EV vs ICE Efficiency Modeling — Explanatory Narrative (No Figures)

# Executive Summary

This narrative explains the end‑to‑end approach used to compare Electric Vehicles (EV) and Internal Combustion Engine (ICE) vehicles, focusing on how the data was prepared, how models were trained and validated for each group separately, how efficiency was defined, and why the enhanced (feature‑engineered) pipeline improves stability and accuracy over the baseline. On EVs, average cross‑validated MAE changed by -2.4% after feature engineering. On ICE, average cross‑validated MAE changed by -8.2% after feature engineering. Cross‑validation variance reduced (negative % means more stable): EV 23.9% | ICE 35.5%. Average test R² changed by EV +0.116 and ICE +0.079.

# 1. Problem Statement & Scope

We were asked to evaluate EV and ICE separately, identify the best three models for each, and compare models by their ability to predict an efficiency signal. The original dataset was split into EV‑only and ICE‑only tables. We then created two workflows: a baseline pipeline without feature engineering and an enhanced pipeline that applies feature engineering before model training.

# 2. Data & Efficiency Definition

Key variables considered include CO₂ emission, cost per km, energy storage, mileage\_km, acceleration, torque\_nm, lifespan\_years, and maintenance cost. Efficiency is computed as mileage divided by energy consumption (km per unit energy). Treating EV and ICE separately prevents mixing fundamentally different drivetrain dynamics and avoids leakage between groups.

The EV table and ICE table were each cleaned, with any missing or inconsistent values handled prior to modeling. Where necessary, skewed variables were transformed and continuous features were scaled within the enhanced pipeline to stabilize model behavior.

# 3. Modeling Approach

We trained a family of regressors on the EV subset and the ICE subset independently. Model quality was assessed with k‑fold cross‑validation. We prioritized two complementary metrics: R² (explained variance) and MAE (absolute error), using MAE to favor robust, unit‑consistent error comparisons and R² to gauge explanatory power. Models were ranked primarily by lower MAE and secondarily by higher R².

Baseline pipeline: raw features feed into each model without engineered features. Enhanced pipeline: a preprocessing stage applies transformations and feature selection before modeling. This reduces noise, normalizes scales, and can expose meaningful interactions, which collectively improve generalization and reduce the spread of CV scores.

# 4. What Feature Engineering Added

From the enhanced script, the feature engineering stack includes elements such as: ColumnTransformer, Interaction, Lasso, PolynomialFeatures, PowerTransformer, RFE, Ridge, StandardScaler, log1p. These steps standardize scales, optionally add interactions or non‑linear transforms, and select informative features.

For EV, the selected/retained features included: co2\_emissions\_g\_per\_km, normalized\_cost, acceleration\_0\_100\_kph\_sec, maintenance\_per\_year, torque\_squared, lifespan\_years, torque\_x\_lifespan.

For ICE, the selected/retained features included: cost\_x\_maintenance, log\_torque, maintenance\_per\_torque, normalized\_cost, power\_efficiency.

# 5. Models Evaluated & Why

We evaluated a compact yet diverse set of learning algorithms: linear models (Linear Regression, Ridge, Lasso) that establish a strong baseline and guard against overfitting via regularization; tree‑based ensembles (Random Forest, Gradient Boosting, XGBoost) that capture non‑linearities and interactions; and simpler trees for interpretability checks. This spread balances bias and variance and yields resilient top‑k picks.

The three best models per category were chosen by the composite ranking (MAE primary, R² secondary) on held‑out validation. For transparency, here are the top‑three names discovered in this run:

• EV (Enhanced) top‑3: Ridge Regression, Lasso Regression, Linear Regression.

• ICE (Enhanced) top‑3: Ridge Regression, Lasso Regression, Linear Regression.

• EV (Baseline) top‑3: Linear Regression, Gradient Boosting, Random Forest.

• ICE (Baseline) top‑3: Linear Regression, Random Forest, Gradient Boosting.

# 6. Key Findings

For EV (baseline), the best MAE came from Linear Regression (avg CV MAE ≈ 1932.7). For EV (enhanced), the best MAE came from Ridge Regression (avg CV MAE ≈ 1933.2). For ICE (baseline), the best MAE came from Linear Regression (avg CV MAE ≈ 6090.1). For ICE (enhanced), the best MAE came from Ridge Regression (avg CV MAE ≈ 5834.8).

EV average MAE: -2.4% (enhanced vs baseline). ICE average MAE: -8.2% (enhanced vs baseline). EV CV stability: +23.9% change in CV MAE stdev (negative is better). ICE CV stability: +35.5% change in CV MAE stdev (negative is better).

# 7. Interpretation & What It Means

Feature engineering primarily helps the algorithms focus on signal by dampening scale effects and isolating informative patterns. In practice, this yields more stable validation folds and narrows the error distribution across random splits. Linear‑family models benefit from standardized inputs and can outperform when the true relationships are near‑linear; tree‑based ensembles excel when interactions and thresholds dominate. The separate EV/ICE training captures the physics differences between electric drivetrains and combustion systems, which improves model fit and reduces cross‑group bias.

Negative test R² values can occur when the target is noisy or the test fold is small—MAE is therefore emphasized for robust model selection. As data volume grows or as domain features (drive cycle, ambient temperature, payload) are added, R² typically improves while MAE continues to decline.

# 8. Recommendations

• Operationalize the top EV and ICE models from the enhanced pipeline, where generalization is most stable.

• Log additional covariates (drive cycle, temperature, payload, tire type) and retrain quarterly to catch drift.

• Couple efficiency predictions to a cost‑per‑km scenario model using energy/fuel price curves to inform TCO decisions.

• For explainability, retain SHAP/feature importance tracking in production to monitor changes in drivers over time.

# 9. Reproducibility Notes

Re‑run `vehicle\_efficiency\_analysis.py` for the baseline and `enhanced\_vehicle\_efficiency\_analysis.py` for the engineered pipeline. Both scripts export EV/ICE splits, metrics JSON, rankings CSVs, and figures under `output/`. Use the enhanced pipeline for deployment candidates.