

Monthly Product Sales Chart

# DỰ ĐOÁN GIÁ, XÁC ĐỊNH BẤT THƯỜNG GIÁ CHO XE MÁY TRÊN NỀN TẢNG CHỢ TỐT

75,000  
60,000  
45,000  
30,000



[bachxcdn@gmail.com](mailto:bachxcdn@gmail.com)  
[bachxcdn@gmail.com](mailto:bachxcdn@gmail.com)



**Hoang Anh Vo-Thi**  
[anhvo.bio@gmail.com](mailto:anhvo.bio@gmail.com)

# CONTENT

## **I. BUSINESS UNDERSTANDING**

→ *Bối cảnh, mục tiêu và vấn đề cần giải quyết*

## **II. DATA UNDERSTANDING**

→ *Dataset overview: attributes, sources, and structure*

## **III. DATA PREPARATION**

→ *Data cleaning, encoding, and feature engineering*

## **IV. MODELING**

→ *a. Price Prediction (Regression) – Dự đoán giá (Hồi quy)*

→ *b. Anomaly Detection – Phát hiện giá bất thường*

## **V. EVALUATION & INSIGHTS**

→ *Model comparison, key results, and feature analysis*

→ **Đánh giá mô hình, so sánh kết quả và phân tích đặc trưng**

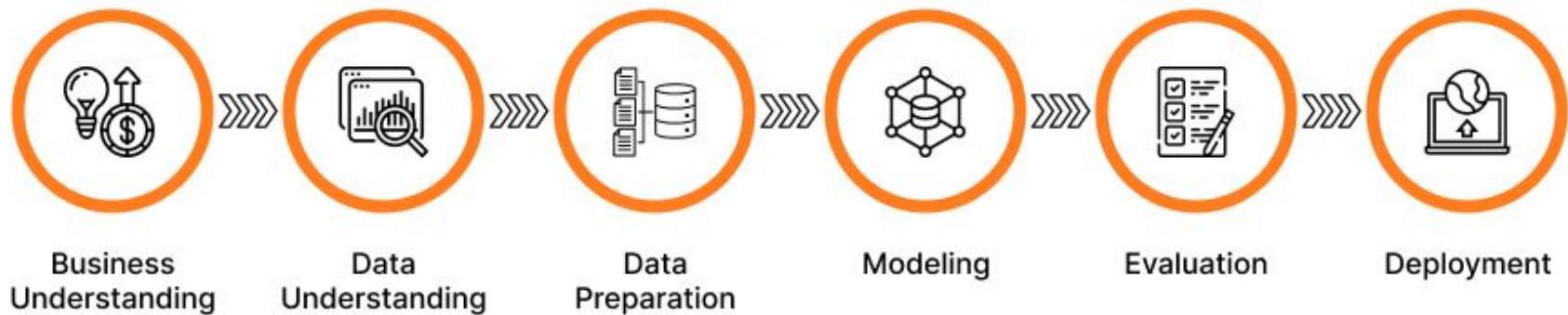
## **VI. DEPLOYMENT & FUTURE WORK**

→ *Demo, application, and future improvements*

## **VII. CONCLUSION & Q&A**

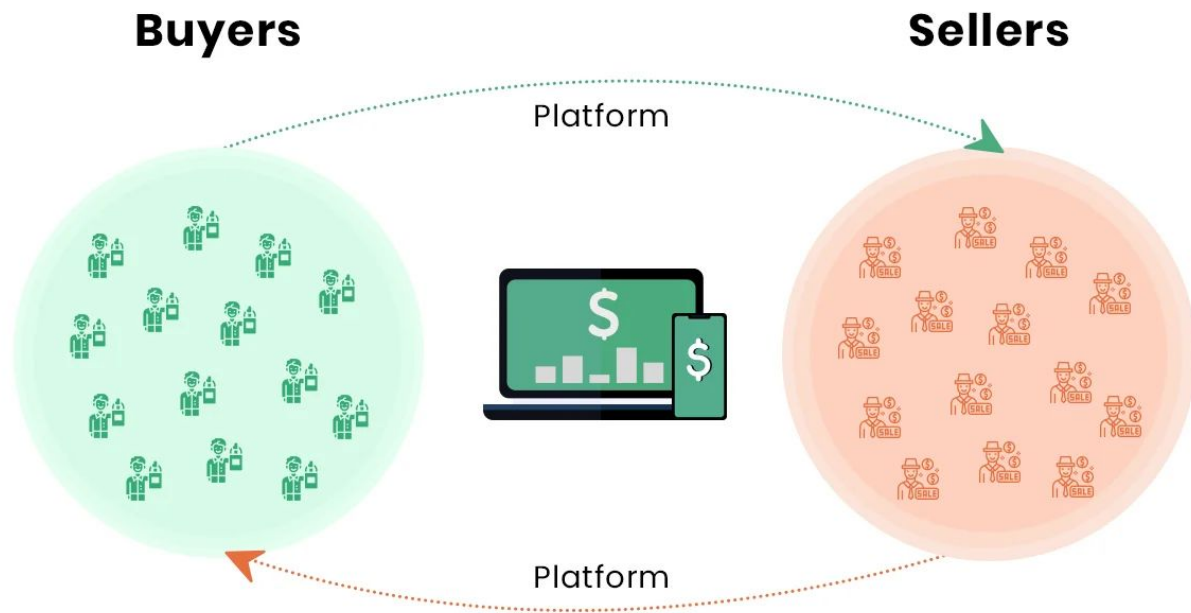
→ *Summary and discussion*

# DATA SCIENCE PROCESS



 **Chợ Tốt – Nền tảng mua bán trực tuyến lớn tại Việt Nam**

 **DATA – Thông tin về xe máy cũ được đăng bán trên nền tảng Chợ Tốt.**



## BÀI TOÁN – MỤC TIÊU – PHƯƠNG PHÁP

### 🎯 Mục tiêu

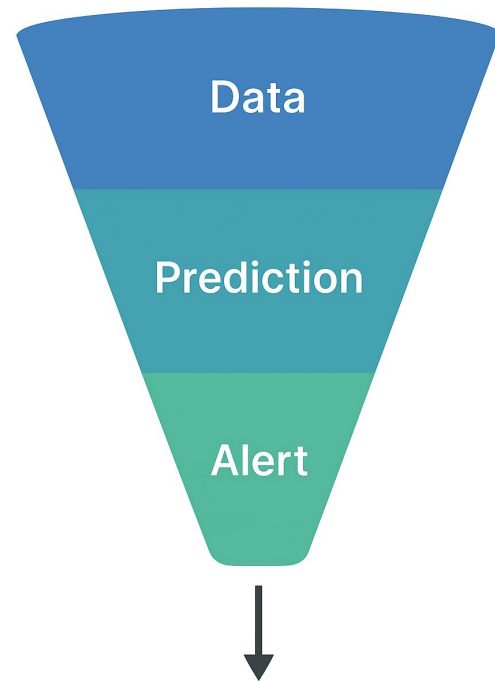
Dự đoán giá hợp lý cho xe máy cũ khi đăng bán.  
Phát hiện bài đăng bất thường có giá quá rẻ hoặc quá đắt.

### 🧩 Phương pháp

Dự đoán giá → Hồi quy (Regression).  
Phát hiện bất thường → Phát hiện bất thường (Anomaly Detection).

### 💡 Hai hướng tiếp cận này giúp:

Người bán nhận được gợi ý giá phù hợp.  
Nền tảng phát hiện và xử lý các bài đăng bất thường.



