**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

# BÁO CÁO ĐỒ ÁN CUỐI KỲ

# MÔN: HOẠCH ĐỊNH NGUỒN LỰC DOANH NGHIỆP

*HỌC KỲ I - NĂM HỌC 2022 - 2023*

ĐỀ TÀI:

## DỰ ĐOÁN GIÁ Ô TÔ ĐÃ QUA SỬ DỤNG BẰNG THUẬT TOÁN HỒI QUY TUYẾN TÍNH VỚI NGÔN NGỮ PYTHON

Giảng viên: Đỗ Duy Thanh

Lớp: IS336.N11.HTCL

# PHẦN 1: THÔNG TIN NHÓM

## 1, Thành viên

| Họ và tên | MSSV | Lớp |
|---|---|---|
| Trương Mỹ Song Dân | 20520424 | IS336.N11.HTCL |
| Đặng Trần Tuấn Anh | 20521058 | IS336.N11.HTCL |
| Phạm Thanh Nhựt | 20521728 | IS336.N11.HTCLHTCL |

## 2. Phân công nhiệm vụ

| Họ và tên | Nhiệm vụ | Mức độ hoàn thành |
|---|---|---|
| Trương Mỹ Song Dân | - Thuyết trình<br>- Viết code | 100% |
| Đặng Trần Tuấn Anh | - Tìm dữ liệu cho các thuật toán<br>- Làm powerpoint | 100% |
| Phạm Thanh Nhựt | - Tìm, chọn ý tưởng cho đề tài<br>- Viết báo cáo | 100% |

# PHẦN 2: THÔNG TIN ĐỒ ÁN

## 1. Giới thiệu chung

Ngành kinh doanh đang phát triển và tìm cách sử dụng máy học vì nó tạo ra các dịch vụ hiệu quả nhằm gia tăng giá trị của việc mua bán hàng hóa trong kinh doanh.

Dự đoán giá ô tô là hành động dự đoán giá dựa trên dữ liệu lịch sử. Sử dụng dữ liệu lịch sử trong máy học để nhận ra xu hướng và hiểu hơn về xu hướng thị trường hiện tại. Máy học tự động hóa dự đoán bằng cách sử dụng các mô hình thống kê để rút ra thông tin chi tiết và đưa ra dự đoán. Máy học có thể thu thập và kiểm tra một lượng lớn dữ liệu lẫn cấu trúc và không có cấu trúc. Nó có thể áp dụng các thuật toán phù hợp, biến đổi, tìm kiếm mẫu và đưa ra quyết định dựa trên dữ liệu mới.

Do sự gia tăng và tầm quan trọng của máy học trong ngành công nghiệp, việc sử dụng dự đoán rất có tiềm năng trong các mô hình kinh doanh hiện nay.

Trong đồ án này, mô hình Thuật toán Hồi quy tuyến tính được sử dụng để dự đoán xe đã qua sử dụng

Mục tiêu của đồ án là thu thập dữ liệu, xử lý dữ liệu và xây dựng thuật toán để dự đoán.

