Link Google Colab: Link

Họ tên	MSSV
Nguyễn Võ Ngọc Bảo	23520131
Vũ Việt Cương	23520213
Ngô Phương Nam	23520974

BÁO CÁO BƯỚC 4 - FEATURE ENGINEERING

Môn học: Lập trình Python cho Máy học

1. **Loại bỏ các đặc trưng không có nhiều giá trị dự đoán**
   Ta sẽ xóa bỏ cột *running* và giữ lại cột *running_km* vì cột running không đồng nhất đơn vị nên ta chuyển về chung một đơn vị ở cột *running_km*.
   Bên cạnh đó, ta thấy cột *wheel* chỉ mang 1 giá trị, không phục vụ trong việc dự đoán nên ta cũng bỏ cột *wheel*.
2. **Tạo mới các đặc trưng**
   Ta tạo thêm cột *running_per_year* = *running_km* / ([số năm hiện tại] - *year*).

```
train['run_per_year'] = train['running_km'] / (2025-train['year'])
test['run_per_year'] = test['running_km'] / (2025-test['year'])
print(train[['running_km','year','run_per_year']])
```

```
      running_km    year  run_per_year
0       3000.000  2022.0   1000.000000
1     132000.000  2014.0  12000.000000
2     152887.300  2018.0  21841.042857
3     220479.580  2002.0   9586.068696
4     130000.000  2017.0  16250.000000
...          ...     ...           ...
1637  193120.800  2017.0  24140.100000
1638  170000.000  2014.0  15454.545455
1639  110883.526  2018.0  15840.503714
1640   49889.540  2019.0   8314.923333
1641      20.000  2022.0      6.666667

[1633 rows x 3 columns]
```

Giá trị *running_per_year* cao hơn cho biết xe đã được sử dụng nhiều hơn mỗi năm, điều này có thể cho thấy mức độ hao mòn cao hơn.
Giá trị *running_per_year* thấp hơn cho biết mức độ sử dụng ít hơn, điều này có thể cho thấy mức độ bảo dưỡng tốt hơn và có khả năng giá trị bán lại cao hơn.

→ Đặc trưng này năng này giúp nắm bắt mối quan hệ giữa tuổi của xe và mức độ sử dụng của xe, đây có thể là yếu tố quan trọng trong việc xác định giá của xe.