# DATA UNDERSTANDING

	id	Tiêu đề	Giá	Khoảng giá min	Khoảng giá max	Địa chỉ	Mô tả chi tiết	Thương hiệu	Dòng xe	Năm đăng ký	Số Km đã đi	Tình trạng	Loại xe	Ước tính xe	Xuất xứ	Chính sách bảo hành	Trọng lượng	Href
0	1	Bán Vespa Sprint 125cc 2024 xanh dương, xe đẹp...	66.000.000 đ	72.53 tr	85.14 tr	Phường Bến Thành, Quận 1, Tp Hồ Chí Minh	Bán xe #Vespa Sprint 125cc. Mua mới tại #Topco...	Piaggio	Vespa	2024	14000	Đã sử dụng	Tay ga	100 - 175 cc	Đang cập nhật	Bảo hành hãng	> 50 kg	<a href="https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...">https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...</a>
1	2	🔥 SH 150i Trắng ABS 2019 BSTP Chính Chủ	79.500.000 đ	62.76 tr	73.68 tr	Phường Tân Định, Quận 1, Tp Hồ Chí Minh	Bán SH 150i Trắng ABS 2019 Xám Bạc, Up Team X...	Honda	SH	2019	28000	Đã sử dụng	Tay ga	100 - 175 cc	Đang cập nhật	Bảo hành hãng	> 50 kg	<a href="https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...">https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...</a>
2	3	CC Vision Thể Thao 2023 Đen+bộ đèn Demi audi A7	37.000.000 đ	28 tr	32.86 tr	Phường Cầu Kho, Quận 1, Tp Hồ Chí Minh	Chính chủ bán Vision phiên bản Thể Thao 2023 Đ...	Honda	Vision	2023	12000	Đã sử dụng	Tay ga	100 - 175 cc	Đang cập nhật	Bảo hành hãng	> 50 kg	<a href="https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...">https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...</a>

- Dataset có 7208 dòng và 18 cột.
- Một vài cột chính:
- tiêu đề, giá, khoảng giá min/max, địa chỉ, thương hiệu, dòng xe, năm đăng ký, tình trạng, loại xe, dung tích xe, xuất xứ, href, ...

✔ **Nhận xét:** Dữ liệu dạng văn bản mô tả (Mô tả chi tiết, Địa chỉ, Tiêu đề) có thể chứa hashtag, ký tự đặc biệt, emoji 🔥 ... → Cần làm sạch để sử dụng trong phân tích NLP hoặc trích xuất thông tin.

Dữ liệu số (Giá, Km đã đi, Khoảng giá) có thể cần chuyển đổi về dạng số thực (float) để phân tích, vì hiện tại nhiều giá trị ở dạng chuỗi có ký tự “đ”, “tr”.

## Thư viện & công cụ sử dụng

numpy, pandas, matplotlib, seaborn, scikit-learn, xgboost, lightgbm, imbalanced-learn, PySpark, ydata-profiling,...



# EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

## Thống kê về giá

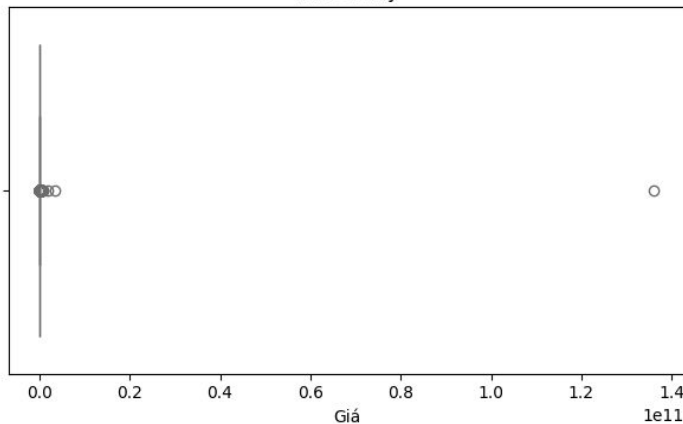
### THỐNG KÊ GIÁ TRƯỚC XỬ LÝ

```
count      7,203
mean      49,237,142
std      1,603,410,679
min         0
25%       8,500,000
50%      16,500,000
75%      32,500,000
max     136,000,000,000
Name: Giá, dtype: object
```

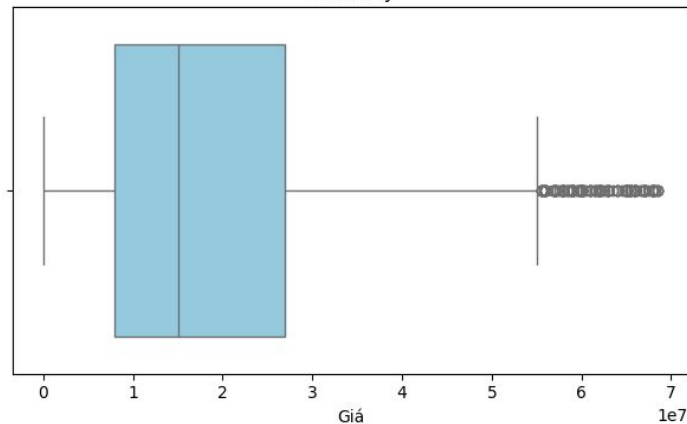
### THỐNG KÊ GIÁ SAU XỬ LÝ

```
count      6,544
mean      19,200,902
std      15,044,528
min         0
25%       8,000,000
50%      15,000,000
75%      26,922,500
max      68,500,000
Name: Giá, dtype: object
```

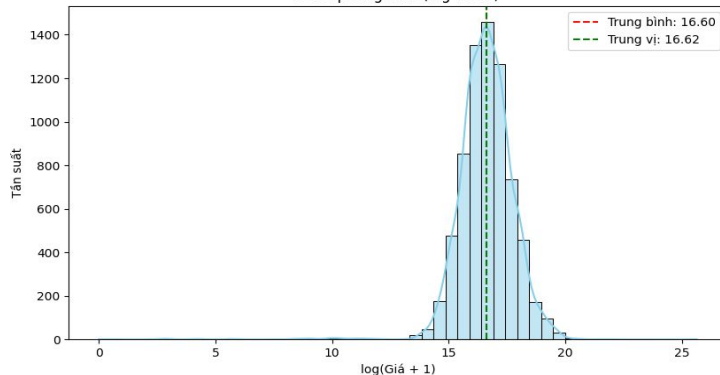
Trước xử lý



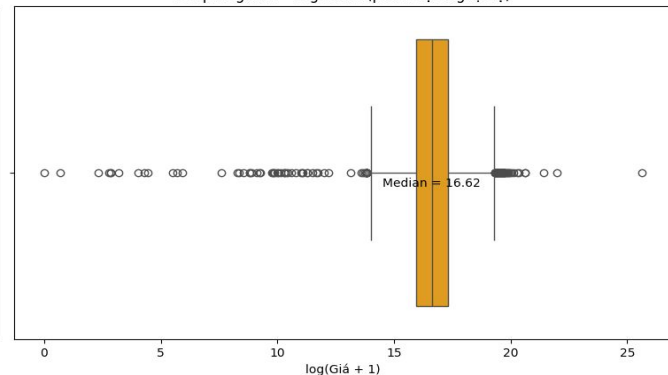
Sau xử lý



Phân phối giá xe (log-scale)