## 2. Giới thiệu về Thuật toán Hồi quy tuyến tính

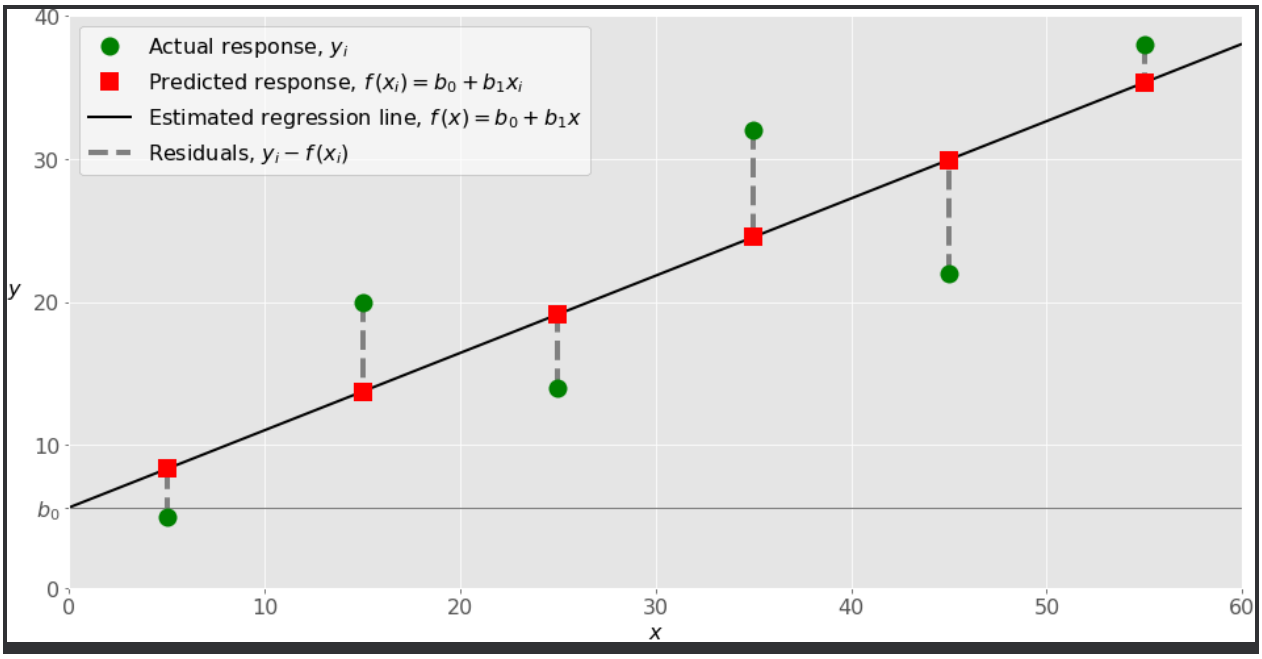
Hồi quy tuyến tính là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị 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…

### *2.1 Hồi quy tuyến tính đơn biến*

Hồi quy tuyến tính đơn biến là trường hợp đơn giản nhất của hồi quy tuyến tính, vì nó có một biến độc lập duy nhất, $\mathbf{x} = x$.

Hình dưới đây minh họa hồi quy tuyến tính đơn biến:




Khi triển khai hồi quy tuyến tính đơn giản, ta thường bắt đầu với một tập hợp các cặp đầu vào-đầu ra ($x$-$y$) nhất định. Các cặp này là được hiển thị dưới dạng các vòng tròn màu xanh lá cây trong hình. Ví dụ: cặp ngoài cùng bên trái có đầu vào $x = 5$ và đầu ra thực tế là $y = 5$. Cặp tiếp theo có $x = 15$ và $y = 20$, v.v.

Hàm hồi quy ước tính, được biểu thị bằng đường màu đen, có phương trình $f(x) = b_0 + b_1x$. Mục tiêu của ta là tính toán các giá trị tối ưu của trọng số dự đoán $b_0$ và $b_1$ để giảm thiểu SSR và xác định hàm hồi quy ước tính.

Giá trị của $b_0$, còn được gọi là giao điểm chặn, cho biết điểm tại đó đường hồi quy ước tính cắt qua trục $y$. Đó là giá trị của phản hồi ước tính $f(x)$ cho $x = 0$. Giá trị của $b_1$ xác định độ dốc của đường hồi quy ước tính.

Các câu trả lời dự đoán, được hiển thị dưới dạng hình vuông màu đỏ, là các điểm trên đường hồi quy tương ứng với các giá trị đầu vào. Ví dụ: đối với đầu vào $x = 5$ được phản hồi dự đoán là $f(5) = 8,33$, mà hình vuông màu đỏ ngoài cùng bên trái đại diện.

