ĐỒ ÁN TỐT NGHIỆP DATA SCIENCE (DL07_308T37_ON)

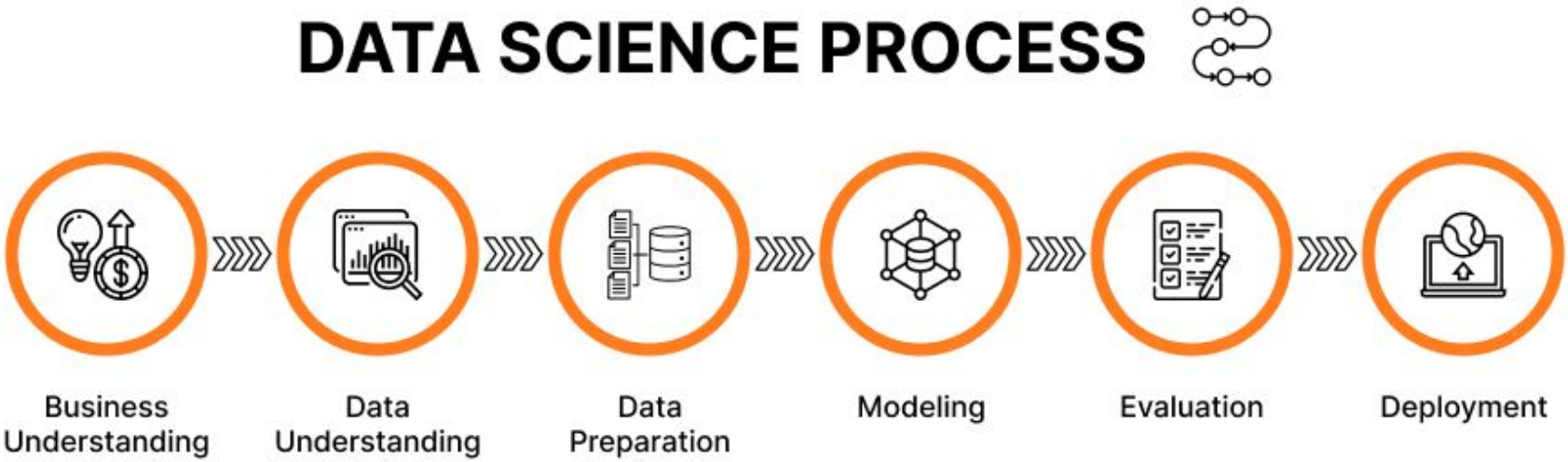
DỰ ĐOÁN VÀ XÁC ĐỊNH BẤT THƯỜNG GIÁ XE MÁY TRÊN NỀN TẢNG CHỢ TỐT

GVHD: Th.S. KHUẤT THÙY PHƯƠNG

HVTH: NGUYỄN MAI XUÂN BÁCH
VÕ THỊ HOÀNG ANH

Tp.HCM, 29/11/2025

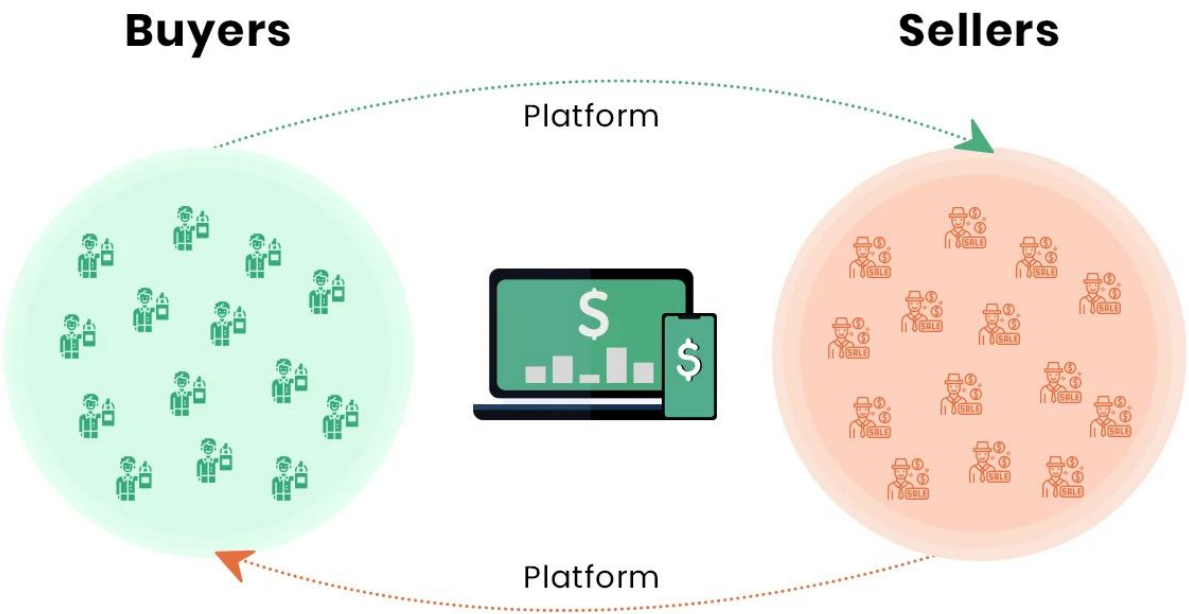
CONTENT



Nguồn hình: https://intellipaat.com/blog/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.

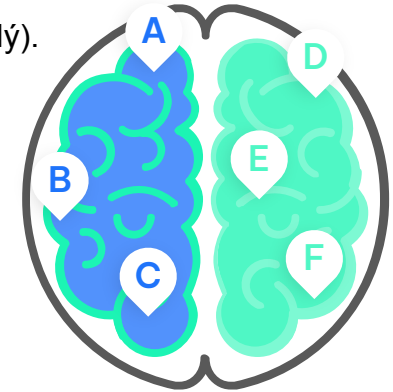


Nguồn hình: https://www.excellentwebworld.com/two-sided-marketplace/

BÀI TOÁN & MỤC TIÊU

🎯 Bối cảnh & Vấn đề

- ❖ Giá xe máy cũ trên thị trường biến động lớn theo hãng, dòng xe, năm sản xuất, tình trạng và số km đã dùng.
- ❖ Người bán thường định giá theo cảm tính → giá thiếu nhất quán, chênh lệch so với giá trị thực.
- ❖ Xuất hiện nhiều tin đăng bất thường: giá quá rẻ (nguy cơ lừa đảo) hoặc quá cao (không hợp lý).