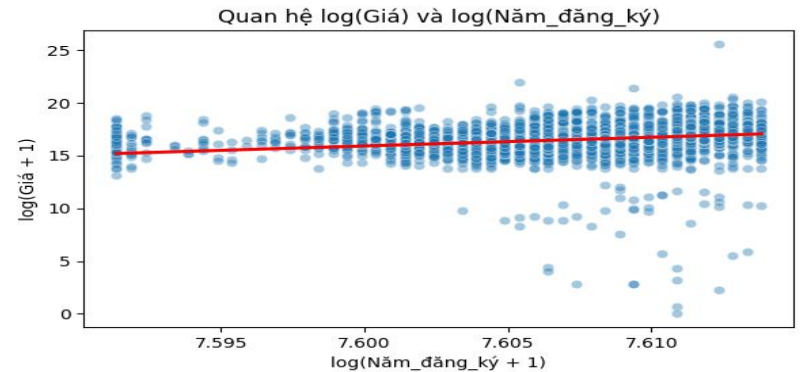
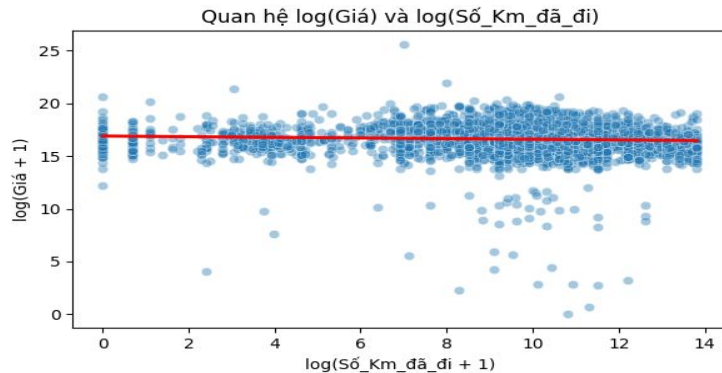
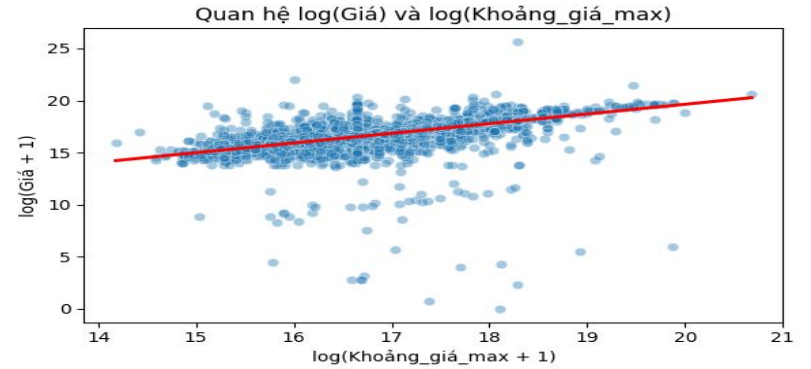
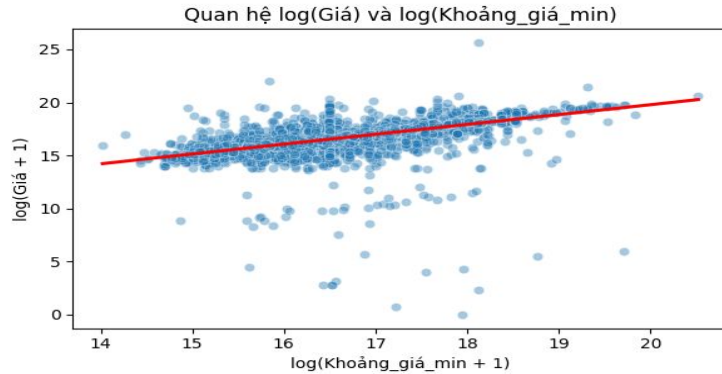
Boxplot giá xe - log-scale (phát hiện ngoại lệ)





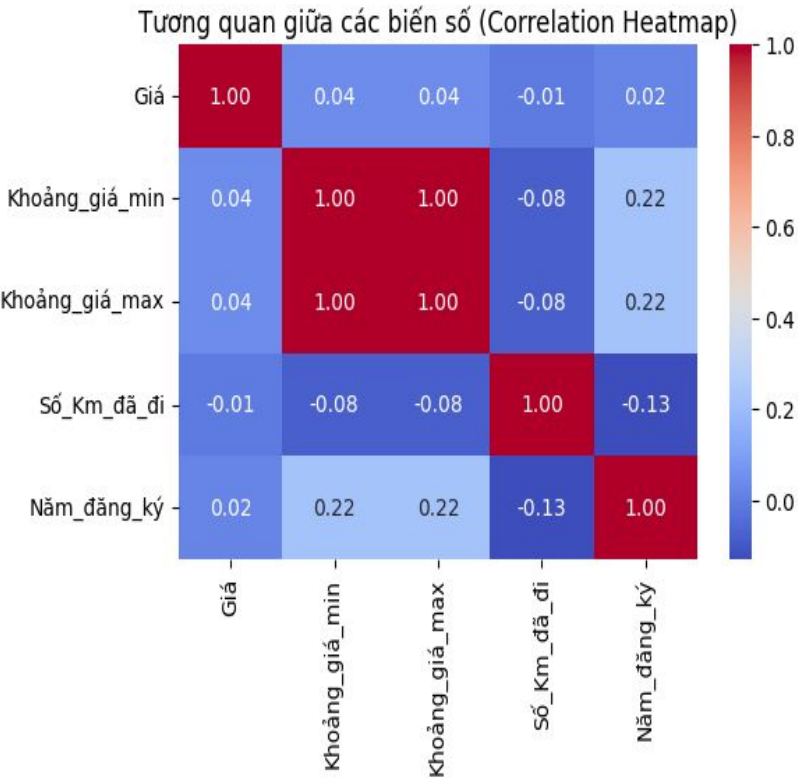
# EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

Các biến số (numeric variable) ảnh hưởng đến biến giá (xàì log)



# EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

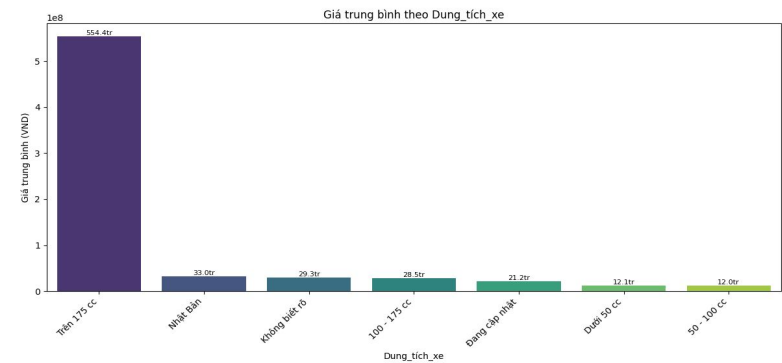
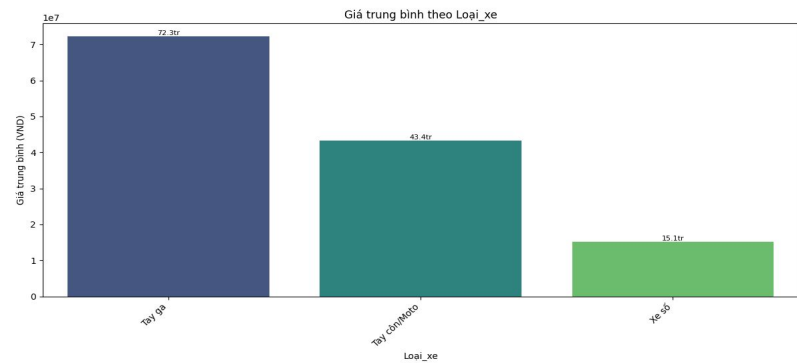
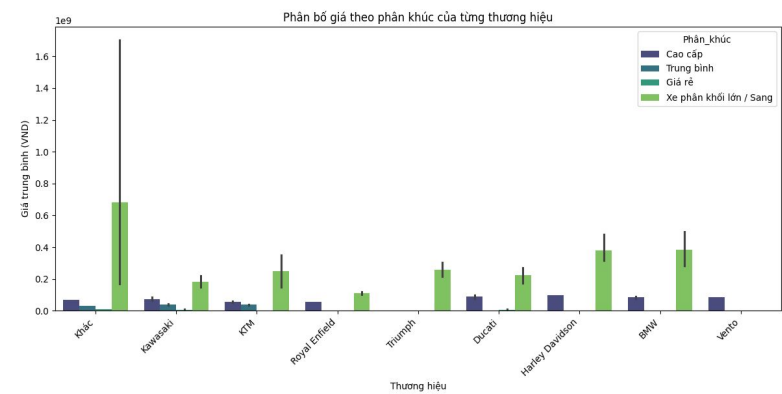
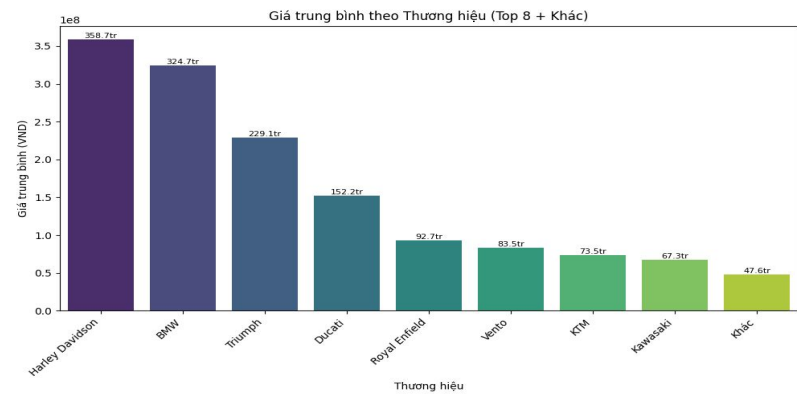
## Các biến số (numaric variable)