Các đường màu xám nét đứt dọc biểu thị phần dư, có thể được tính như $y_i - f(\mathbf{x}_i) = y_i - b_0 - b_1 x_i$ cho $i = 1, \ldots, n$. Chúng là khoảng cách giữa các hình tròn màu xanh lá cây và hình vuông màu đỏ. Khi thực hiện hồi quy tuyến tính, thực tế là ta đang cố gắng giảm thiểu những khoảng cách này và làm cho hình vuông màu đỏ càng gần với hình tròn màu xanh lá cây được xác định trước càng tốt.

### *2.2 Hồi quy tuyến tính đa biến*

Hồi quy tuyến tính đa biến là trường hợp hồi quy tuyến tính có hai biến độc lập trở lên.

Nếu chỉ có hai biến độc lập thì hàm hồi quy ước tính là $f(x_1, x_2) = b_0 + b_1 x_1 + b_2 x_2$. Nó đại diện cho một mặt phẳng hồi quy trong không gian ba chiều. Mục tiêu của hồi quy là xác định giá trị của các trọng số $b_0$, $b_1$ và $b_2$ sao cho mặt phẳng này càng gần với phản hồi thực tế càng tốt, đồng thời mang lại SSR tối thiểu.

Trường hợp có nhiều hơn hai biến độc lập cũng tương tự nhưng tổng quát hơn. Hàm hồi quy ước tính là $f(x_1, \ldots, x_r) = b_0 + b_1 x_1 + \cdots + b_r x_r$, và có $r + 1$ trọng số được xác định khi số lượng đầu vào là $r$.

### *2.3 Hiệu suất của mô hình*

Một khi ta xây dựng mô hình, câu hỏi tiếp theo đến trong đầu là để biết liệu mô hình của ta có đủ để dự đoán trong tương lai hoặc là mối quan hệ mà bạn đã xây dựng giữa các biến phụ thuộc và độc lập là đủ hay không.

Vì mục đích này có nhiều chỉ số mà chúng ta cần tham khảo R - Square (R^2)

Công thức tính R^2 sẽ bằng : R^2 = $\frac{TSS - RSS}{TSS}$

- Tổng các diện tích (TSS): TSS là một phép đo tổng biến thiên trong tỷ lệ đáp ứng / biến phụ thuộc YY và có thể được coi là số lượng biến thiên vốn có trong đáp ứng trước khi hồi quy được thực hiện.
- Sum of Squares (RSS): RSS đo lường lượng biến đổi còn lại không giải thích được sau khi thực hiện hồi quy.
- (TSS - RSS) đo lường mức độ thay đổi trong đáp ứng được giải thích (hoặc loại bỏ) bằng cách thực hiện hồi quy

Mean Absolute Error (MAE)

MAE là thước đo lỗi giữa các quan sát được ghép nối biểu thị cùng một hiện tượng. Các ví dụ về Y so với X bao gồm so sánh dự đoán so với quan sát, thời gian tiếp theo so với thời gian ban đầu và một kỹ thuật đo lường so với kỹ thuật đo lường thay thế.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

Mean Squared Error (MSE)

MSE của một phép ước lượng là trung bình của bình phương các sai số, tức là sự khác biệt giữa các ước lượng và những gì được đánh giá. MSE 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. Sự khác biệt xảy ra do ngẫu nhiên, hoặc vì các ước lượng không tính đến thông tin có thể cho ra một ước tính chính xác hơn

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}( Y_i - \hat{Y}_i)^2$$

Root Mean Square Error (RMSE)

RMSE cho biết mức độ phân tán các giá trị dự đoán từ các giá trị thực tế. Công thức tính RMSE là:

$$R^2 = \{ ( 1 / N ) * \Sigma [ (x_i - mean(x)) * (y_i - mean(y)) ] / (\sigma_x * \sigma_y ) \}^2$$