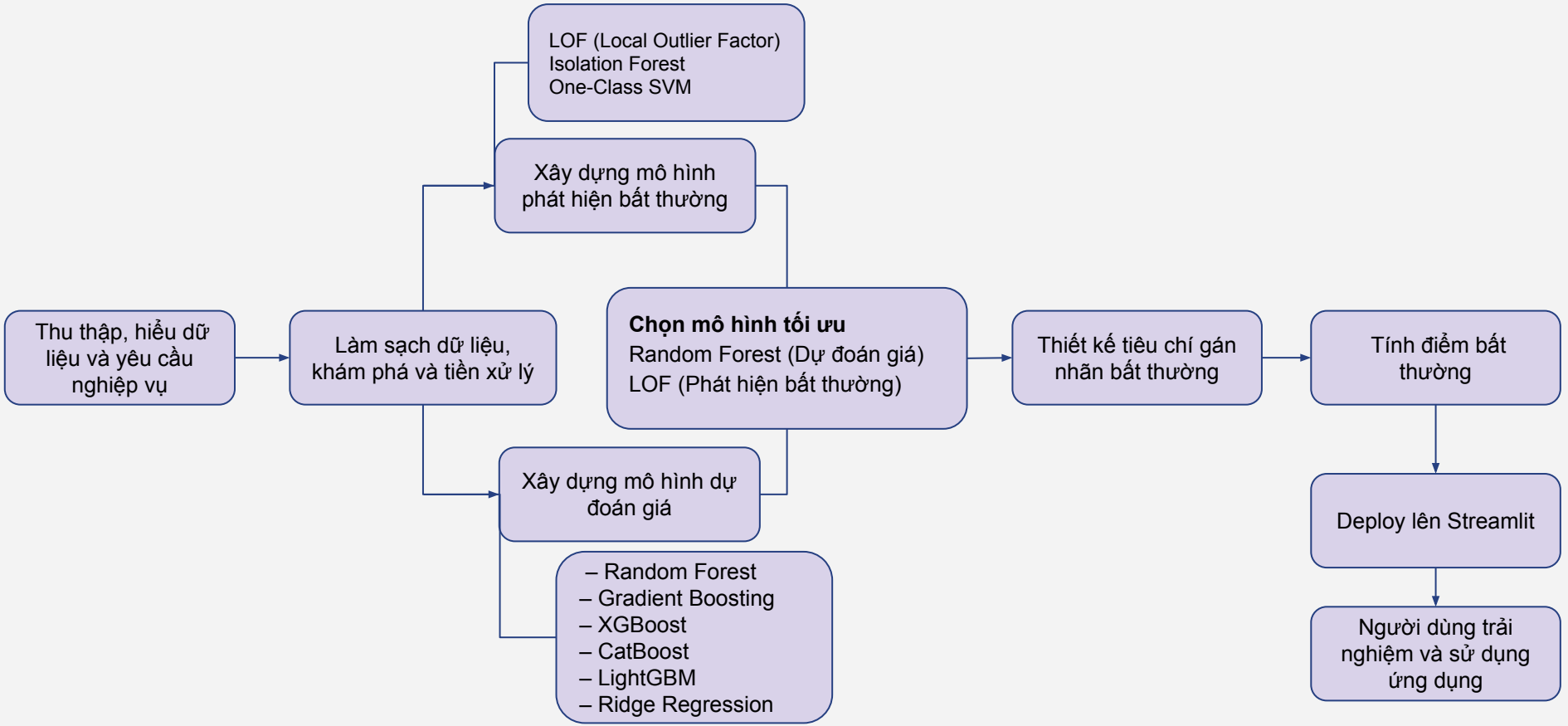


🎯 Mục tiêu của dự án

- ➢ Xây dựng mô hình dự đoán mức giá hợp lý cho từng tin đăng dựa trên dữ liệu thực.
- ➢ Áp dụng phát hiện bất thường (Anomaly Detection) để nhận diện các tin đăng có giá sai lệch.
- ➢ Giúp người mua tham khảo giá chính xác hơn và nền tảng có cơ sở kiểm duyệt các tin bất thường.

QUY TRÌNH THỰC HIỆN

Từ tiền xử lý dữ liệu tới ứng dụng cho người dùng cuối



🛠️ 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	Dung tích 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	https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...
1	2	🔥🔥 SH 150i Thắng 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 Thắng ABS 2019 Xám Bạc, Úp Team X...	Honda	SH	2019	28000	Đã sử dụng	Tay ga	100 - 175 cc	Đang cập nhật	Bảo hành hãng	> 50 kg	https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...
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	https://xe.chotot.com/mua-ban-xe-may-quan-1-tp...

- 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.

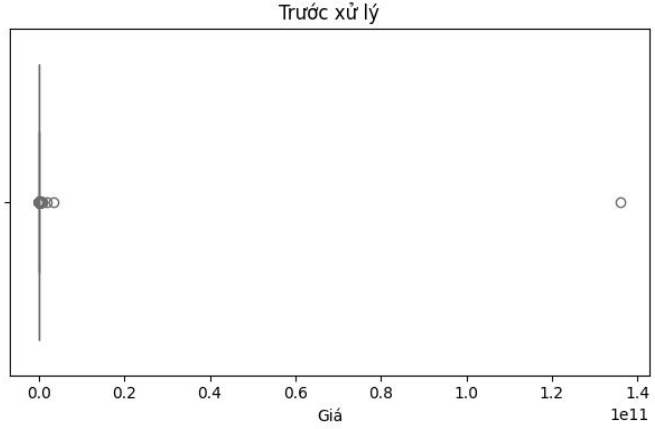
Dữ liệu số (Giá, Km đã đi, Khoảng giá) có thể cần chuyển đổi về dạng số thực (float) để phân tích, vì hiện tại nhiều giá trị ở dạng chuỗi có ký tự "đ", "tr".

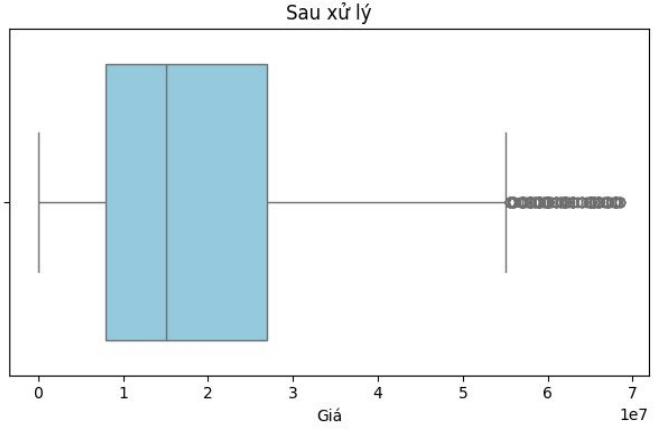
🛠️ EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

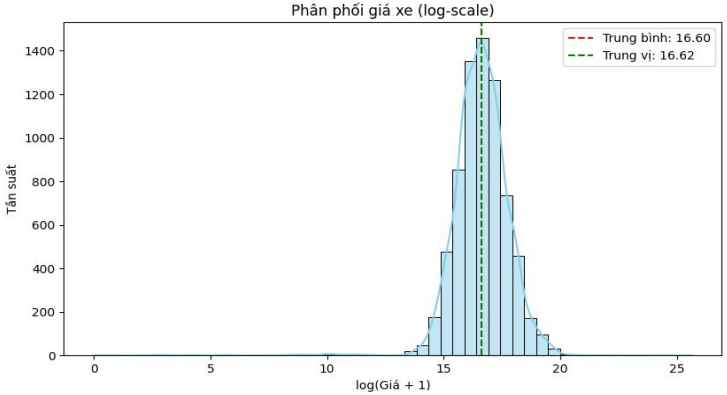
Thống kê về giá

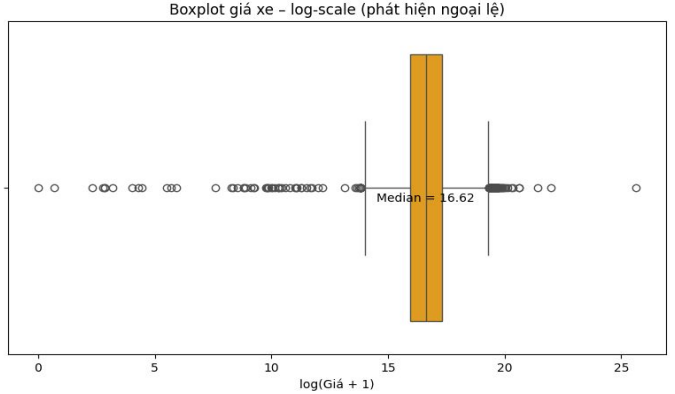
```
 THỐNG KÊ GIÁ TRƯỚC XỬ LÝ
count              7,203
mean          49,237,142
std        1,603,410,679
min                   0
25%           8,500,000
50%          16,500,000
75%          32,500,000
max     136,000,000,000
Name: Giá, dtype: object

 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
```