Cặp biến	Hệ số tương quan (r)	Nhận xét
Giá ↔ Khoảng_giá_min	0.04	Quan hệ rất yếu, gần như không tuyến tính
Giá ↔ Khoảng_giá_max	0.04	Tương tự, không có mối quan hệ tuyến tính rõ ràng
Giá ↔ Số_Km_đã_đi	-0.01	Cực kỳ yếu, số km gần như không ảnh hưởng đến giá
Giá ↔ Năm_đăng_ký	0.02	Gần như không có mối quan hệ tuyến tính đáng kể

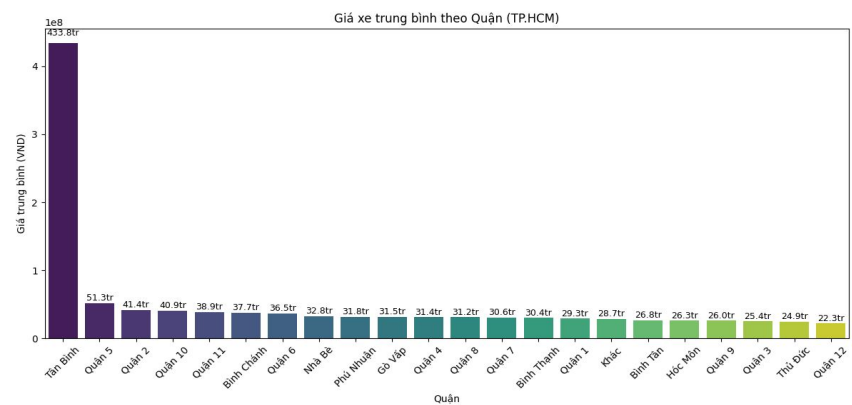
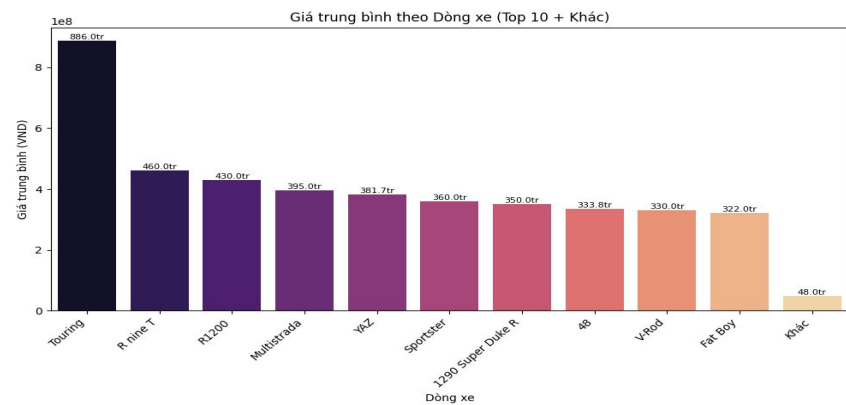
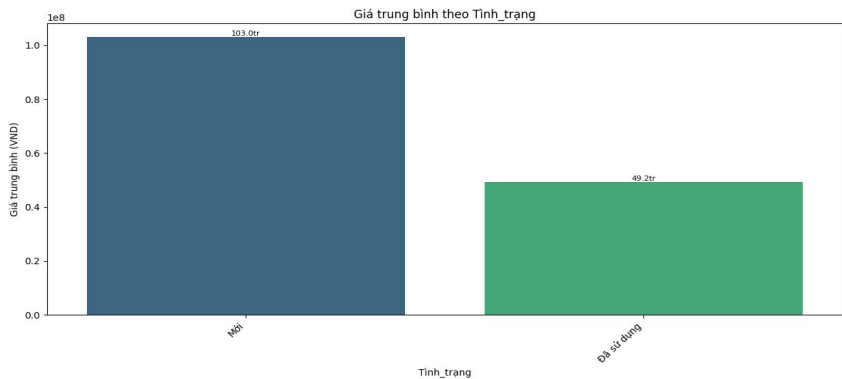
# EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

Các biến phân loại (categorical variable) ảnh hưởng đến biến giá



# EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

Các biến phân loại (categorical variable) ảnh hưởng đến biến giá



# EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

Kiểm định Chi-square giữa các biến phân loại  
Dựa vào giá trị **\*\*p-value\*\***, với ngưỡng  $\alpha = 0.05$ :

- Nếu  $p\text{-value} < 0.05 \rightarrow$  có mối quan hệ.
- Nếu  $p\text{-value} \geq 0.05 \rightarrow$  độc lập.

Cặp biến	p-value	Kết luận
Thương_hiệu ↔ Dòng_xe	0	✔ Có mối quan hệ chặt chẽ
Thương_hiệu ↔ Loại_xe	$1.894 \times 10^{-289}$	✔ Có mối quan hệ mạnh
Thương_hiệu ↔ Dung_tích_xe	0	✔ Có mối quan hệ rõ rệt
Thương_hiệu ↔ Tình_trạng	0	✔ Có mối quan hệ đáng kể
Thương_hiệu ↔ Xuất_xứ	0	✔ Có mối quan hệ rất mạnh
Dòng_xe ↔ Tình_trạng	1.0	⚠ Gần như độc lập
Loại_xe ↔ Tình_trạng	0.13	⚠ Gần như độc lập



## DATA INSIGHT:

Dữ liệu sau làm sạch phản ánh chính xác hơn thị trường xe cũ, giảm nhiều từ các giá trị cực đoan.

Giá xe bị chi phối mạnh bởi yếu tố thương hiệu, dung tích và tình trạng xe, trong khi các yếu tố kỹ thuật khác (năm đăng ký, số km) tác động yếu.

Phân khúc cao cấp (Harley, BMW, Triumph, Touring...) có mức giá vượt trội, thể hiện rõ ranh giới thị trường giữa xe phổ thông và xe sang.

Xe tay ga và xe phân khối lớn là nhóm có biên độ giá rộng, phù hợp cho các mô hình dự đoán và phát hiện bất thường.

Vị trí địa lý ảnh hưởng gián tiếp đến giá – khu vực trung tâm (Tân Bình, Phú Nhuận, Quận 5) có giá cao hơn vùng ven (Thủ Đức, Bình Tân).

Phân tích từ khóa gợi ý khả năng phát hiện hành vi người bán:

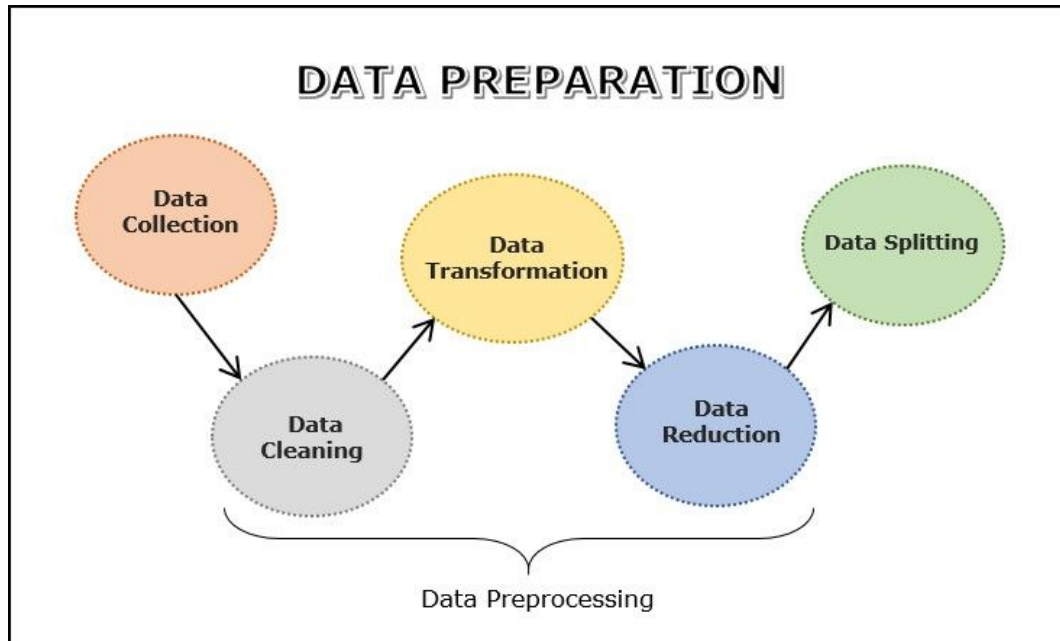
Tin giá cao → mô tả kỹ, nhiều chi tiết tin cậy.