Mặc dù RMSE là một đánh giá tốt cho các sai số nhưng vấn đề là nó rất dễ bị ảnh hưởng bởi phạm vi của biến phụ thuộc của ta. Nếu biến phụ thuộc của ta có dải biến thiên hẹp, RMSE của ta sẽ thấp và nếu biến phụ thuộc có phạm vi rộng RMSE sẽ cao. Do đó, RMSE là một số liệu tốt để so sánh giữa các lần lặp lại khác nhau của mô hình

### *2.3 Underfitting và Overfitting*

Một câu hỏi rất quan trọng có thể nảy sinh khi ta thực hiện hồi quy đa thức có liên quan đến việc lựa chọn bậc tối ưu của hàm hồi quy đa thức.

Không có quy tắc đơn giản để làm điều này. Nó phụ thuộc tùy vào trường hợp. Lưu ý hai vấn đề có thể xảy ra sau khi lựa chọn mức độ: trang bị thiếu và trang bị thừa.

Việc trang bị thiếu xảy ra khi một mô hình không thể nắm bắt chính xác sự phụ thuộc giữa các dữ liệu, thường là do tính đơn giản của chính nó. Nó thường mang lại $R^2$ thấp với dữ liệu đã biết và khả năng khái quát hóa kém khi áp dụng với dữ liệu mới.

Trang bị thừa xảy ra khi một mô hình học cả phụ thuộc dữ liệu và dao động ngẫu nhiên. Nói cách khác, một mô hình học dữ liệu hiện có quá tốt. Các mô hình phức tạp, có nhiều tính năng hoặc thuật ngữ, thường dễ bị quá khớp. Khi áp dụng cho dữ liệu đã biết, các mô hình như vậy thường mang lại $R^2$ cao. Tuy nhiên, chúng thường không khái quát hóa tốt và có $R^2$ thấp hơn đáng kể khi được sử dụng với dữ liệu mới.

## 3. Thực hiện

### *3.1 Cách thực hiện*

Chúng tôi sử dụng dữ liệu lịch sử về giá xe ô tô từ năm 2003 đến năm 2018 tại Kaggle và áp dụng các thuật toán để dự đoán giá ô tô. Hơn nữa, sử dụng xác thực chéo để tìm kiếm ngẫu nhiên, điều chỉnh và huấn luyện cho dự đoán. Sau khi dự đoán, phân tích sai số rất quan trọng để xác định cách thức hoạt động của mô hình và mức độ chính xác của các giá trị dự đoán.

### *3.2 Các bước áp dụng*

Thông tin cơ bản: Nguồn dữ liệu được lấy từ Kaggle

Bước 1: Import thư viện

```
[1]  1 #import libaries
     2 import pandas as pd
     3 import numpy as np
     4 import matplotlib.pyplot as plt
     5 import seaborn as sns
     6 from sklearn.model_selection import train_test_split
     7 from sklearn.linear_model import LinearRegression
     8 from sklearn import metrics
```

Bước 2: Import data và đọc file

```
[2]  1 from google.colab import drive
     2 drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
1 #import and read data
2 path = "/content/drive/MyDrive/csv file/car data.csv"
3 data = pd.read_csv(path)
4 data.head()
```

| | Car_Name | Year | Selling_Price | Present_Price | Kms_Driven | Fuel_Type | Seller_Type | Transmission | Owner |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ritz | 2014 | 3.35 | 5.59 | 27000 | Petrol | Dealer | Manual | 0 |
| 1 | sx4 | 2013 | 4.75 | 9.54 | 43000 | Diesel | Dealer | Manual | 0 |
| 2 | ciaz | 2017 | 7.25 | 9.85 | 6900 | Petrol | Dealer | Manual | 0 |
| 3 | wagon r | 2011 | 2.85 | 4.15 | 5200 | Petrol | Dealer | Manual | 0 |
| 4 | swift | 2014 | 4.60 | 6.87 | 42450 | Diesel | Dealer | Manual | 0 |

```
1 data.tail()
```