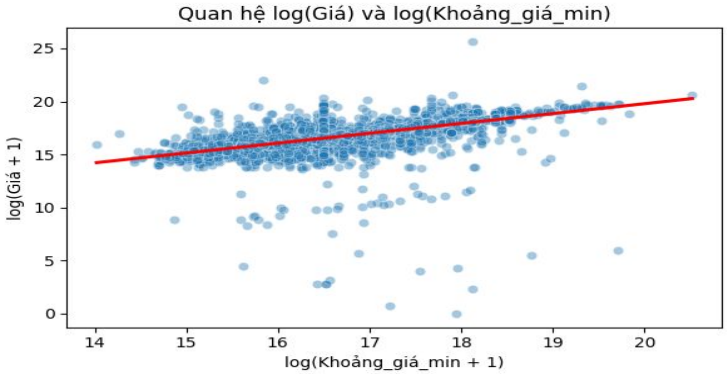


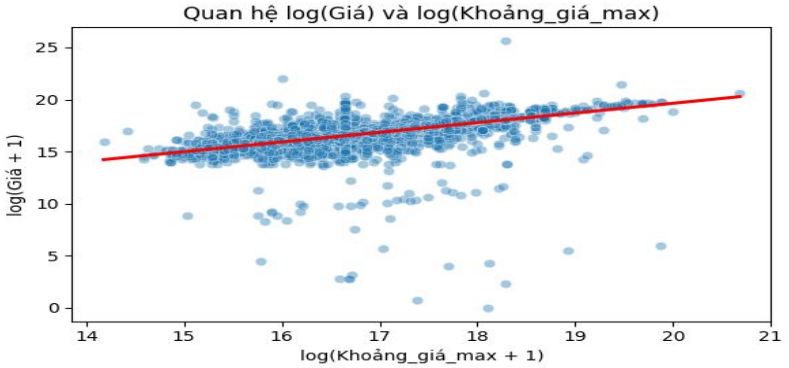


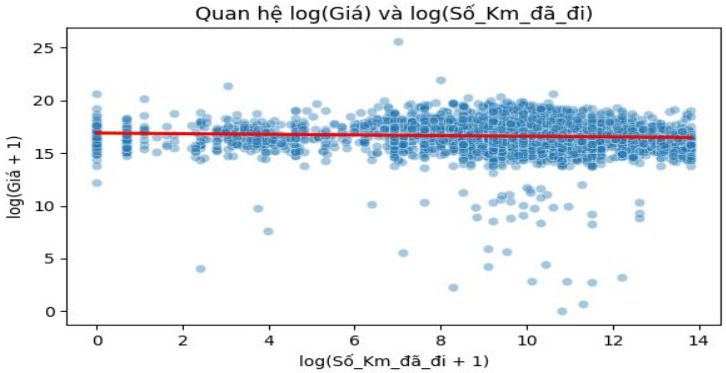


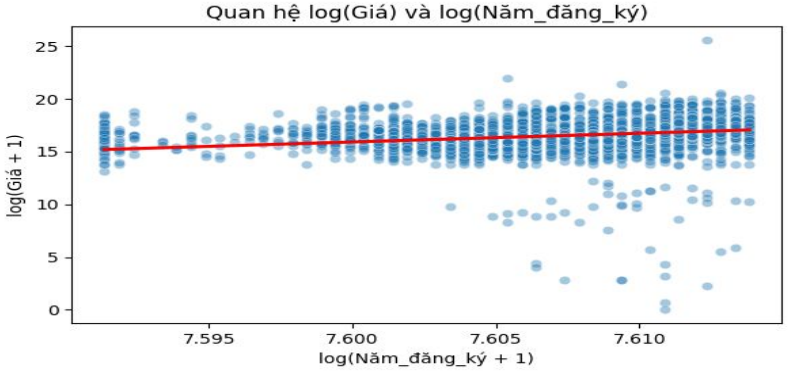
🛠️ EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

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



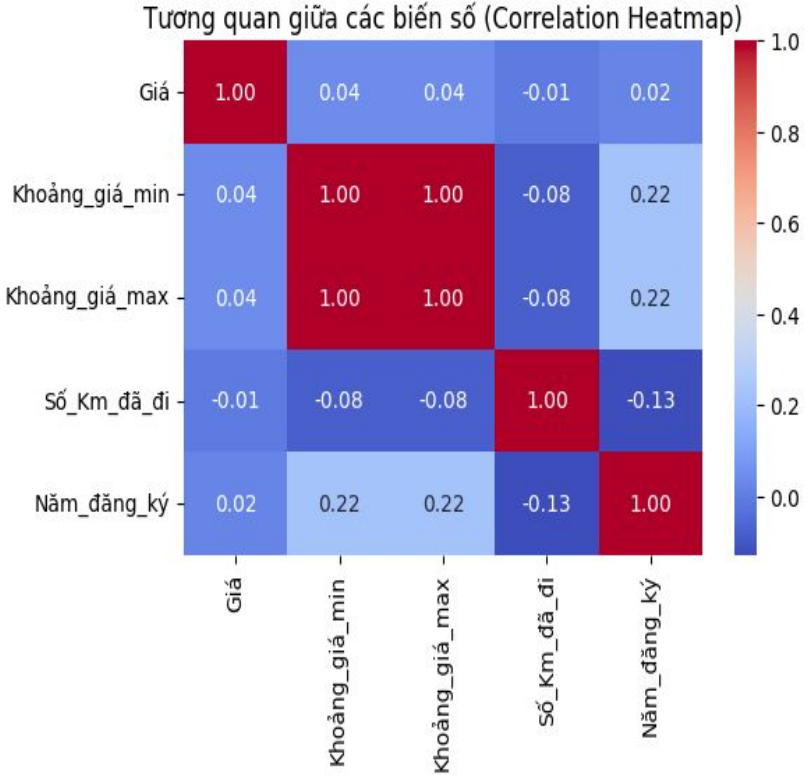






🛠️ EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

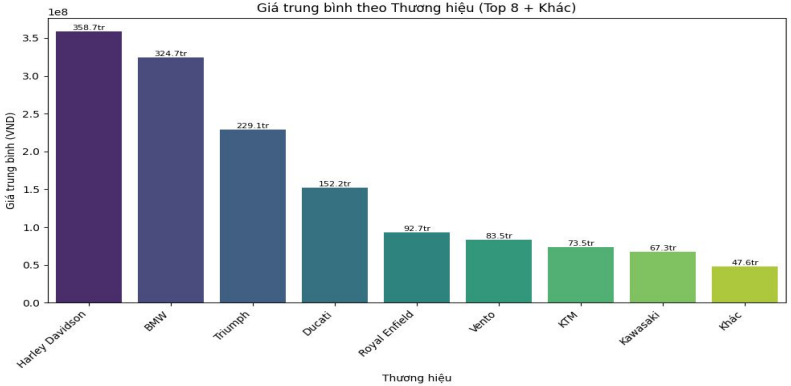
Các biến số (numeric variable)

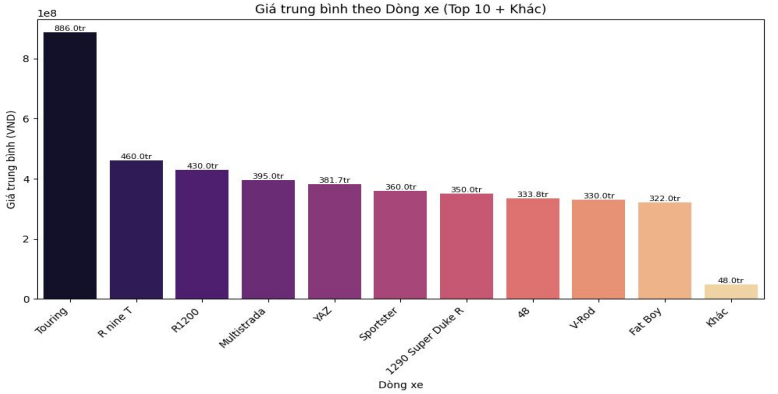


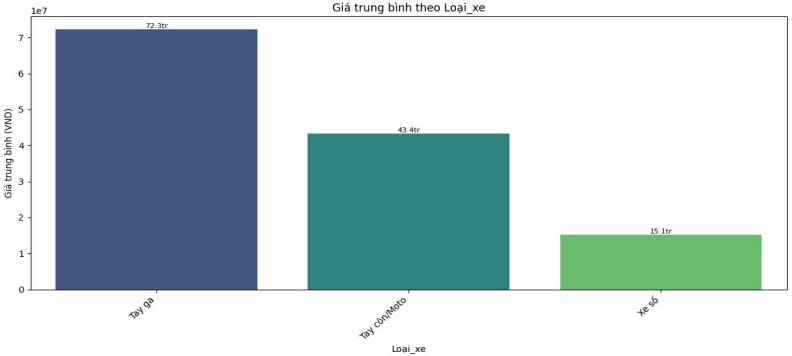
Cặp biến	Hệ số tương quan (r)	Nhận xét
Giá ↔ Khoảng_giá_min	0.04	Quan hệ rất yếu, gần như không tuyến tính
Giá ↔ Khoảng_giá_max	0.04	Tương tự, không có mối quan hệ tuyến tính rõ ràng
Giá ↔ Số_Km_đã_đi	-0.01	Cực kỳ yếu, số km gần như không ảnh hưởng đến giá
Giá ↔ Năm_đăng_ký	0.02	Gần như không có mối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