Tin giá thấp → mô tả ngắn, nhấn mạnh thanh lý, bán nhanh.

Các yếu tố này có thể được dùng để huấn luyện mô hình dự đoán giá hợp lý và phát hiện tin bất thường (giá ảo, nhập sai hoặc gian lận).

# DATA PREPARATION

- Loại null, trùng lặp, ngoại lệ, các cột không cần thiết.
- Xử lý văn bản text
- Mã hóa biến phân loại (Label).
- Tạo đặc trưng mới (tuổi xe, giá/km, quận).
- Chia train/test.








# DATA PREPARATION

## Quyết định giữ / bỏ cột

Quyết định	Cột	Lý do
✓ Giữ	Khoảng_giá_min	Là mốc giá thị trường, đại diện cho giá trung bình hợp lý nhất.
✓ Giữ	Tuổi_xe, log_Km, Km_trên_năm	Phản ánh hao mòn và mức độ sử dụng xe — tác động mạnh đến giá.
✓ Giữ	Dòng_xe_top, Loại_xe, Dung_tích_xe, Tình_trạng, Xuất_xứ, Quận, Phân_khúc, Thương_hiệu	Thể hiện phân khúc, đặc điểm kỹ thuật và vị trí xe — giúp mô hình học được khác biệt giữa các nhóm.
— Bỏ	Giá	Được dùng làm biến mục tiêu riêng (label_log).
✗ Bỏ	id, Href, Địa_chỉ, Tiêu_đề, Mô_tả_chi_tiết, tieu_de_clean, mo_ta_chi_tiet_clean, text_all_clean	text_all_clean không dùng trong PySpark vì TF-IDF làm phình dữ liệu, dễ gây lỗi bộ nhớ; tuy nhiên vẫn hữu ích cho mô hình ML truyền thống (RandomForest, XGBoost) nếu xử lý riêng.
✗ Bỏ	Khoảng_giá_max, Thương_hiệu_top, Chính_sách_bảo_hành, Trọng_lượng	Tương quan cao hoặc ít giá trị phân biệt, gây nặng pipeline.







# MODELING (Regression)

## Pyspark

Mô hình	Ưu điểm	Giải thích & Ứng dụng
 <b>Random Forest</b>	Ổn định, dễ song song hóa, chống overfitting	Thích hợp cho dữ liệu lớn và phức tạp; mô hình tạo nhiều cây quyết định độc lập nên chạy tốt trong Spark cluster, cho kết quả ổn định và ít cần tinh chỉnh.
 <b>Gradient Boosting (GBRegressor)</b>	Độ chính xác cao, tối ưu hóa mạnh trong Spark MLlib	Kết hợp dãy cây liên tiếp để giảm sai số còn lại; phù hợp khi cần dự đoán chính xác giá trị liên tục như <b>giá xe</b> ; hiệu quả hơn RF nếu dữ liệu đã được xử lý kỹ.
 <b>Ridge Regression</b>	Nhanh, nhẹ, dễ giải thích hệ số	Là mô hình tuyến tính có regularization giúp giảm nhiễu; phù hợp làm baseline, kiểm tra quan hệ tuyến tính giữa biến độc lập và giá bán, hoặc chạy thử trên tập lớn để benchmark.




# MODELING (Regression)

## ML truyền thống

Mô hình	Ưu điểm nổi bật	Giải thích & Ứng dụng
 <b>Random Forest</b>	Ổn định, ít cần tuning, kháng nhiễu tốt	Dễ triển khai, phù hợp cho bài toán tabular như dự đoán giá xe; cho baseline đáng tin cậy.
 <b>Gradient Boosting</b>	Chính xác cao, xử lý tốt quan hệ phi tuyến	Xây chuỗi cây liên tiếp để tối ưu lỗi còn lại; phù hợp khi cần độ chính xác cao hơn RF.
 <b>CatBoost</b>	Tối ưu cho dữ liệu phân loại, không cần mã hóa	Tự xử lý biến dạng category; tránh overfitting tốt, thường outperform GBT khi dữ liệu có nhiều biến rời rạc.
 <b>XGBoost</b>	Nhanh, hiệu quả, nhiều tham số tối ưu	Rất phổ biến, dễ tinh chỉnh; phù hợp khi muốn tối đa hóa hiệu suất và độ chính xác.
 <b>LightGBM</b>	Huấn luyện nhanh, tiết kiệm RAM, mạnh với dữ liệu lớn	Dùng khi cần tốc độ cao, đặc biệt với dữ liệu nhiều chiều hoặc tập lớn.
 <b>Ridge Regression</b>	Đơn giản, dễ giải thích, kiểm soát overfitting	Mô hình tuyến tính cơ bản, dùng làm baseline hoặc phân tích ảnh hưởng của từng biến đầu vào.

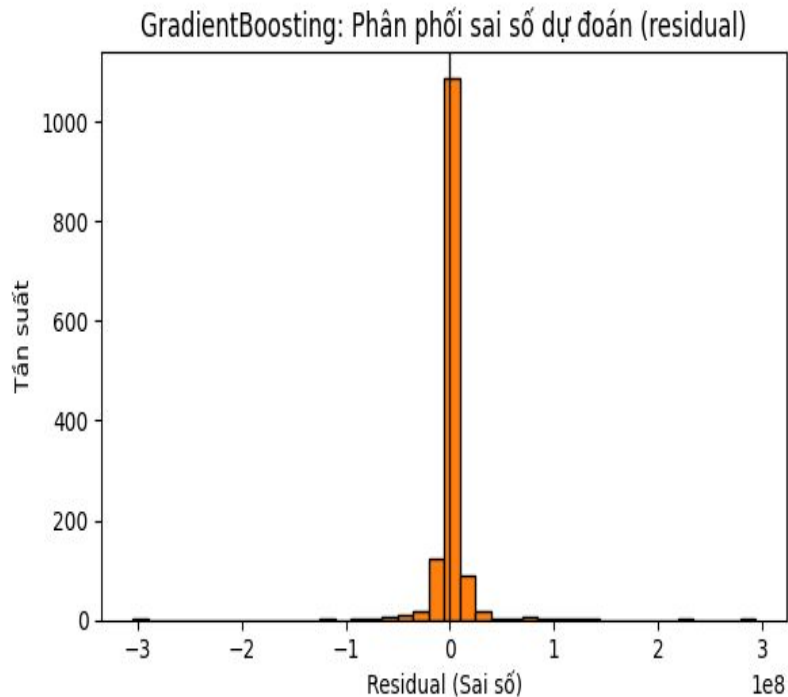
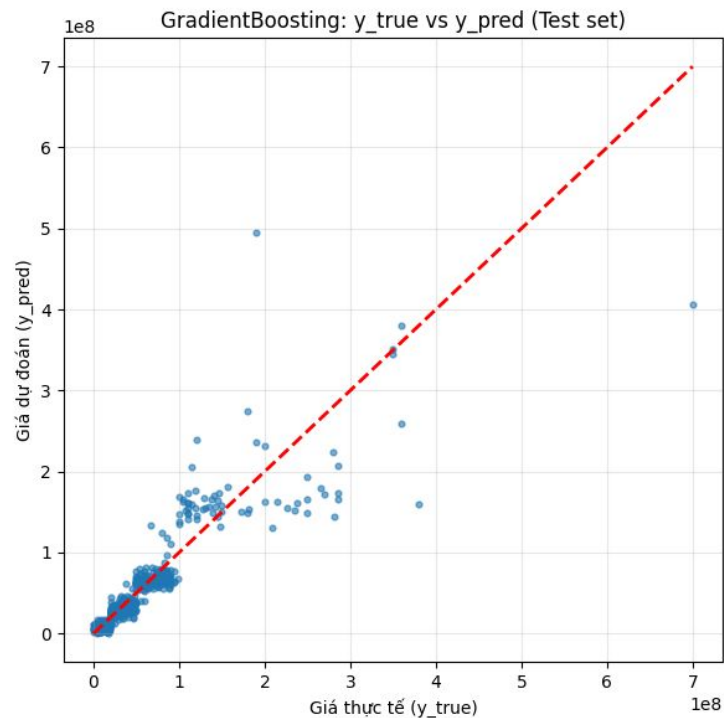
→ Kết quả đánh giá: Dựa trên giá thật (không biến đổi log)