| | Car_Name | Year | Selling_Price | Present_Price | Kms_Driven | Fuel_Type | Seller_Type | Transmission | Owner |
|---|---|---|---|---|---|---|---|---|---|
| 296 | city | 2016 | 9.50 | 11.6 | 33988 | Diesel | Dealer | Manual | 0 |
| 297 | brio | 2015 | 4.00 | 5.9 | 60000 | Petrol | Dealer | Manual | 0 |
| 298 | city | 2009 | 3.35 | 11.0 | 87934 | Petrol | Dealer | Manual | 0 |
| 299 | city | 2017 | 11.50 | 12.5 | 9000 | Diesel | Dealer | Manual | 0 |
| 300 | brio | 2016 | 5.30 | 5.9 | 5464 | Petrol | Dealer | Manual | 0 |

Thông tin về dữ liệu:

```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Car_Name       301 non-null    object
 1   Year           301 non-null    int64
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Kms_Driven     301 non-null    int64
 5   Fuel_Type      301 non-null    object
 6   Seller_Type    301 non-null    object
 7   Transmission   301 non-null    object
 8   Owner          301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

Don't have missing value

```
[5]  1 #columns and rows of data
     2 data.shape
```

```
(301, 9)
```

Bước 3: Số hóa dữ liệu

```
[7]  1 #encoding Fuel_Type, Seller_Type, Transmission
     2 data.replace({'Fuel_Type':{'Petrol':0, 'Diesel':1, 'CNG': 2}}, inplace = True)
     3 data.replace({'Seller_Type':{'Dealer':0, 'Individual':1}}, inplace = True)
     4 data.replace({'Transmission':{'Manual':0, 'Automatic':1}}, inplace = True)
```

```
1 data.head()
```

| | Car_Name | Year | Selling_Price | Present_Price | Kms_Driven | Fuel_Type | Seller_Type | Transmission | Owner |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ritz | 2014 | 3.35 | 5.59 | 27000 | 0 | 0 | 0 | 0 |
| 1 | sx4 | 2013 | 4.75 | 9.54 | 43000 | 1 | 0 | 0 | 0 |
| 2 | ciaz | 2017 | 7.25 | 9.85 | 6900 | 0 | 0 | 0 | 0 |
| 3 | wagon r | 2011 | 2.85 | 4.15 | 5200 | 0 | 0 | 0 | 0 |
| 4 | swift | 2014 | 4.60 | 6.87 | 42450 | 1 | 0 | 0 | 0 |

Bước 4: Tách mẫu X và Y

```
1 #Split data
2 X = data.drop(['Car_Name','Selling_Price'], axis = 1)
3 Y= data['Selling_Price']
```

```
[10]  1 X
```

| | Year | Present_Price | Kms_Driven | Fuel_Type | Seller_Type | Transmission | Owner |
|---|---|---|---|---|---|---|---|
| 0 | 2014 | 5.59 | 27000 | 0 | 0 | 0 | 0 |
| 1 | 2013 | 9.54 | 43000 | 1 | 0 | 0 | 0 |
| 2 | 2017 | 9.85 | 6900 | 0 | 0 | 0 | 0 |
| 3 | 2011 | 4.15 | 5200 | 0 | 0 | 0 | 0 |
| 4 | 2014 | 6.87 | 42450 | 1 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 296 | 2016 | 11.60 | 33988 | 1 | 0 | 0 | 0 |
| 297 | 2015 | 5.90 | 60000 | 0 | 0 | 0 | 0 |
| 298 | 2009 | 11.00 | 87934 | 0 | 0 | 0 | 0 |
| 299 | 2017 | 12.50 | 9000 | 1 | 0 | 0 | 0 |
| 300 | 2016 | 5.90 | 5464 | 0 | 0 | 0 | 0 |

301 rows × 7 columns

```
1 Y
```

```
0       3.35
1       4.75
2       7.25
3       2.85
4       4.60
       ...
296     9.50
297     4.00
298     3.35
299    11.50
300     5.30
Name: Selling_Price, Length: 301, dtype: float64
```

Bước 5: Chia dữ liệu Training và Testing