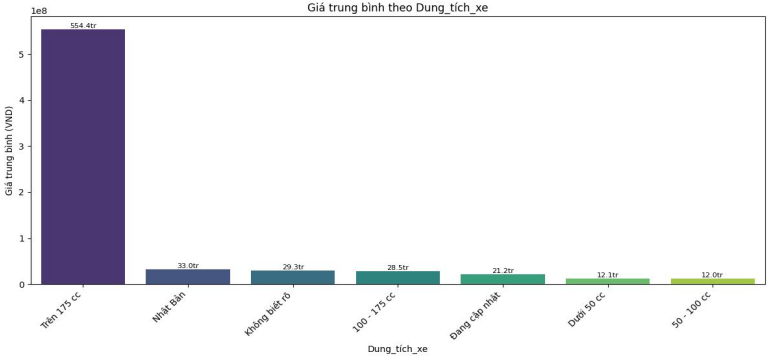
EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

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



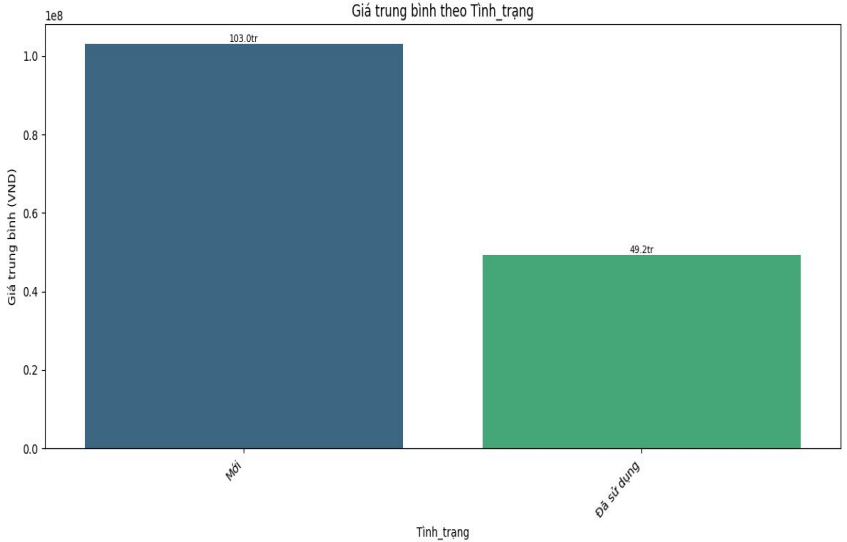


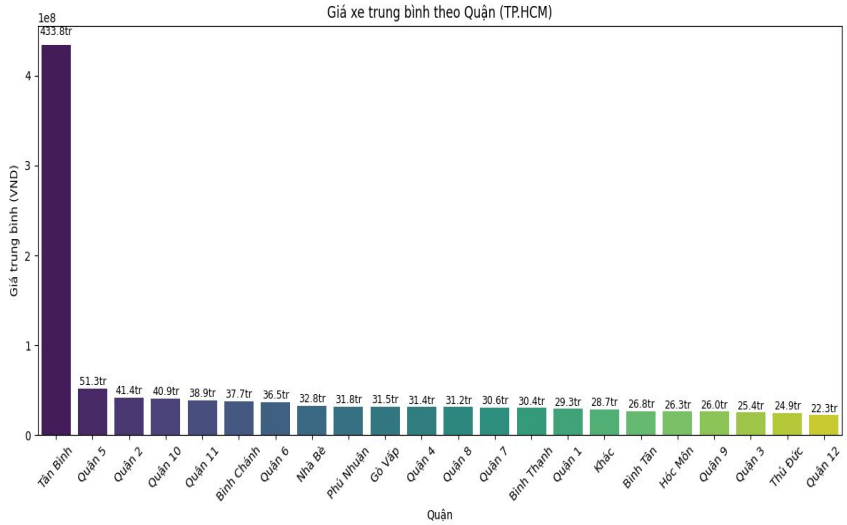




🛠️ EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

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 α = 0.05:
- Nếu *p-value* < 0.05 → có mối quan hệ
- Nếu *p-value* ≥ 0.05 → độc lập

Cặp biến	p-value	Kết luận
Thương_hiệu ↔ Dòng_xe	0	✅ Có mối quan hệ chặt chẽ
Thương_hiệu ↔ Loại_xe	~0	✅ 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

🛠️ EDA – PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

Ngưỡng phân chia giá cao/thấp: 16,500,000 VND

Xe GIÁ CAO (từ khóa nổi bật)

Xe GIÁ THẤP (từ khóa nổi bật)

Nhóm	Từ khóa đặc trưng	Hàm ý nội dung
Giá cao	*chính chủ, sang tên, công chứng, liên hệ, xem xe*	Nhấn mạnh uy tín, pháp lý và chất lượng → dễ làm tăng giá trị cảm nhận
Giá thấp	*bán xe, tờ đầy, zin, đầy đủ, cần bán*	Nhấn mạnh tình trạng sử dụng và nhu cầu bán gấp → thể hiện xu hướng giảm giá

🛠️ DATA INSIGHT

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

🏷️ **Yếu tố ảnh hưởng giá mạnh**: thương hiệu, dung tích, tình trạng xe; các yếu tố kỹ thuật khác (năm đăng ký, số km) tác động yếu.

🏍️ **Phân khúc cao cấp** (Harley, BMW, Triumph, Touring…) có giá vượt trội, tạo ranh giới rõ rệt giữa xe phổ thông & xe sang.

📊 **Xe tay ga & phân khối lớn**: biên độ giá rộng → phù hợp mô hình dự đoán & phát hiện bất thường.

📍 **Giá theo quận** chủ yếu 30–50 triệu; vượt xa vùng này có thể là dấu hiệu bất thường hoặc thị trường riêng.

✏️ **Hành vi người bán từ mô tả**:

- Giá cao → mô tả chi tiết, đáng tin cậy
- Giá thấp → mô tả ngắn, nhấn mạnh bán nhanh

🤖 **Ứng dụng**: sử dụng các yếu tố này để huấn luyện mô hình dự đoán giá hợp lý và phát hiện tin bất thường (giá ảo, nhập sai, gian lận).

