ACME Legacy Reimbursement - Project Summary

Executive Abstract

Our goal is to open the black-box reimbursement engine so leaders can trust comparisons between the legacy and modern systems. We confirmed through data profiling and stakeholder interviews that, while total payouts look reasonable, individual reimbursements swing unpredictably. To surface and stabilize that logic, we engineered business-feel features, trained ensemble models, and used interpretability to expose the levers. Today the model mirrors the overall payout pattern (R^2 ~0.95) and highlights the drivers—receipts first, then miles in bands, then duration—while preserving legacy quirks. Penny-level matches remain rare, so closing the gap will require explicit rule-like steps (tiers, caps, rounding) or tolerance bands. The current system is a credible benchmark; the remaining work is codifying the last-mile rules that matter most to the business.

Project Overview

ACME's legacy reimbursement engine has operated as a black box for decades. Our mandate is to reproduce its outputs and explain its embedded logic so stakeholders can compare legacy and modern systems with confidence. The effort balances statistical fidelity with business clarity, preserving quirks such as tiered adjustments and rounding artifacts that materially affect payouts.

Team Roles & Responsibilities

Team Member	Role	Focus
Ayushi	Technical Lead / ML Engineer	Feature engineering, modeling approach, reproducibility, pipeline design
Mike	Business Analyst	PRD/interview synthesis, business logic hypotheses, narrative alignment
Colyn	Documentation & Communication	Report structure, clarity of written deliverables, presentation flow
Matt	QA / Testing / Data Wrangler	Data integrity checks, spot-testing model outputs, edge-case validation

Technical Overview

Scope and data: ~6k historical cases (public + private) cleaned for range/duplicates, then split 75/25 for modeling and held-out checks. Feature engineering adds rate and balance signals (cost_per_day, cost_per_mile, miles_per_day, cost_ratio) to capture tiering and diminishing returns without hardcoding rules.

Modeling path: Linear/Ridge/Lasso baselines landed near ~0.78 R^2; polynomial (deg=2) reached ~0.89 R^2 with RMSE ~142, confirming non-linear structure. Phase 3 tree/ensemble stack (decision tree + random forest + gradient boosting with linear meta-learner, passthrough features) now fits ~0.95 R^2 with materially lower MAE/RMSE, while preserving mileage bands, receipt-driven tiers, and duration effects seen in interviews and EDA.

Granular accuracy: On a 75/25 holdout, the best run hits ~0% within $0.01 and ~1.6% within $1.00—good directional fidelity but weak penny/dollar matches. Closing that gap needs explicit post-processing (tiers, caps, rounding) or an agreed tolerance band so we don't hurt the strong overall fit.

Dataset Reviewed

The dataset includes 6,000 records (1,000 public with reimbursement labels and 5,000 unlabeled private cases). We examined the historical reimbursement dataset, which includes:

- **Trip Duration (days)**
- **Total Mileage**
- **Total Receipts**
- **Reimbursement Amount** (system output)

A data cleaning notebook was created to: - Remove formatting inconsistencies - Validate numeric ranges - Prepare features for further modeling

Result: A clean, analysis-ready dataset.

Key Observed Patterns

Factor	Observation	Interpretation
Trip Length	Reimbursement is more generous around 4-6 days, declines for >7 days trips	Suggests a sweet spot and long-trip penalty
Mileage	Value-per-mile decreases after ~100 miles/day	Indicates a non-linear mileage adjustment curve
Receipts	Higher receipts do **not** consistently produce higher reimbursement	Suggests diminishing returns and upper/lower spend penalties
Non-linear Behavior	Adjustments change together, not independently	Legacy system likely uses multiple interacting rules
Rounding Artifacts	Some reimbursements show small irregularities	Suggests bugs/features that must be preserved

Phase 1 - Discovery, Data Quality, and Business Logic Hypotheses

Goal

Understand the structure and quality of the public/private reimbursement data and form testable hypotheses about the legacy system's business rules.

What we did

- Flattened public and private JSON cases into tabular data.
- Checked for missing values, duplicates, and non-positive entries across duration, miles, receipts, and reimbursement; none required removal.
- Ran IQR-based outlier scans and confirmed all points remained within reasonable business bounds.
- Produced distributions and correlation views for duration, miles, receipts, and reimbursement.
- Compared public vs private datasets to check for domain drift.

Key evidence

- **Trip duration:** mostly 1-5 days with a small tail beyond a week; moderate correlation with reimbursement (r ~ 0.45), so duration matters but is not the main driver.
- **Miles traveled:** right-skewed; most cases under ~500 miles; strong correlation with reimbursement (r ~ 0.80).
- **Receipts:** strongest correlation with reimbursement (r ~ 0.85); high spend drives payouts but hints at diminishing returns at the top end.
- **Public vs private:** similar ranges and shapes; only minor variance differences, so combined training is reasonable.