Pyspark

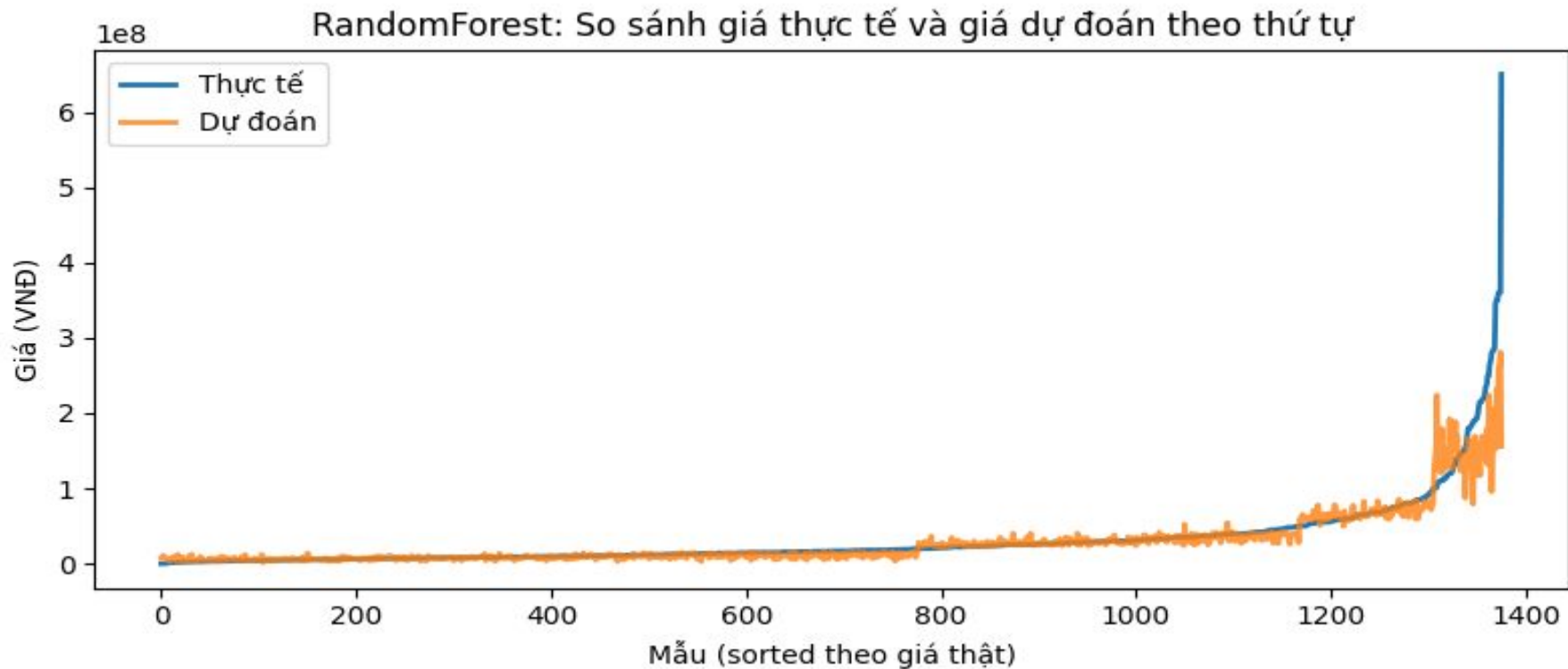
Mô hình	$R^2$ (Test)	MAE (Test)	RMSE (Test)	Thời gian (s)
 Random Forest	0.799	6,525,991	20,885,617	107.35
 Ridge Regression	0.533	12,584,408	31,862,088	14.07
 Gradient Boosting	0.412	8,112,962	35,745,903	114.18

→ Kết quả đánh giá:

Pyspark









## So sánh giá thực tế và giá dự đoán theo thứ tự (trên pyspark)



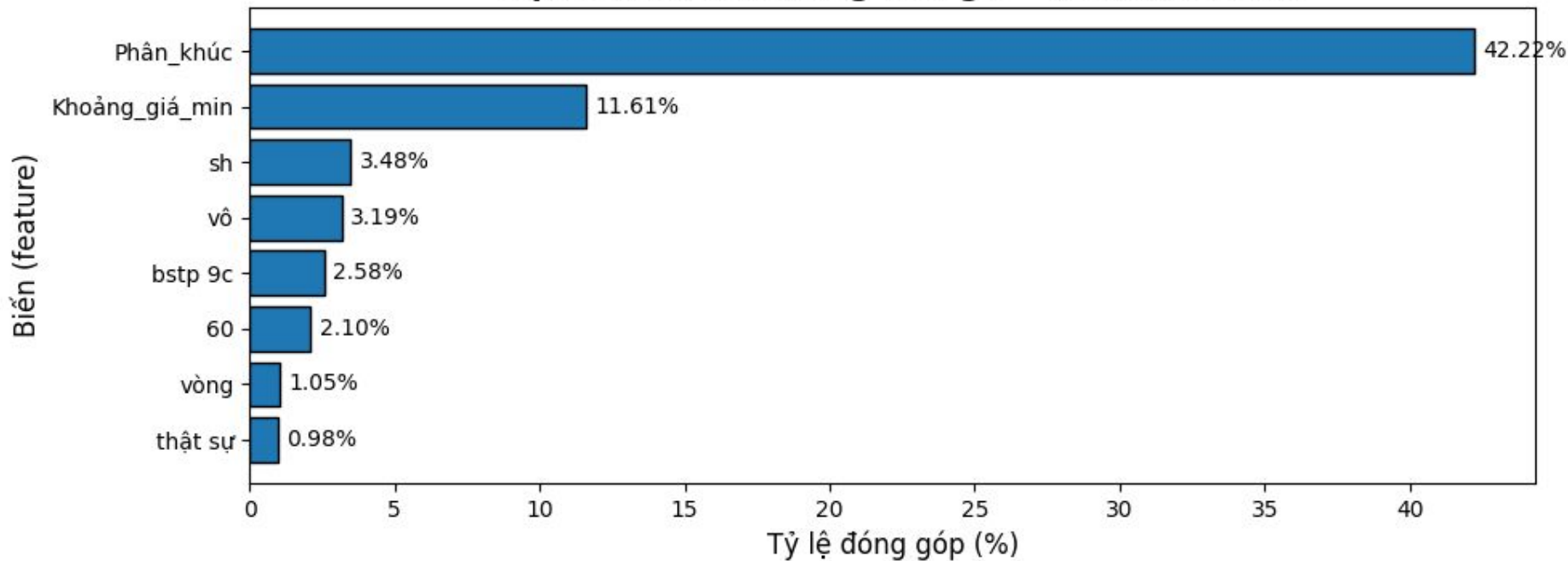
→ Kết quả đánh giá: Dựa trên giá thật (không biến đổi log)