🛠️ DATA PREPARATION

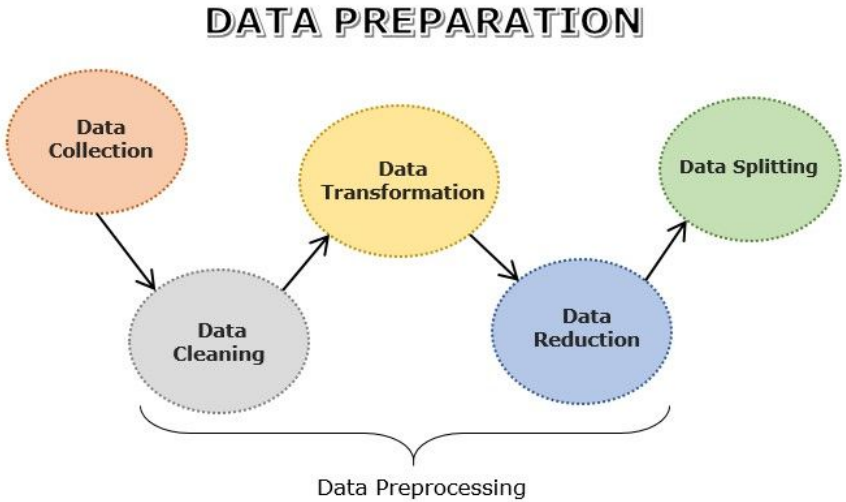
🧹 **Làm sạch dữ liệu**: loại bỏ giá trị null, bản ghi trùng lặp, ngoại lệ và các cột không cần thiết.

✏️ **Xử lý dữ liệu văn bản**: chuẩn hóa, làm sạch text để sử dụng cho mô hình.

🔢 **Mã hóa biến phân loại**: áp dụng Label Encoding cho các cột categorical.

⚙️ **Tạo đặc trưng mới**: ví dụ tuổi xe, giá/km, quận để tăng hiệu quả mô hình.

📊 **Chia tập dữ liệu**: phân tách **train/test** để huấn luyện và đánh giá mô hình.



🛠️ 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)

ML truyền thống

Mô hình	Ưu điểm nổi bật	Giải thích & Ứng dụng
🌲 **Random Forest**	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.
🚀 **Gradient Boosting**	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.
🐱 **CatBoost**	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.
⚡ **XGBoost**	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.
💡 **LightGBM**	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.
📉 **Ridge Regression**	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.

🛠️ MODELING (Regression)

Pyspark

Mô hình	Ưu điểm	Giải thích & Ứng dụng
🌲 **Random Forest**	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.
🚀 **Gradient Boosting (GBTRegressor)**	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 **giá xe**; hiệu quả hơn RF nếu dữ liệu đã được xử lý kỹ.
📉 **Ridge Regression**	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.

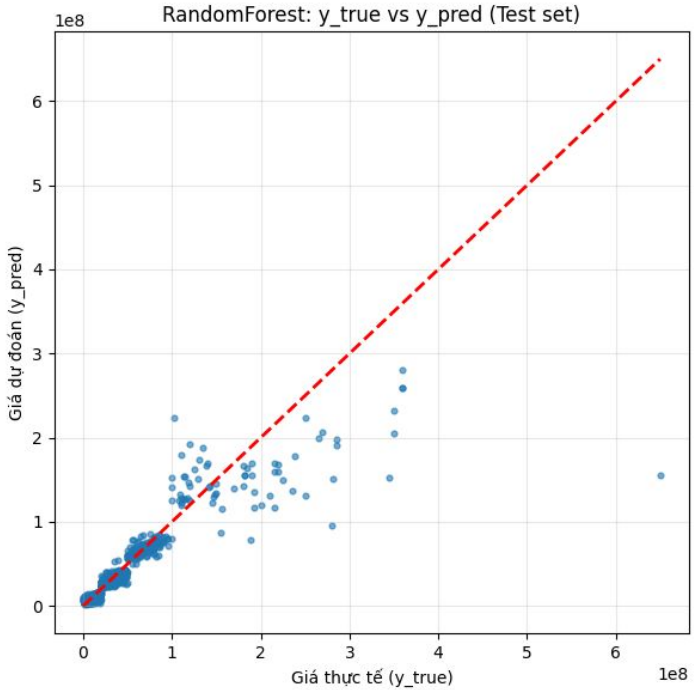
→ **KẾT QUẢ ĐÁNH GIÁ:** Dựa trên giá thật (không biến đổi log)

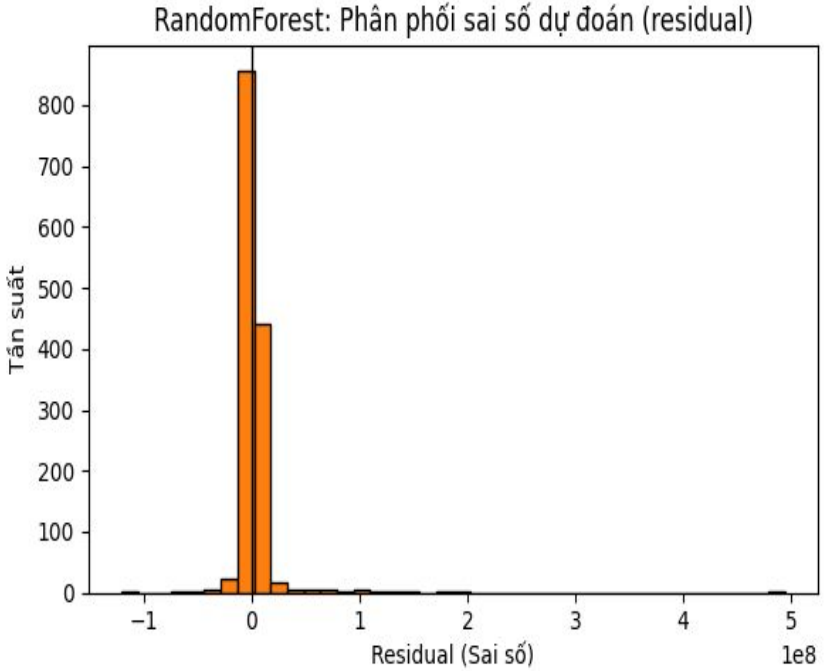
Pyspark

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

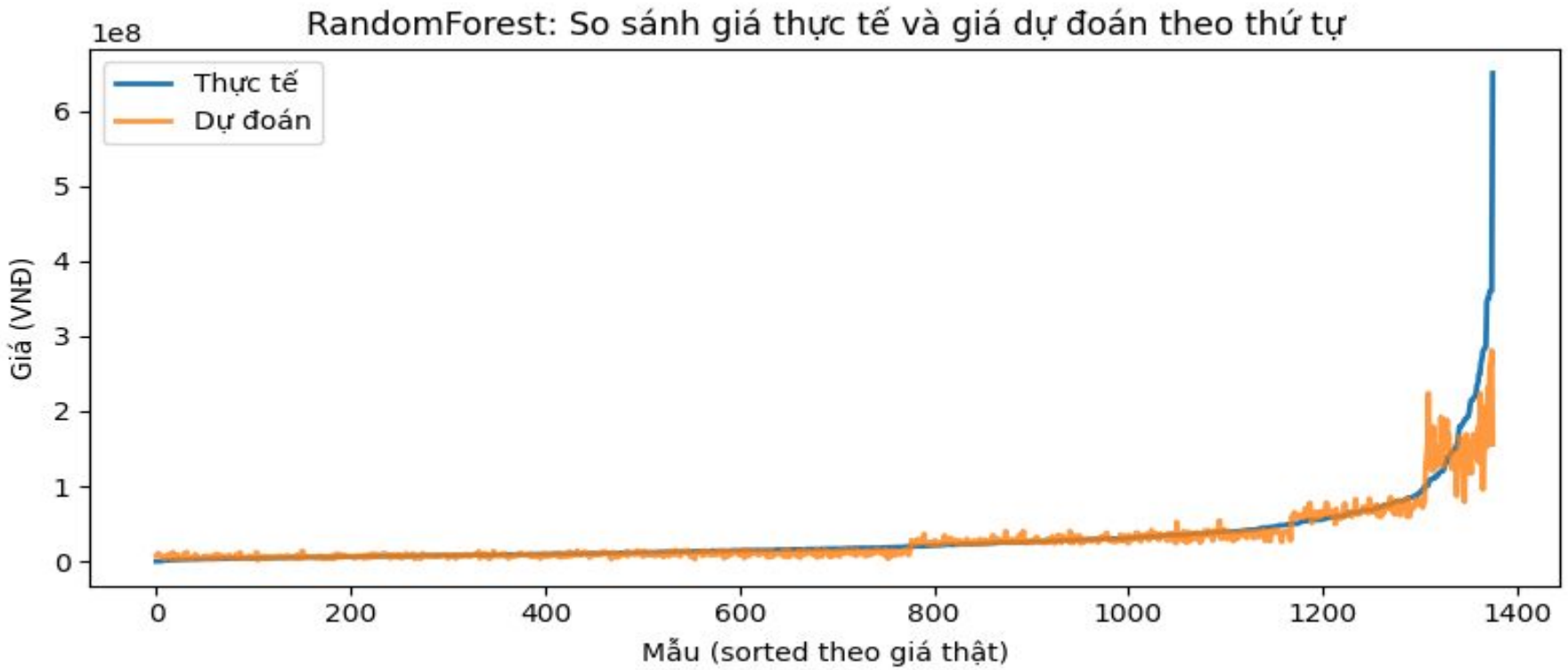
→ KẾT QUẢ ĐÁNH GIÁ:

Pyspark





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



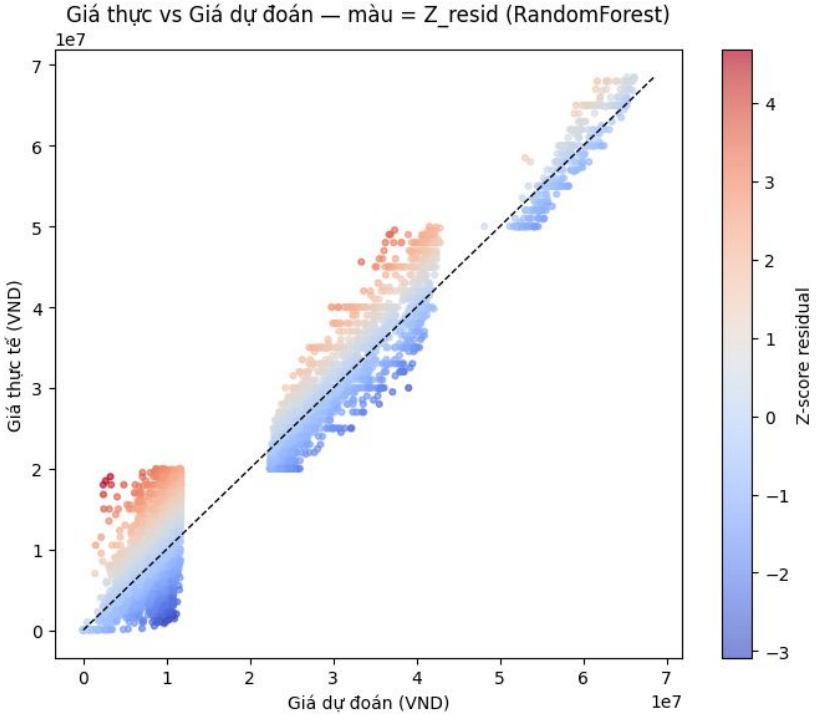
→ KẾT QUẢ ĐÁNH GIÁ: Dựa trên giá thật (không biến đổi log)

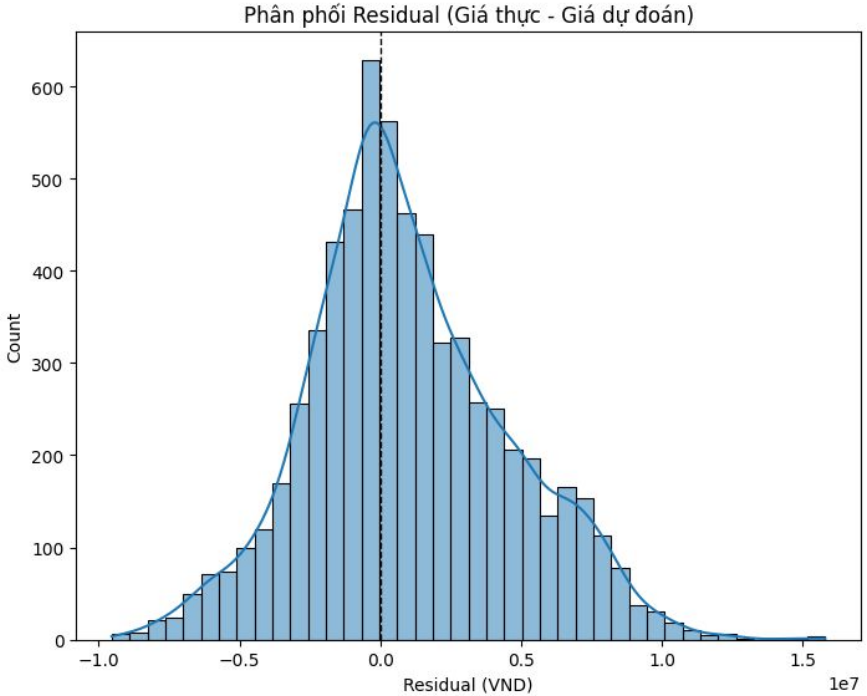
ML truyền thống

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

→ KẾT QUẢ ĐÁNH GIÁ:

ML truyền thống





📊 So sánh hiệu năng giữa hai môi trường

Tiêu chí	ML truyền thống	PySpark
Độ chính xác (R^2)	Cao: ~0.86–0.89 nhờ tinh chỉnh & boosting mạnh	Ổn định: ~0.8, phù hợp dữ liệu lớn
Mô hình hiệu quả nhất	Random Forest: cân bằng độ chính xác & thời gian huấn luyện	Random Forest: ổn định trên dữ liệu phân tán
Thời gian & hiệu suất	Nhanh trên tập nhỏ, nhưng tốn bộ nhớ khi mở rộng	Tối ưu cho dữ liệu lớn, hỗ trợ phân tán & song song hóa
Các mô hình khác	LightGBM, XGBoost: nhanh, chính xác, phù hợp local/batch nhỏ	Hai mô hình còn lại thì chưa phù hợp với yêu cầu

🔹 **Nhận xét tổng quan**

- ➢ ML truyền thống cho độ chính xác cao hơn trên dữ liệu nhỏ và khả năng tinh chỉnh mạnh.
- ➢ PySpark ổn định trên dữ liệu lớn, tối ưu hiệu suất phân tán.
- ➢ Random Forest là lựa chọn cân bằng giữa chính xác và thời gian huấn luyện trong cả hai môi trường.
- ➢ Các mô hình boosting (LightGBM, XGBoost) linh hoạt, nhanh và chính xác → phù hợp cho chạy local hoặc batch nhỏ.

Phát hiện bất thường (Anomaly Detection)

→ 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

🚨 Quy luật phát hiện giá bất thường

Sử dụng ML truyền thống kết hợp với phương pháp xử lý bất thường

Nguyên tắc chung

- So sánh Giá thực với Giá dự đoán (Random Forest).
- Kiểm tra tin đăng khác biệt so với thị trường (LOF – Local Outlier Factor).
- Kết hợp giá và mẫu tin để đánh giá bất thường.