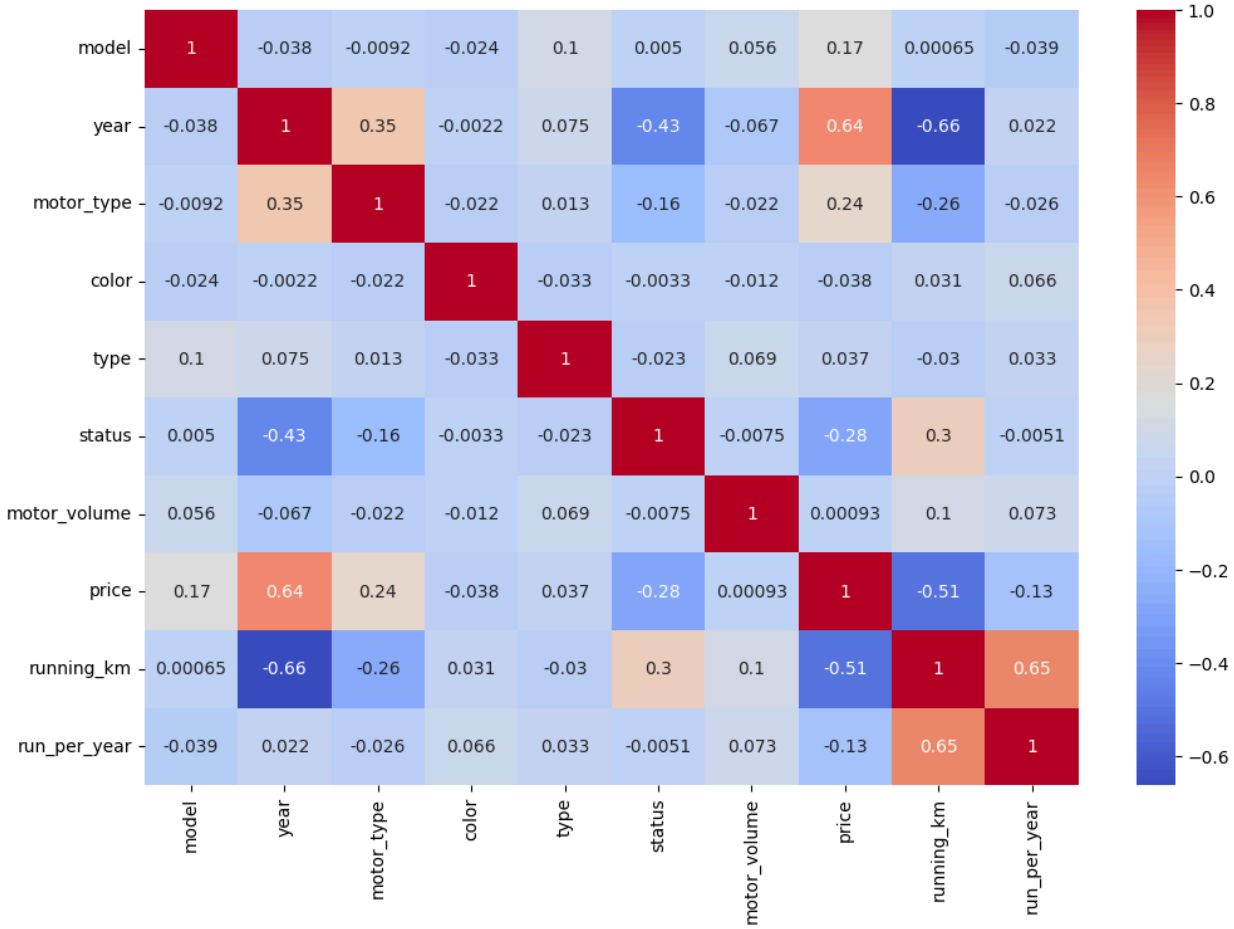
3. **Chọn encoding phù hợp**

   Do sử dụng model dựa trên cây quyết định nên chọn label encoding thay vì one-hot encoding sẽ tạo ra nhiều cột feature hơn.

   Vì ở phần 4.3, ta đã sử dụng LabelEncoder để encode các dữ liệu rồi nên ta bỏ qua phần này.

4. **Chọn lựa đặc trưng**

   Ta áp dụng *matrix_correlation* để chọn các đặc trưng phù hợp cho việc dự đoán.



Có thể thấy, không có đặc trưng nào có độ tương quan cao (> 0.9) để ta có thể loại bỏ.

Ta sẽ sử dụng *model-based method* để biết được các đặc trưng chiếm độ quan trọng bao nhiêu % trong việc dự đoán.

```
[ ]  X = train.drop(columns=['price'], axis=1)
     y = train['price']
```

```
[ ]  from sklearn.ensemble import RandomForestRegressor
     model = RandomForestRegressor(random_state=42)
     model.fit(X, y)
     feature_importances = pd.DataFrame({
         'Feature': X.columns,
         'Importance': model.feature_importances_
     })
     feature_importances = feature_importances.sort_values(by='Importance', ascending=False)
     top_features = feature_importances.head(10)
     print(top_features)
     X.drop(columns=['motor_type'], inplace=True)
```

```
         Feature  Importance
1           year    0.539081
0          model    0.224644
7    running_km     0.061097
6   motor_volume    0.056952
8   run_per_year    0.050471
5         status    0.030933
3          color    0.021651
4           type    0.011559
2    motor_type     0.003612
```

Ở đây, cột *year* chiếm tỉ lệ cao nhất (~ 53,91%).

Cuối cùng, ta áp dụng phương pháp *Recursive Feature Elimination* (RFE) để loại bỏ dần các đặc trưng chiếm tỉ lệ thấp và sau đó train lại.

```
[ ]  from sklearn.feature_selection import RFE

     rfe = RFE(estimator=RandomForestRegressor(), n_features_to_select=10)
     rfe.fit(X, y)
     selected_features = X.columns[rfe.support_]
     print(selected_features)
```

```
Index(['model', 'year', 'motor_volume', 'running_km', 'run_per_year'], dtype='object')
```

Sau khi loại bỏ, ta chọn ra được 5 đặc trưng cao nhất như hình.