## ML truyền thống

Mô hình	$R^2$ (Test)	MAE (Test)	RMSE (Test)	Thời gian (s)
 Random Forest	0.890	3,687,591	23,934,808	293.2
 Gradient Boosting	0.875	3,682,696	27,208,377	243.2
 CatBoost	0.863	4,002,870	29,976,868	396.5
 XGBoost	0.858	3,832,443	30,924,787	95.8
 LightGBM	0.285	4,939,162	156,238,045	45.1
 Ridge Regression	0.260	7,806,546	161,607,611	2.5

# ML truyền thống

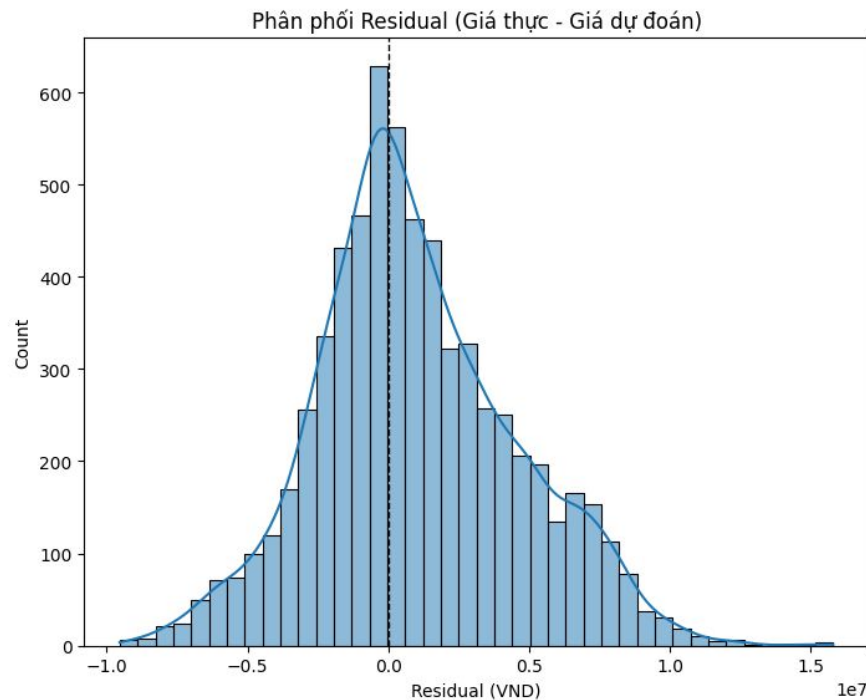
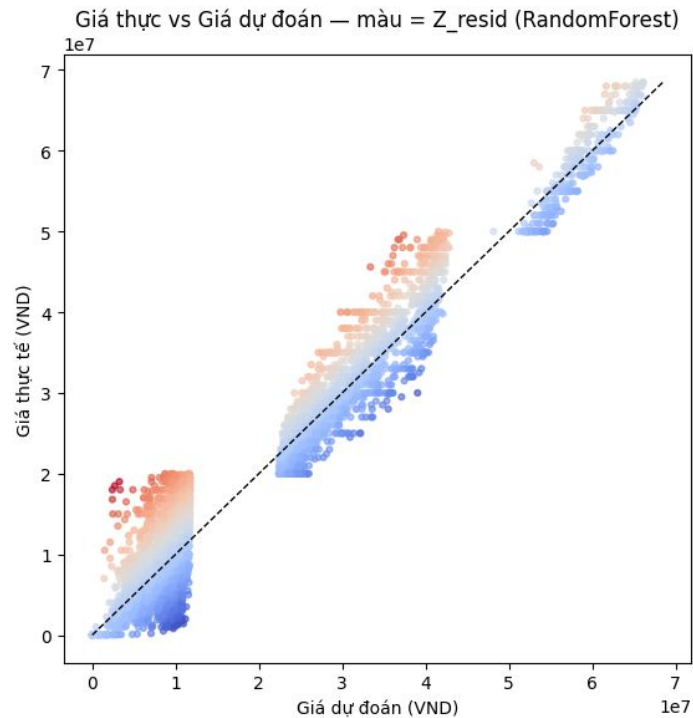
**Top 8 biến ảnh hưởng đến giá - RandomForest**





→ Kết quả đánh giá:

## ML truyền thống



→ Kết quả đánh giá:  
5 dòng ngẫu nhiên để kiểm tra mô hình

STT	Giá thực tế (VND)	Giá dự đoán (VND)	Sai lệch (VND)	Lệch (%)
1	33,000,000	37,966,662	4,966,062	15.0%
2	55,500,000	55,098,578	401,422	0.7%
3	10,500,000	9,790,583	709,417	6.8%
4	10,000,000	11,060,004	1,060,004	10.6%
5	5,600,000	4,578,752	1,021,248	18.2%

→ Kết quả đánh giá:  
5 dòng có độ lệch cao nhất

STT	Giá thực tế (VND)	Giá dự đoán (VND)	Sai lệch (VND)	Lệch (%)
1	10,500	10,826,702	10,016,202	95,932.4%
2	7,300	6,446,279	6,438,979	88,025.2%
3	6,800	5,732,646	5,725,846	84,203.6%
4	66,000	6,490,453	6,424,453	9,734.0%
5	80,000	6,410,070	6,330,070	7,912.6%

## Nhận xét tổng quan giữa hai môi trường:

- ♦ Hiệu năng mô hình

ML truyền thống đạt độ chính xác cao hơn ( $R^2 \sim 0.86\text{--}0.89$ ) nhờ khả năng tinh chỉnh và mô hình boosting mạnh.

PySpark cho hiệu năng thấp hơn một chút ( $R^2 \sim 0.8$ ) nhưng vẫn đủ ổn định trên dữ liệu lớn.

Random Forest là mô hình hiệu quả nhất ở cả hai môi trường, cân bằng giữa độ chính xác và thời gian huấn luyện.

- ♦ Thời gian và hiệu suất

PySpark tối ưu tốt khi xử lý dữ liệu quy mô lớn, phân tán, song song hóa hiệu quả.

ML truyền thống nhanh hơn trên tập nhỏ nhưng tốn bộ nhớ khi mở rộng.

Các mô hình như LightGBM, XGBoost có tốc độ nhanh và chính xác cao → phù hợp cho chạy local hoặc batch nhỏ.




# Quy luật phát hiện giá bất thường (Sử dụng ML truyền thống kết hợp với phương pháp xử lý bất thường)

## Nguyên tắc chung

- Dựa trên kết hợp giữa quy tắc (rule-based) và mô hình thống kê / ML.
- Chỉ xét xe có vi phạm\_minmax = 1 → giá nằm ngoài khoảng hợp lý.
- Mỗi xe được gán điểm bất thường tổng hợp (abnormal\_score) từ 0 → 100.

## Cách tính điểm

- Score A (30%) → dựa trên độ lệch  $Z_{\text{resid}}$  giữa giá thực tế và giá dự đoán.
- $|Z|$  càng lớn → giá càng bất thường.
- Score B (70%) → dựa trên B\_score (mức độ bất thường mô hình).
- So sánh với ngưỡng bách phân vị 98% để xác định bất thường mạnh.
- Công thức tổng hợp:
- $\text{abnormal\_score} = 0.3 \times A + 0.7 \times B$
- Nếu  $\text{abnormal\_score} \geq 60$  → đánh dấu “bất thường mạnh”.