Cách tính điểm

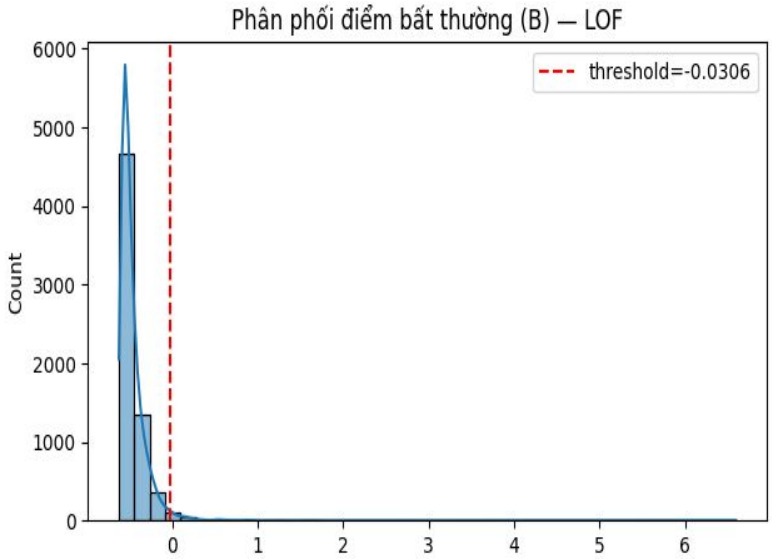
Điểm	Mô tả
A-score (Z-score)	Mức độ lệch giữa Giá thực ↔ Giá dự đoán
B-score (LOF)	Mức độ khác biệt của tin đăng so với dữ liệu chung
Abnormal_score	0.3×A + 0.7×B → 0–100 (càng cao → càng bất thường)

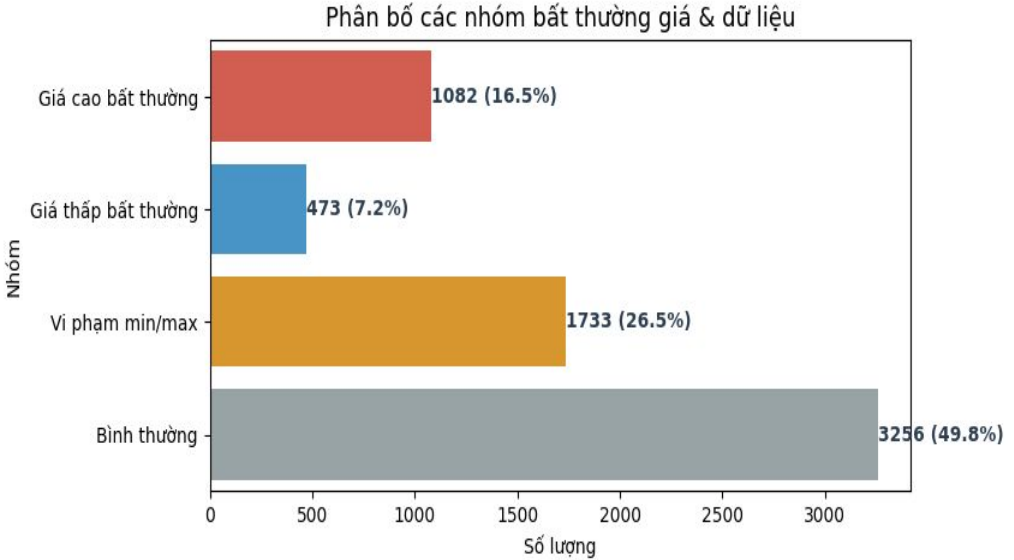
Khi nào xem là bất thường?

- **Giá cao bất thường**: Z ≥ +3 hoặc Giá thực > Giá dự đoán +40%
- **Giá thấp bất thường**: Z ≤ –3 hoặc Giá thực < Giá dự đoán –40%
- **Vi phạm min/max**: Giá nằm ngoài khung min–max ±15%
- **LOF B_flag = 1** → tin có đặc điểm lạ, ưu tiên kiểm tra