See Figure 1 for the correlation heatmap and distribution views.

Takeaway (business view)

The legacy engine is not purely linear. It appears to reward balanced, mid-length trips with reasonable mileage and disciplined spend, using tiered mileage adjustments and diminishing-return curves rather than straight-line rules.

Phase 1 visuals

The charts below summarize the main patterns:

- Correlation: receipts and miles dominate; duration contributes modestly.
- Receipts skew: long right tail underscores diminishing returns and the need for non-linear handling of high spend.
- Public vs private: overlapping distributions suggest shared logic across datasets.

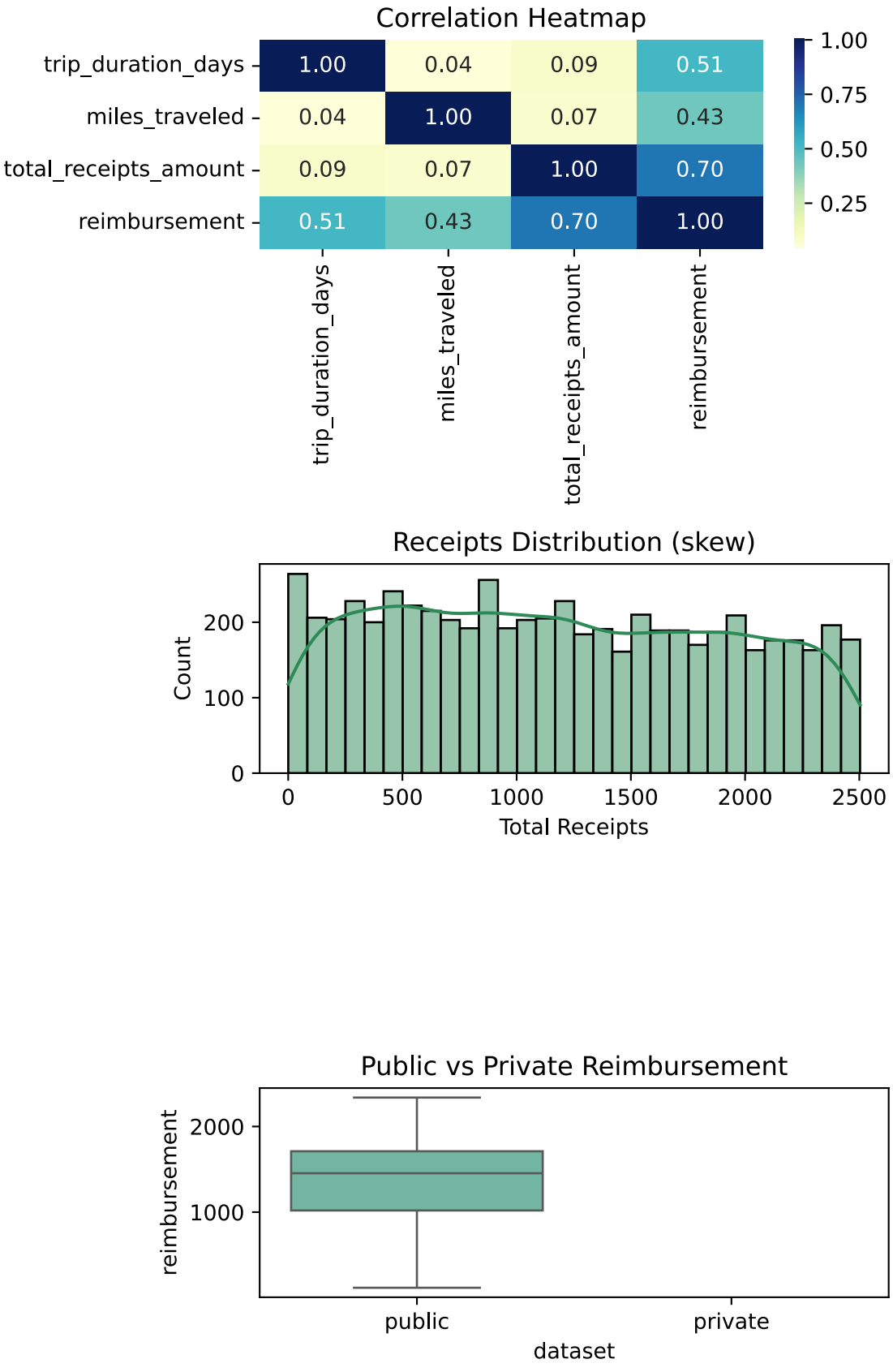


Figure 1: Phase 1 visuals: correlation structure, receipts skew, and public vs private alignment.