```
[12]  1 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state = 2)
```

Bước 6: Đọc dữ liệu vào mô hình hồi quy tuyến tính

```
[13]  1 #loading the linear regression model
      2 lin_reg_model = LinearRegression()
```

Bước 7: Huấn luyện dữ liệu với X_train

```
[14]  1 #training the model with X_train
      2 lin_reg_model.fit(X_train,Y_train)

LinearRegression()
```

Bước 8: Dự đoán dữ liệu Training

```
[15]  1 #Predict on Training Data
      2 training_data_predict = lin_reg_model.predict(X_train)

[16]  1 #R squared error
      2 error_score = metrics.r2_score(Y_train, training_data_predict)

[17]  1 "R Squared error:", error_score

('R Squared error:', 0.8799451660493711)
```

Bước 9: Thể hiện sơ đồ trực quan dữ liệu Training

```
#Visualize the actual prices and predict prices
plt.scatter(Y_train, training_data_predict)
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title("Actual Prices vs Predicted Prices")
plt.show()
```


Bước 10: Dự đoán dữ liệu Testing

```
#Predict on Testing Data
test_data_predict = lin_reg_model.predict(X_test)
```

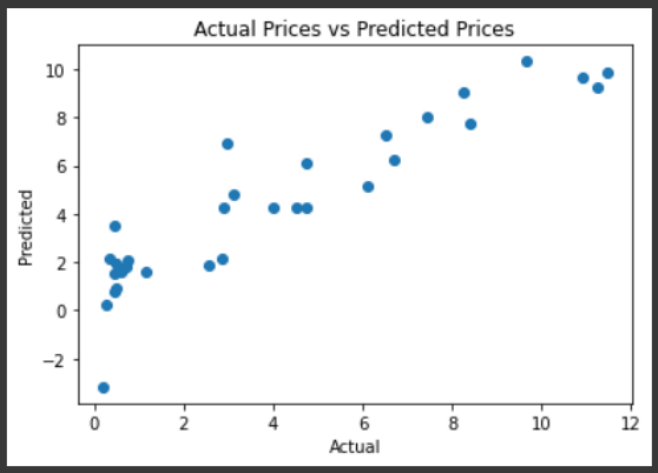
```
[20] #R squared error
error_score = metrics.r2_score(Y_test, test_data_predict)
```

```
[32] "R Squared error:", error_score
```

```
('R Squared error:', 0.8365766715027051)
```

Bước 11: Thể hiện sơ đồ trực quan dữ liệu Testing

