
   - Là biến cuối cùng được thêm vào, giúp cải thiện R^2 từ 0,7863 lên 0,7925
   - Trong thực tế, xe dẫn động cầu trước thường có giá thành thấp hơn so với dẫn động cầu sau hoặc 4 bánh
   - Phổ biến ở các xe phân khúc thấp và trung cấp, thường tiết kiệm nhiên liệu hơn

Đánh giá mô hình

1. **Hệ số xác định (R^2) = 0,7925:**
   - Mô hình giải thích được khoảng 79,25
   - Đây là một giá trị R^2 khá cao, cho thấy mô hình có khả năng dự đoán tốt.
2. **Sai số bình phương trung bình (MSE) = 0,1743:**
   - Giá trị MSE thấp chỉ ra rằng sai số dự đoán trung bình của mô hình thấp.
   - Từ kết quả bảng thực nghiệm, ta thấy MSE giảm dần khi thêm các biến mới, đặc biệt là khi thêm biến Age (x_9).
3. **Đánh giá quá trình xây dựng mô hình:**
   - Các biến được thêm dần vào mô hình theo phương pháp tiến (forward selection).
   - Mỗi biến mới được thêm vào đều được đánh giá về mức độ cải thiện MSE và R^2.
   - Bước nhảy lớn nhất về chất lượng mô hình đến từ việc thêm biến Age (x_9), khi R^2 tăng từ 0,7391 lên 0,7817.

Kết luận

1. **Về mô hình tối ưu:**
   - Mô hình bao gồm tất cả 12 biến là mô hình tối ưu nhất, với $R^2 = 0{,}7925$ và MSE = 0,1743.
   - Sự kết hợp giữa mã lực và thương hiệu là yếu tố quyết định nhất đến giá xe.
2. **Về các biến dự đoán:**
   - Tuổi xe là yếu tố quan trọng thứ hai, phản ánh thực tế về khấu hao và mất giá theo thời gian.
   - Các đặc điểm kỹ thuật như dung tích động cơ, hệ dẫn động và loại hộp số có tác động đáng kể đến giá xe.
   - Các yếu tố vật lý (kích thước) cũng đóng góp vào việc xác định giá xe nhưng ở mức độ thấp hơn.

3.5 Mô hình 4

Công thức hồi quy

Công thức cho mô hình hồi quy:

$$\sqrt[3]{Price} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_8 x_8 + \beta_9(x_9 + x_{10}) + \beta_{10}(x_{11} + x_{12} + x_{13}) + \beta_{11}(x_{14} + x_{15})$$

trong đó:
$x_1, ..., x_8$: lần lượt là `Engine, Make_encoded, Transmission_Automatic, Transmission_Manual, Drivetrain_FWD, Drivetrain_AWD, Age, Kilometer`
$x_9 + x_{10}$: là `Length + Witdth`
$x_{11} + x_{12} + x_{13}$: là `max_power_bhp + Fuel_type_encoded + Fuel Tank Capacity`
$x_{14} + x_{15}$: là `max_power_rpm + max_torque_rpm`

Quan sát

Trước tiên, qua nhiều lần thử thì ta nhận thấy `Price` khi lấy căn bậc 3 cho ra hệ số tương quan so với các feature cao hơn đáng kể, hay làm cho các đồ thị biểu diễn điểm dữ liệu tuyến tính rõ ràng hơn nên ta sẽ chọn mô hình cho $\sqrt[3]{Price}$ thay vì `Price`

Việc ghép cặp `Length` và `Width` giúp tạo ra một thuộc tính mới có hệ số tương quan cao hơn cả hai, và hai chỉ số này trên thực tế cũng khá liên quan nhau.

Việc ghép cặp `max_power_bhp`, `Fuel_type_encoded` và `Fuel Tank Capacity` giúp tạo ra một thuộc tính mới có hệ số tương quan cao hơn. Hơn nữa, nó vừa có hệ số tương quan cao vừa giữ được một tính chất có hệ số tương quan thấp là `Fuel_type_encoded`.

Việc ghép cặp `max_power_rpm` và `max_torque_rpm` tạo ra một thuộc tính có tương tác khá tốt với giá. Trên thực tế, đây là một yếu tố quan trọng ảnh hưởng đến giá cả và có mối quan hệ với nhau về mặt vật lý.

Các thuộc tính còn lại được chọn dựa trên độ tương quan cao. Ngoài ra, thuộc tính có tương quan thấp là `Kilometer` cũng được giữ lại, vì nó vẫn ảnh hưởng đến giá trên thực tế.

Kết quả thực nghiệm

Mô hình hoạt động khá tốt với R^2 dao động trong khoảng $0.89 - 0.91$ và không có dấu hiệu overfit nghiêm trọng.

4 Giới hạn của các mô hình

Sau quá trình xây dựng, đánh giá và so sánh các mô hình hồi quy tuyến tính, nhóm nhận thấy rằng mỗi mô hình đều có những ưu điểm riêng, nhưng đồng thời cũng tồn tại một số giới hạn nhất định. Nhận diện các giới hạn này giúp nhóm có cái nhìn toàn diện hơn về mức độ phù hợp và hiệu quả của từng mô hình, từ đó đề xuất các hướng cải tiến hợp lý.