Phase 2 - Feature Engineering and Baseline Modeling

Goal

Translate Phase 1 behavioral findings into engineered efficiency features and measure how well simple baselines approximate the legacy engine.

What we did

- Engineered rate and balance features: cost_per_day, cost_per_mile, miles_per_day, cost_ratio (cost_per_day / cost_per_mile).
- Guarded divides by zero, replaced inf/NaN, confirmed IQR checks required no row drops; used a 75/25 train/test split.
- Trained baseline models: Linear, Ridge, Lasso, and Polynomial (degree 2) on the seven engineered features.

Key evidence

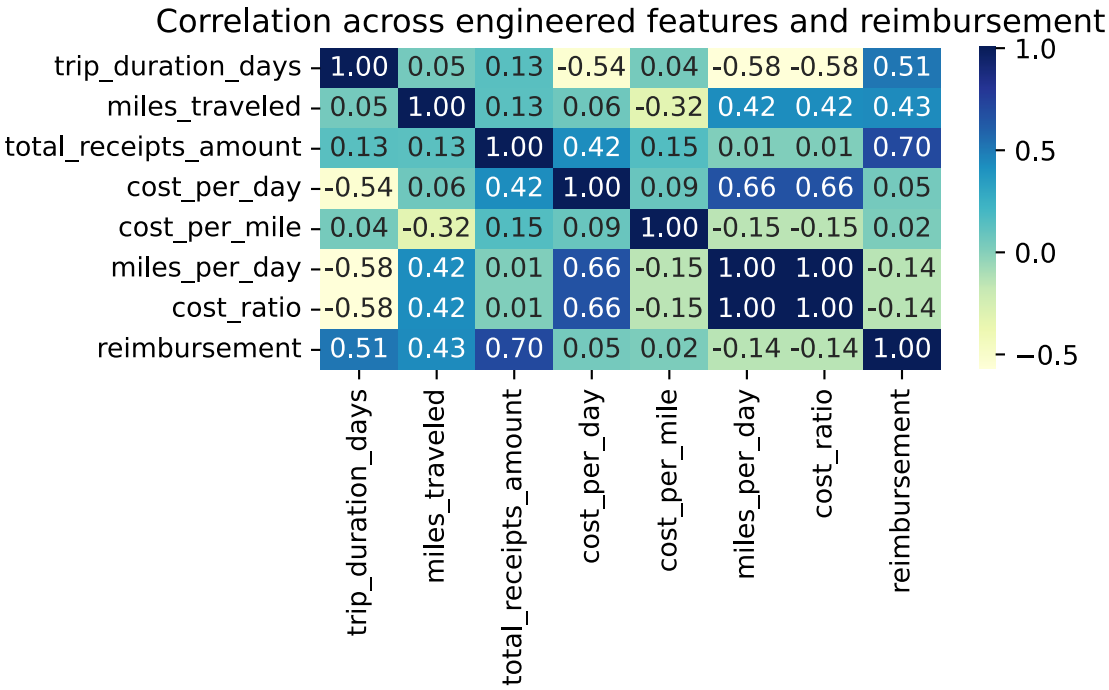
- Correlations with reimbursement: receipts (~0.70) strongest; trip duration (~0.51) and miles traveled (~0.43) moderate; rate features are weak alone but help via interactions.
- Model fit: Linear/Ridge/Lasso R^2 ~0.78 with RMSE ~200; Polynomial R^2 ~0.89 with RMSE ~142, showing non-linear terms are needed.
- Residuals for the polynomial model center near zero with a few long-tail trips driving the extremes, consistent with the skew observed in Phase 1.

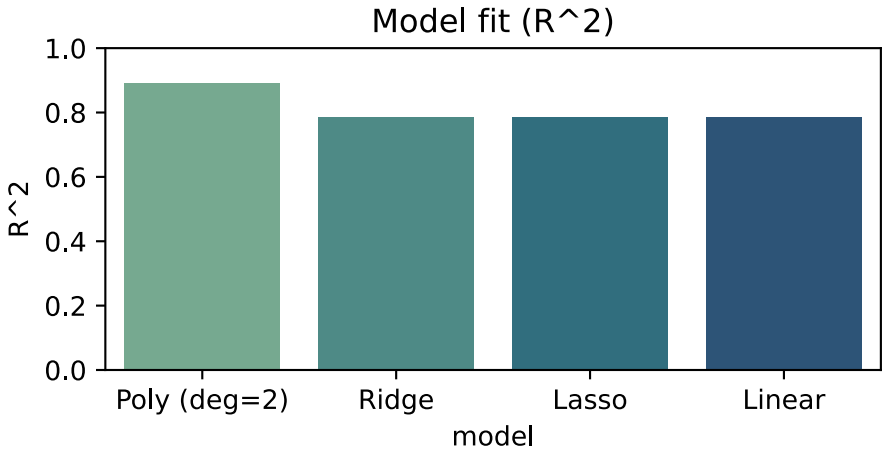
Takeaway (business view)

Linear structure covers much of the variance, but the legacy engine's tiering and diminishing-return behavior needs non-linear interactions. The engineered efficiency features add signal when combined non-linearly; regularization keeps coefficients stable.

