

Mini Project

DỰ ĐOÁN GIÁ BÁN XE Ô TÔ



Giới thiệu

Phân tích dữ liệu

Cài đặt mô hình

Kết luận và hướng phát triển

- **Mô tả đề tài:** sử dụng Linear Regression... và sử dụng các dữ liệu gồm 8 thuộc tính của xe: tên xe, số năm sử dụng, giá bán hiện tại trên thị trường, số kilomet đã đi, loại dầu/xăng, mô hình kinh doanh, xe số/xe tự động, chủ sở hữu xe... để dự đoán giá bán xe

Bộ dữ liệu sử dụng: lấy từ tập dataset Kaggle: [Vehicle Dataset](#)

- Car_Name: tên của xe
- Year: năm xe được mua
- Present_Price: Giá hiện tại của xe trên thị trường
- Kms_Driven: Quãng đường xe đã đi
- Fuel_Type: Loại xăng/dầu mà xe sử dụng
- Seller_Type: Mô hình kinh doanh (cá nhân/tổ chức)
- Transmission: Động cơ xe
- Owner: Số lượng người từng sở hữu xe
- Selling_Price: Giá bán

- Thực hiện một số khảo sát về dữ liệu
 - Thống kê dữ liệu
 - Phân tích dữ liệu
 - Xử lý dữ liệu
- ...

Tập dữ liệu chứa 301 hàng bản ghi và 9 cột thuộc tính.

Các loại dữ liệu của các thuộc tính bao gồm 3 thuộc tính kiểu số nguyên rời rạc, 2 thuộc tính số thực liên tục và 4 thuộc tính kiểu đối tượng

Sử dụng không gian bộ nhớ ít nhất là 21.3 kilobyte (KB).

```
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
```

```
data.head(10)
```

	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
5	vitara brezza	2018	9.25	9.83	2071	Diesel	Dealer	Manual	0
6	ciaz	2015	6.75	8.12	18796	Petrol	Dealer	Manual	0
7	s cross	2015	6.50	8.61	33429	Diesel	Dealer	Manual	0
8	ciaz	2016	8.75	8.89	20273	Diesel	Dealer	Manual	0
9	ciaz	2015	7.45	8.92	42367	Diesel	Dealer	Manual	0

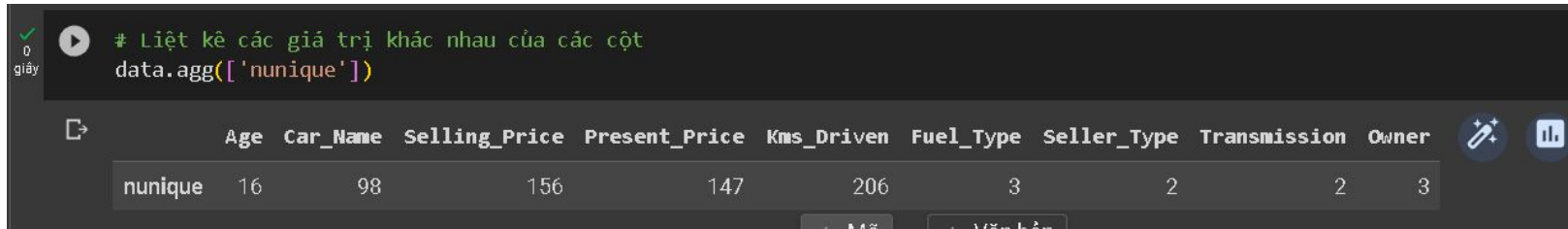
Hình ảnh minh họa dữ liệu trong tập dữ liệu

```
0 giây [▶] data.isnull().sum()

Car_Name      0
Year          0
Selling_Price 0
Present_Price 0
Kms_Driven    0
Fuel_Type     0
Seller_Type   0
Transmission  0
Owner         0
dtype: int64
```