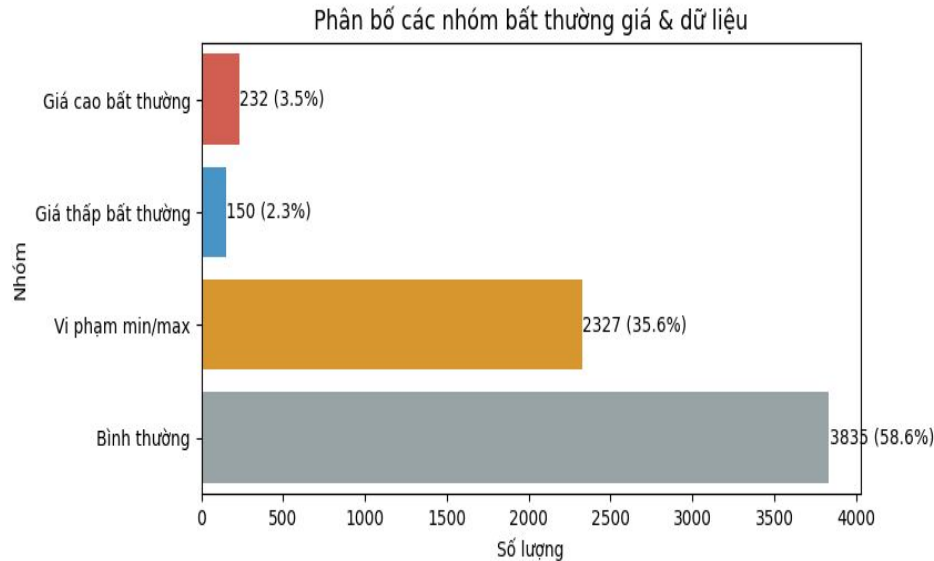
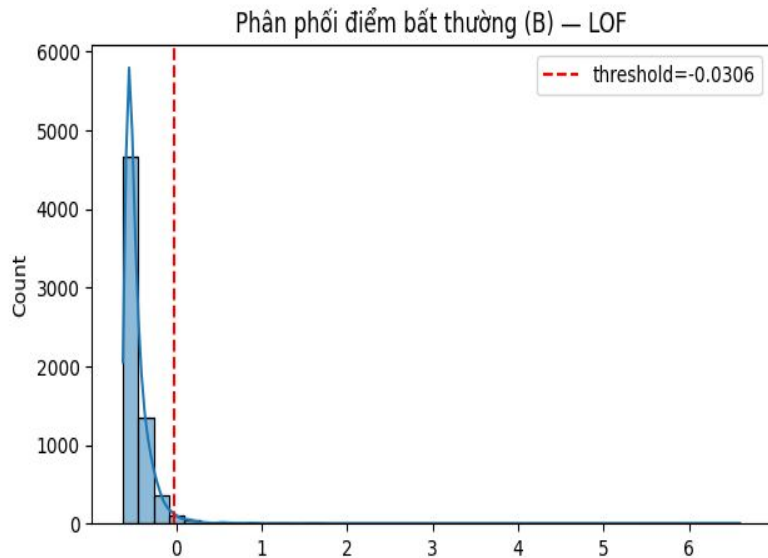
Điều kiện	Kết luận
$Z \geq +2$	 Giá cao bất thường
$Z \leq -2$	 Giá thấp bất thường
Vi phạm min/max khác	 Vi phạm biên giá
Không vi phạm	 Bình thường

→ Phương pháp:  
Isolation Forest, LOF, One-Class SVM.

Mô hình	AUC (weak)	AP (weak)	Thời gian (s)
LOF (Local Outlier Factor)	0.742	0.746	0.62
Isolation Forest	0.713	0.726	1.45
One-Class SVM	0.543	0.583	0.25

# Evaluation & Comparison

- Regression: Random Forest
- Anomaly Detection: Isolation Forest ổn định nhất.
- Các chỉ số đánh giá tổng hợp (RMSE, MAE, Recall anomaly).



## Save the model and apply it to new data.

```
Đã lưu model A: /content/gdrive/MyDrive/DL07_K308_VoThiHoangAnh_NguyenMaiXuanBach/Project_1/models_final_project_1_bai2/model_A_price_predictor.pkl
Đã lưu model B: /content/gdrive/MyDrive/DL07_K308_VoThiHoangAnh_NguyenMaiXuanBach/Project_1/models_final_project_1_bai2/model_B_lof.pkl
Đã lưu TF-IDF vectorizer
Đã lưu LabelEncoders
```

Tất cả mô hình đã được lưu tại thư mục: /content/gdrive/MyDrive/DL07\_K308\_VoThiHoangAnh\_NguyenMaiXuanBach/Project\_1/models\_final\_project\_1\_bai2

## Mẫu thử

## Kết quả khi áp dụng

```
data_new = pd.DataFrame([
{
    "Khoảng_giá_min": 45_000_000,
    "NămĐăng_ký": 2020,
    "Số_Km_đã_đi": 20000,
    "Thương_hiệu": "Honda",
    "Dòng_xe": "SH",
    "Loại_xe": "Tay ga",
    "Dung_tích_xe": "100 - 175 cc",
    "Tình_trạng": "Đã sử dụng",
    "Xuất_xứ": "Đang cập nhật",
    "Địa_chỉ": "Phường Bến Nghé, Quận 1, Tp Hồ Chí Minh",
    "Tiêu_đề": "Bán SH Mode 125 chính chủ",
    "Mô_tả_chi tiết": "Xe đẹp, bao test, biển số TP, giá 52tr."
},
{
    "Khoảng_giá_min": 60_000_000,
    "NămĐăng_ký": 2021,
    "Số_Km_đã_đi": 12000,
    "Thương_hiệu": "Vespa",
    "Dòng_xe": "Sprint",
    "Loại_xe": "Tay ga",
    "Dung_tích_xe": "100 - 175 cc",
    "Tình_trạng": "Đã sử dụng",
    "Xuất_xứ": "Đang cập nhật",
    "Địa_chỉ": "Quận 7, Tp Hồ Chí Minh",
    "Tiêu_đề": "Vespa Sprint 125cc cần bán gấp",
    "Mô_tả_chi tiết": "Chính chủ chạy kỹ. Giá 30tr (deal nhẹ)."
}
])
data_new["Giá"] = [52_000_000, 30_000_000]
```

	Thương_hiệu	Dòng_xe	Khoảng_giá_min	NămĐăng_ký	Tuổi_xe	Số_Km_đã_đi	Km_trên_năm	Giá	Giá_dự_đoán	Kết_luận_cuối
0	10	SH	45000000	2020	5	20000	4000.0	52000000	56497404.0	Bình thường
1	0	Sprint	60000000	2021	4	12000	3000.0	30000000	38294093.0	Giá thấp bất thường



## KẾT QUẢ & INSIGHT

**Mô hình tốt nhất:** Random Forest → cho độ chính xác cao nhất trong dự đoán giá. Còn LOF (Local Outlier Factor) → hiệu quả nhất trong phát hiện bất thường, giúp nhận diện cả giá lệch ngữ cảnh và điểm dữ liệu lạ.

### Kết quả

- Bình thường: 3,835 bản ghi ( $\approx 58\%$ ) → Phần lớn dữ liệu có giá hợp lý, chứng tỏ mô hình hoạt động ổn định.
- Vi phạm min/max: 2,327 bản ghi ( $\approx 35\%$ ) → Giá rơi ra ngoài khoảng giá tham chiếu (cao hơn hoặc thấp hơn khung hợp lý).  
Thường gặp ở xe cũ, xe độ, hoặc xe bán gấp.  
Nhóm này không phải lỗi, mà là vùng rủi ro cao, cần xem xét kỹ trong kiểm duyệt.
- Giá bất thường: 382 bản ghi ( $\approx 6\%$ ) → Xe được rao giá quá cao so với mô hình, có thể là phiên bản hiếm hoặc người bán nâng giá. Hoặc là Xe có giá rao thấp bất thường, thường do nhập sai, hỏng hóc hoặc quảng cáo mời giá rẻ.

### Ứng dụng:

- Gợi ý giá tự động cho người bán.
- Cảnh báo kiểm duyệt khi giá bất hợp lý.
- Phân tích xu hướng thị trường theo khu vực & dòng xe.








**bachxdn@gmail.com**

bachxdn@gmail.com



**Hoang Anh Vo-Thi**

anhvo.bio@gmail.com

Thành viên	Nhiệm vụ
 Võ Thị Hoàng Anh	Xây dựng mô hình dự đoán giá Soạn thuyết trình
 Nguyễn Mai Xuân Bách	Khám phá & xử lý dữ liệu Phát hiện bất thường Kiểm tra code
 Cả hai	Viết báo cáo So sánh kết quả Chuẩn bị slide