1. Giới hạn chung

- **Quan hệ phi tuyến tính:** Mô hình tuyến tính đơn thuần khó nắm bắt các quan hệ phi tuyến giữa đặc trưng và giá.
- **Dữ liệu lệch và ngoại lai:** Giá xe phân bố lệch mạnh, gây ảnh hưởng đến độ chính xác nếu không biến đổi hợp lý.
- **Đa cộng tuyến:** Là có sự tương quan mạnh giữa các biến độc lập trong mô hình hồi quy dẫn đến giảm tính chính xác, khó khăn trong việc xác định ảnh hưởng riêng biệt của từng biến độc lập đối với biến phụ thuộc

  Một số đặc trưng có thể bị lặp thông tin, gây nhiễu hoặc làm giảm hiệu năng mô hình.
- **Overfitting:** Việc thêm nhiều biến tương tác hay bậc cao nếu không kiểm soát tốt sẽ khiến mô hình bị quá khớp.
- **Thiếu kỹ thuật regularization(Ridge/Lasso):** Là thêm phạt vào hàm mất mát bằng tổng bình phương hệ số để tránh overfitting, xử lý đa cộng tuyến. Không có mô hình nào áp dụng phương pháp như Ridge hoặc Lasso để kiểm soát overfitting hay chọn biến hiệu quả.

2. Hướng cải tiến chung

- **Áp dụng biến đổi thích hợp:** Sử dụng log, căn bậc 2 hoặc bậc 3 để xử lý phân bố lệch.
- **Tạo feature tương tác hợp lý:** Chọn lọc dựa trên hiểu biết vật lý/thực tế thay vì tạo đại trà.
- **Thử nghiệm mô hình phi tuyến:** Áp dụng các mô hình như Random Forest, XGBoost hoặc Polynomial Regression.
- **Sử dụng kỹ thuật regularization:** Dùng Ridge hoặc Lasso để chọn biến và kiểm soát overfitting.
- **Áp dụng cross-validation:** Là đánh giá độ chính xác và khả năng tổng quát hóa của mô hình trên dữ liệu chưa từng thấy (unseen data)

  Giúp đánh giá độ ổn định và chọn siêu tham số tối ưu cho mô hình.

Bảng 11: Giới hạn và hướng cải tiến cho từng mô hình

Mô hình	**Giới hạn**	**Hướng cải tiến**
Mô hình 1: Log-Linear Regression	- Chỉ dựa vào biến log(Price) nên chỉ cải thiện phần tuyến tính. - Không tận dụng được tương tác hoặc tính phi tuyến. - Bỏ qua một số feature có tương quan thấp nhưng giá trị thực tiễn cao.	- Kết hợp thêm biến tương tác hoặc đa thức bậc 2. - Áp dụng biến đổi log cho cả các đặc trưng đầu vào.
Mô hình 2: $\sqrt{\text{Price}}$ + mở rộng không gian	- Sử dụng nhiều biến đa thức nhưng không đánh giá mức độ đóng góp của từng biến. - Có nguy cơ overfitting nếu không kiểm soát. - Chưa kiểm tra tính tương quan giữa các biến mới.	- Dùng PCA để giảm chiều. - Dùng cross-validation để đánh giá khả năng tổng quát.
Mô hình 3: Tương tác max_power × Make	- Chỉ chọn 2 biến cuối cùng, đơn giản nhưng có thể thiếu sót. - Các biến bị loại chưa xem xét kỹ ảnh hưởng phi tuyến.	- Tạo thêm tương tác có ý nghĩa vật lý (vd: Engine × Width). - Kiểm tra p-value của từng hệ số.
Mô hình 4: $\sqrt[3]{\text{Price}}$ + nhóm biến	- Tạo nhóm biến nhưng cộng lại nên mất ý nghĩa riêng. - Các nhóm có thể nhiễu nếu có biến không quan trọng. - Không kiểm tra tương quan giữa nhóm.	- Thử PCA hoặc tương tác cặp riêng biệt. - Tách các biến trong nhóm để kiểm tra riêng.

5 Kết luận

Qua quá trình nghiên cứu và thực nghiệm, nhóm đã áp dụng hồi quy tuyến tính để mô tả mối quan hệ giữa giá xe cũ và các thuộc tính (features) trong dữ liệu. Tuy nhiên, thực tế cho thấy

giá xe cũ có mối quan hệ phi tuyến với các thuộc tính này thay vì tuyến tính như giả định ban đầu. Để giải quyết vấn đề này, nhóm đã sử dụng kỹ thuật biến đổi đặc trưng để xử lý các mối quan hệ phi tuyến, giúp cải thiện mô hình. Cuối cùng, nhóm đã lựa chọn các thuộc tính ảnh hưởng lớn và áp dụng hồi quy đa thức để dự đoán giá xe cũ một cách chính xác mà không gặp phải hiện tượng overfitting. Kết quả thực nghiệm cho thấy mô hình hồi quy đa thức dựa vào các biến đổi đặc trưng tối ưu hơn trong việc dự đoán giá xe cũ so với hồi quy tuyến tính đơn giản.

Tài liệu

[1] S. Rajpal, "Over-fitting in Polynomial Regression," [Online]. Available: https://shonit2096.medium.com/over-fitting-in-polynomial-regression-ee67c2113344.

[2] J. Brownlee, "Polynomial Features Transforms for Machine Learning," [Online]. Available: https://machinelearningmastery.com/polynomial-features-transforms-for-machine-learning/.

[3] AlmaBetter, "3 Best Ways to Handle Right-Skewed Data," [Online]. Available: https://www.almabetter.com/bytes/articles/3-best-ways-to-handle-right-skewed-data.

[4] F. Abdulazeez, "Essential Regression Evaluation Metrics: MSE, RMSE, MAE, R^2 and Adjusted R^2," [Online]. Available: https://farshadabdulazeez.medium.com/essential-regression-evaluation-metrics-mse-rmse-mae-rÛé-and-adjusted-rÛé-0600daa1c03a.

[5] K. Benoit, "Linear Regression Models with Logarithmic Transformations," [Online]. Available: https://kenbenoit.net/assets/courses/me104/logmodels2.pdf. [Accessed: 07-April-2025].