Số lượng giá trị Null trong các thuộc tính của tập dữ liệu

Phân tích dữ liệu

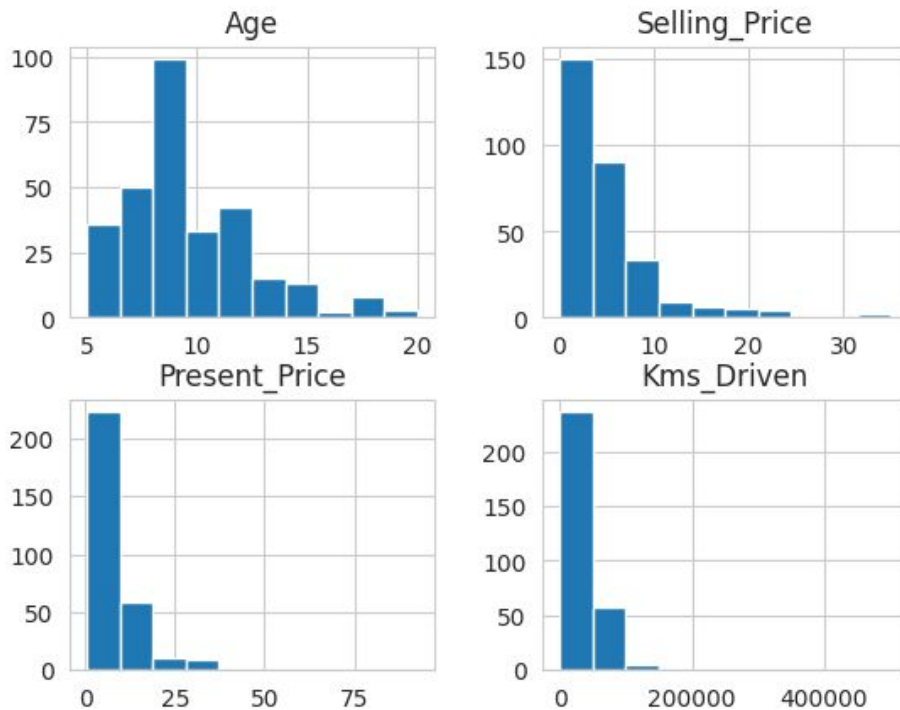


The screenshot shows a Jupyter Notebook interface. The code cell contains a comment in Vietnamese and a pandas command to calculate the number of unique values for each column. The output is a table with 10 columns: Age, Car_Name, Selling_Price, Present_Price, Kms_Driven, Fuel_Type, Seller_Type, Transmission, Owner, and a 'nunique' column. The 'nunique' column contains the cardinality for each of the other columns.

```
# Liệt kê các giá trị khác nhau của các cột  
data.agg(['nunique'])
```

	Age	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
nunique	16	98	156	147	206	3	2	2	3

Số giá trị khác nhau (cardinality) của các thuộc tính



Biểu đồ histogram

- Bổ sung cột “Age” là số năm xe được sử dụng

```
# convert Year to Age of each cars  
Age = 2023 - data.Year  
  
data.insert(0, "Age", Age)  
data.drop('Year', axis = 1, inplace = True)  
data
```

	Age	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	9	ritz	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	10	sx4	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	6	ciaz	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	12	wagon r	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	9	swift	4.60	6.87	42450	Diesel	Dealer	Manual	0
...
296	7	city	9.50	11.60	33988	Diesel	Dealer	Manual	0
297	8	brio	4.00	5.90	60000	Petrol	Dealer	Manual	0
298	14	city	3.35	11.00	87934	Petrol	Dealer	Manual	0
299	6	city	11.50	12.50	9000	Diesel	Dealer	Manual	0
300	7	brio	5.30	5.90	5464	Petrol	Dealer	Manual	0

301 rows x 9 columns

- Chuyển dữ liệu dạng “String” thành số

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

data
```

	Age	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	9	ritz	3.35	5.59	27000	2	2	2	0
1	10	sx4	4.75	9.54	43000	3	2	2	0
2	6	ciaz	7.25	9.85	6900	2	2	2	0
3	12	wagon r	2.85	4.15	5200	2	2	2	0
4	9	swift	4.60	6.87	42450	3	2	2	0
...
296	7	city	9.50	11.60	33988	3	2	2	0
297	8	brio	4.00	5.90	60000	2	2	2	0
298	14	city	3.35	11.00	87934	2	2	2	0
299	6	city	11.50	12.50	9000	3	2	2	0
300	7	brio	5.30	5.90	5464	2	2	2	0

301 rows x 9 columns

- Bổ sung cột “Age” là số năm xe được sử dụng

```
# convert Year to Age of each cars  
  
Age = 2023 - data.Year  
  
data.insert(0, "Age", Age)  
data.drop('Year', axis = 1, inplace = True)  
data
```

	Age	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	9	ritz	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	10	sx4	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	6	ciaz	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	12	wagon r	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	9	swift	4.60	6.87	42450	Diesel	Dealer	Manual	0
...
296	7	city	9.50	11.60	33988	Diesel	Dealer	Manual	0
297	8	brio	4.00	5.90	60000	Petrol	Dealer	Manual	0
298	14	city	3.35	11.00	87934	Petrol	Dealer	Manual	0
299	6	city	11.50	12.50	9000	Diesel	Dealer	Manual	0
300	7	brio	5.30	5.90	5464	Petrol	Dealer	Manual	0

301 rows x 9 columns

3. Cài đặt mô hình

Bộ dữ liệu bao gồm: 301 hàng

- **80% bộ dữ liệu được sử dụng làm tập train: 240 hàng**
- **20% bộ dữ liệu được sử dụng làm tập test: 61 hàng**

```
✓ [193] X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size = 0.2, random_state = 101)
giây

✓ [192] len(X_train)
giây
      240

✓ [194] len(X_test)
giây
      61
```

3. Mô hình Linear Regression

Độ chính xác: 91.3%

```
✓ [188] from sklearn.linear_model import LinearRegression  
0 clf = LinearRegression()  
giây clf.fit(X_train, Y_train)  
print("Accuracy: ", clf.score(X_test, Y_test))
```

```
Accuracy: 0.9134181721224436
```

3. Mô hình Linear Regression

Độ chính xác: 84 (k = 5) - 87 (k = 10)%

```
#Train the model
model.fit(X_train, y_train) #Training the model
print(f"Accuracy for the fold no. {i} on the test set: {r2_score(y_test, model.predict(X_test))}")
i += 1
```

```
➤ Accuracy for the fold no. 1 on the test set: 0.8348249360765898
Accuracy for the fold no. 2 on the test set: 0.7751200264750783
Accuracy for the fold no. 3 on the test set: -94.6110521346414
Accuracy for the fold no. 4 on the test set: 0.6108862836379172
Accuracy for the fold no. 5 on the test set: 0.8389219725701988
```

```
[190] print("R2-score: %.2f" % r2_score(y_test , model.predict(X_test) ))
```

```
R2-score: 0.84
```


Thank you