```
[37] 1 #Visualize the actual prices and predict prices
     2 plt.scatter(Y_test, test_data_predict)
     3 plt.xlabel("Actual")
     4 plt.ylabel("Predicted")
     5 plt.title("Actual Prices vs Predicted Prices")
     6 plt.show()
```




*3.3 Chú thích*

- Các thư viện được sử dụng:
    1. **pandas** - tải tệp dữ liệu dưới dạng khung dữ liệu pandas để phân tích dữ liệu
    2. **matplotlib** – vẽ đồ thị
    3. **sklearn** – hỗ trợ các mô hình máy học, xử lý, đánh giá mô hình và đào tạo
    4. **numpy** - thực hiện với mảng
    5. **seaborn** - tạo ra các hình ảnh trực quan đẹp mắt
- Chú thích các dữ liệu:

    **Car_Name**: tên các loại xe
    **Year**: Năm xe được mua
    **Selling_Price**: Giá chủ sở hữu mong muốn bán
    **Present_Price**: Giá xuất xưởng hiện tại của xe
    **Kms_Driven**: Quãng đường ôtô đã hoàn thành được tính bằng km
    **Fuel_Type**: Loại nhiên liệu của xe
    **Seller_Type**: Xác định xem người bán là đại lý hay cá nhân
    **Transmission**: Xác định xem là xe số sàn hay số tự động
    **Owner**: Số lượng chủ sở hữu đã sở hữu xe trước đó

*3.4 Các chỉ số thống kê và báo cáo hiệu suất*

- Các chỉ số thống kê là các thước đo sai số trong cho phép hồi quy nhưng được sử dụng để dự đoán rủi ro. Đánh giá mô hình rất quan trọng và cần được đánh giá để giảm rủi ro và tăng hiệu suất mô hình
- RMSE là thước đo độ lệch chuẩn của các phần dư (sai số dự đoán). Phần dư được định nghĩa là khoảng cách các điểm dữ liệu so với đường hồi quy. RMSE là thước đo để đánh giá sự phát tán của các phần dư này. Nói cách khác, nó cho biết dữ liệu được tập trung như thế nào bằng cách phù hợp nhất. Ngoài ra nó là căn bậc hai của MSE. Giá trị RMSE càng thấp, hiệu suất càng tốt vì nó đo được nhiều sai số hơn các thước đo sai số khác. Mô hình sẽ dự đoán chính xác hơn khi RMSE có giá trị nhỏ hơn 0.5 và lớn hơn 0.3
- MAE được làm thước đo để đo độ lớn trung bình của các sai số trong tập hợp các dự đoán mà không cần xem xét hướng của chúng. Đó là sự khác biệt tuyệt đối trung bình giữa giá trị dự đoán và giá trị thực khi các khoản riêng lẻ có trọng số bằng nhau. Đáng chú ý nhất, MAE có thể đo được giá trị thực và giá trị được dự đoán. Tuy nhiên, MAE không đánh mạnh vào sai số trong dự đoán vì vậy nếu các sai số được xem xét, nó phải là sai số trung bình phương trung bình hoặc sai số bình phương trung bình gốc. Giá trị càng thấp càng tốt.
- MSE lấy tổng giá trị tuyệt đối của sai số. Nó cũng xác định được hiệu suất của mô hình. Trong trường hợp này, các sai số lớn hơn được ghi chú lại nhiều hơn so với MAE. Giá trị MSE càng thấp, độ chính xác càng cao
- R-squared cho biết mức độ phù hợp với dữ liệu được cho. Nó cho biết mức độ gần với đường hồi quy nghĩa là các giá trị thực được hiển thị trên đồ thị. Giá trị lớn nhất là 1.0 và giá trị càng cao, mô hình càng phù hợp khi R-squared có giá trị nằm giữa 0.6-1.0. Giá trị trên 80% được coi là tốt.

- Đánh giá hiệu suất trong máy học rất quan trọng để biết được rõ hơn về tính dự đoán và mô hình đang thực hiện. Trong đề tài này, R-squared được sử dụng để đánh giá mô hình. Giá trị đầu ra của đánh giá mô hình sẽ xác định xem mô hình có nên cải thiện hay không.

*3.55 Ưu, nhược điểm*

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 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:

- Nhạy cảm với dữ liệu nhiễu
- Không biểu diễn được những mô hình phức tạp
- Biến độc lập phải độc lập với nhau nhưng trong thực tế rất hiểm
- Đối với dữ liệu phi tuyến tính, hồi quy đa thức có thể khá khó khăn để thiết kế vì ta phải có một số thông tin về cấu trúc của dữ liệu và mối quan hệ giữa các biến tính năng

*3.66 Kết luận, ý kiến và quan điểm của nhóm*

Việc dự đoán số liệu phụ thuộc rất lớn về kinh tế và đánh giá của các chuyên gia nên việc dự đoán không có độ chính xác cao. Tuy nhiên, việc sử dụng thuật toán cũng giúp ích phần nào cho việc phát triển mô hình này một cách nhanh chóng. Nhìn chung, thuật toán Hồi quy tuyến tính còn hạn chế nhưng vẫn rất nhanh, đơn giản và linh hoạt.