Evaluation & Comparison

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



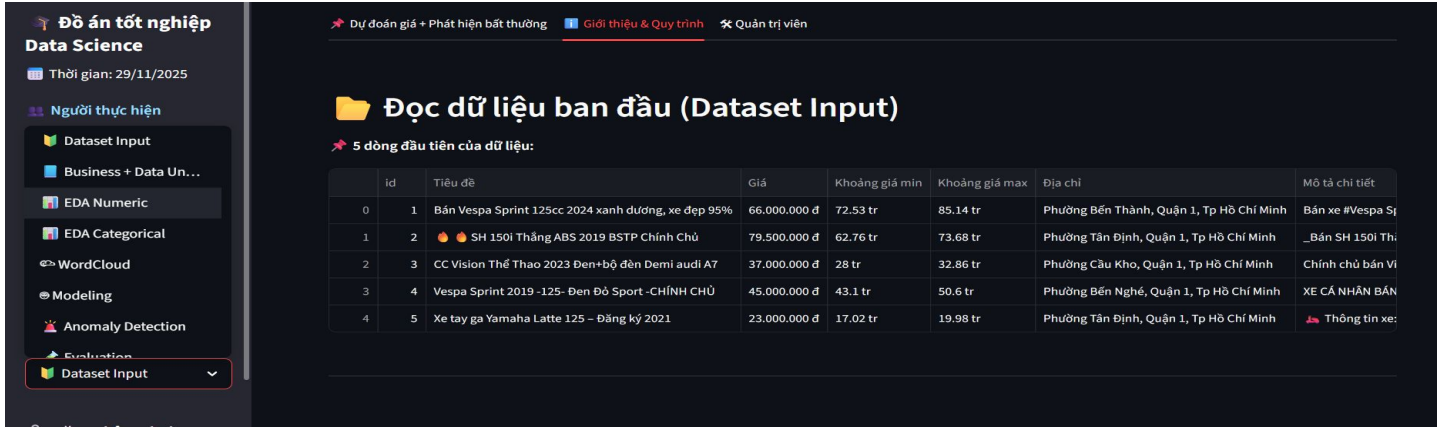


DEPLOYMENT

https://dudoangiaxevaphathienbatthuong-vha-nmxb.streamlit.app/

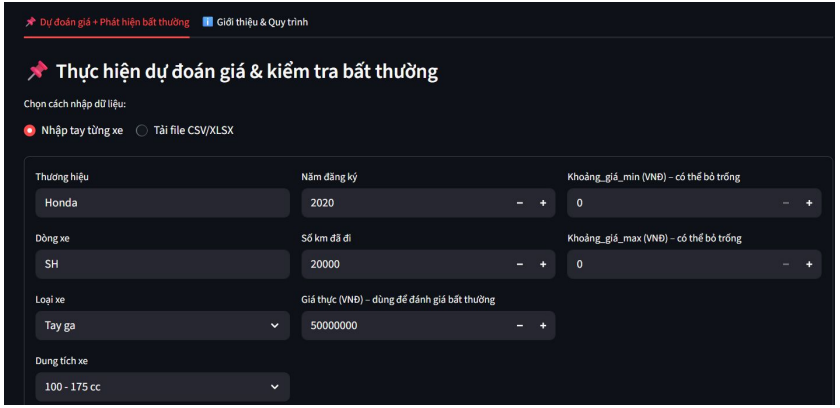
DEPLOYMENT

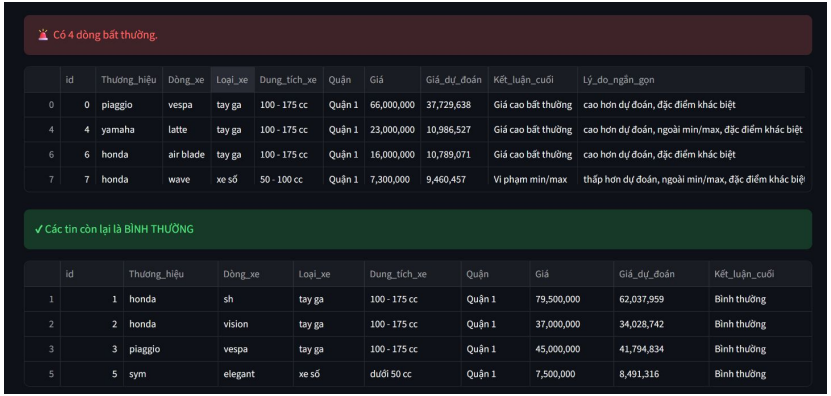
Trang Giới thiệu, Quy trình



DEPLOYMENT

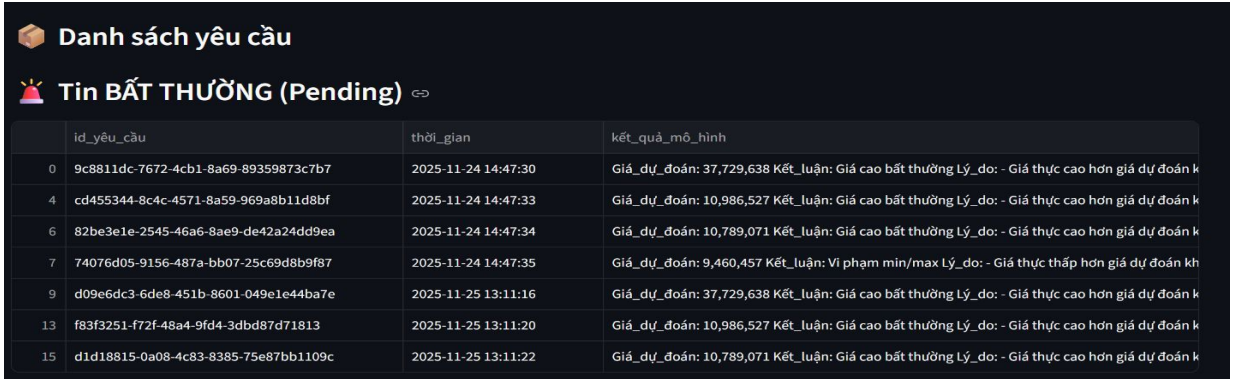
Về phía người đăng tin





DEPLOYMENT

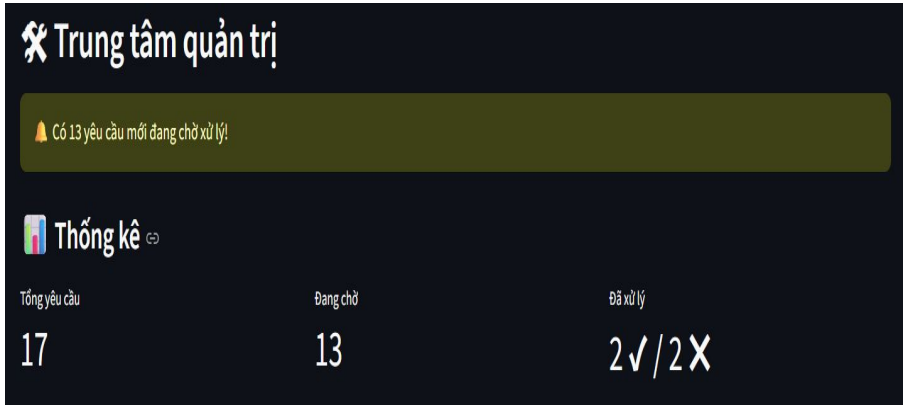
Về phía admin (người quản lý)



Danh sách yêu cầu

Tin BẤT THƯỜNG (Pending)

	id_yêu_cầu	thời_gian	kết_quả_mô_hình
0	9c8811dc-7672-4cb1-8a69-89359873c7b7	2025-11-24 14:47:30	Giá_dự_đoán: 37,729,638 Kết_luận: Giá cao bất thường Lý_do: - Giá thực cao hơn giá dự đoán k
4	cd455344-8c4c-4571-8a59-969a8b11d8bf	2025-11-24 14:47:33	Giá_dự_đoán: 10,986,527 Kết_luận: Giá cao bất thường Lý_do: - Giá thực cao hơn giá dự đoán k
6	82be3e1e-2545-46a6-8ae9-de42a24dd9ea	2025-11-24 14:47:34	Giá_dự_đoán: 10,789,071 Kết_luận: Giá cao bất thường Lý_do: - Giá thực cao hơn giá dự đoán k
7	74076d05-9156-487a-bb07-25c69d8b9f87	2025-11-24 14:47:35	Giá_dự_đoán: 9,460,457 Kết_luận: Vi phạm min/max Lý_do: - Giá thực thấp hơn giá dự đoán kh
9	d09e6dc3-6de8-451b-8601-049e1e44ba7e	2025-11-25 13:11:16	Giá_dự_đoán: 37,729,638 Kết_luận: Giá cao bất thường Lý_do: - Giá thực cao hơn giá dự đoán k
13	f83f3251-f72f-48a4-9fd4-3dbd87d71813	2025-11-25 13:11:20	Giá_dự_đoán: 10,986,527 Kết_luận: Giá cao bất thường Lý_do: - Giá thực cao hơn giá dự đoán k
15	d1d18815-0a08-4c83-8385-75e87bb1109c	2025-11-25 13:11:22	Giá_dự_đoán: 10,789,071 Kết_luận: Giá cao bất thường Lý_do: - Giá thực cao hơn giá dự đoán k



📊 KẾT QUẢ & INSIGHT

🏁 Kết quả chính

- Xây dựng thành công hệ thống dự đoán giá xe máy cũ và phát hiện bất thường từ dữ liệu thực.
- RandomForest cho kết quả dự đoán ổn định và phù hợp với bài toán định giá.
- LOF phát hiện hiệu quả các tin có giá lệch chuẩn hoặc đặc điểm khác thường trong mô tả.
- Hoàn thiện web-app trực quan, giúp người dùng tự định giá và hỗ trợ đội kiểm duyệt nhanh hơn.

🚀 Hướng cải thiện

- Nâng cấp mô hình xử lý văn bản: TF-IDF → BERT để hiểu mô tả chính xác hơn.
- Xây dựng mô hình min/max động, thay vì sử dụng giới hạn cố định ±15%.
- Phát triển API để tích hợp vào hệ thống thực tế ở quy mô lớn.
- Tối ưu UI/UX và cải thiện tốc độ xử lý khi tải file dung lượng lớn.

NHÓM THỰC HIỆN

VÕ THỊ HOÀNG ANH

🎓 ThS. Công nghệ Sinh học
📧 Email: anhvo.bio@gmail.com
📱 SĐT: 0935 719 426
🐙 GitHub: https://github.com/anhvobio
💼 Nội dung truyền thông

🛠️ **Nhiệm vụ đã thực hiện:** Xây dựng mô hình dự đoán giá; soạn bài thuyết trình và viết README.md hướng dẫn sử dụng ứng dụng cho người dùng; cùng phát triển ứng dụng Streamlit.

🎯 **Kết quả & Điều học được:** Qua quá trình làm project, Hoàng Anh hiểu rõ hơn từng bước, phương pháp và ý nghĩa trong toàn bộ quy trình Data Science. Đồng thời biết cách thực hiện trọn vẹn các giai đoạn từ xử lý tiền dữ liệu, xây dựng mô hình đến triển khai giao diện người dùng bằng GUI Streamlit.

NGUYỄN MAI XUÂN BÁCH

🎓 CN. Kỹ Thuật Điều Khiển Và Tự Động Hóa
📧 Email: bachxdn@gmail.com
📱 SĐT: 0982 308 974
🐙 GitHub: https://github.com/BachNguyen2kx
💼 Hỗ trợ kỹ thuật

🛠️ **Nhiệm vụ đã thực hiện:** Khám phá & xử lý dữ liệu, phát hiện bất thường; kiểm tra code và viết README.md cho final project; cùng tham gia xây dựng ứng dụng Streamlit.

🎯 **Kết quả & Điều học được:** Bách nhận ra tầm quan trọng của việc hiểu sâu dữ liệu và lựa chọn mô hình phù hợp — chỉ cần sai vài cột hoặc chọn sai mô hình là toàn bộ kết quả có thể lệch đi.

Việc triển khai ứng dụng lên Streamlit Cloud và Google Sheets mang lại kinh nghiệm thực tế về xây dựng giải pháp end-to-end, không chỉ dừng lại ở việc xây dựng mô hình.

Bên cạnh đó, Bách học được cách quản lý trạng thái, xử lý lỗi, thiết kế UI/UX và viết code sao cho người dùng không rành kỹ thuật vẫn có thể thao tác dễ dàng.