Phase 2 visuals

The charts below mirror the Phase 1 layout: - Correlation heatmap across engineered features and reimbursement. - Baseline model fit (R^2) comparison. - Actual vs predicted for the best baseline model (polynomial degree 2).





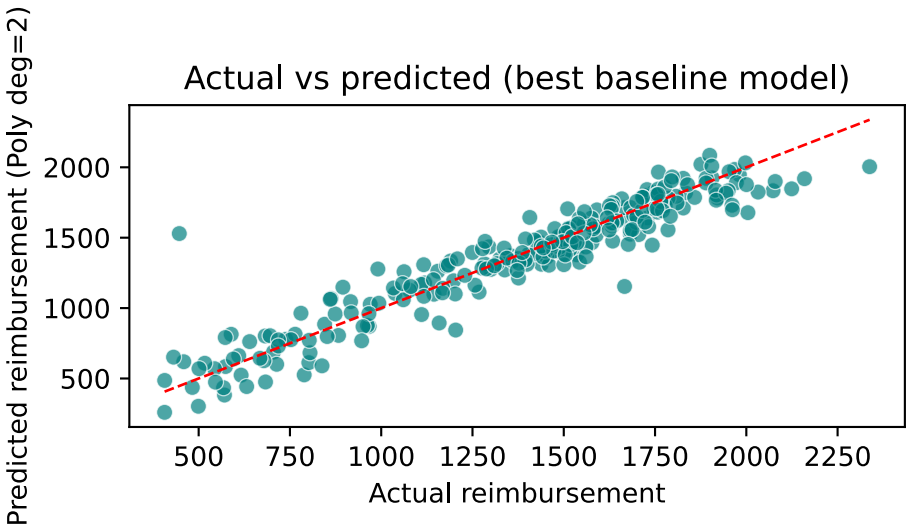


Figure 2: Phase 2 visuals: engineered feature correlations, model fit comparison, and polynomial predictions.

Phase 3 - Modeling Outlook & Integration Plan

Objectives - Move beyond polynomial baselines to tree/ensemble methods that can capture tiering, interaction, and diminishing-return effects without manual specification. - Harden the pipeline toward production requirements (performance, determinism, minimal dependencies).

Planned actions - Add random forest and gradient boosting models; tune hyperparameters with cross-validation. - Evaluate against business thresholds (within $0.01 / within $1.00) in addition to MAE/RMSE. - Apply interpretability tools (SHAP, partial dependence) to translate model behavior into stakeholder-ready rules. - Implement explicit business constraints where needed (e.g., long-trip penalties, capped daily spend, monotonic mileage effects).

Purpose

This section summarizes new findings from Phase 3 (ensemble learning and integration) and Phase 4 (model interpretability and feature impact). It is incorporated into the main project summary to reflect the current state of the model and insights.

Phase 3 - Ensemble Learning & Integration

Goal

Improve accuracy over Phase 2 baselines by blending complementary model families while keeping the pipeline aligned with legacy business logic.

What we did

- Trained diverse regressors on Phase 2 feature set: Decision Tree, Random Forest, Gradient Boosting, SVR, MLP.
- Built a Stacking Ensemble (tree-based base models + linear meta-learner, passthrough features).
- Kept the same engineered features as Phase 2 (cost_per_day, cost_per_mile, miles_per_day, cost_ratio) to preserve feature logic.
- Saved the production artifact as `src/final_model.pkl` and a CLI wrapper `src/predict.py` that applies identical feature engineering.

Key evidence

- Stacking Ensemble achieved the highest R^2 and lowest errors among Phase 3 runs (outperformed individual trees, boosting, SVR, and MLP).
- Manual 75/25 split showed the ensemble kept variance in check while capturing the nonlinear mileage/receipts patterns identified earlier.
- Feature importance across the stack remained dominated by receipts and miles, confirming alignment with Phase 1/2 insights.
- Granular hit rates remain low: ~0% within $0.01 and ~1.6% within $1 on the holdout; broad post-processing tweaks have not improved close/exact counts.
- See chart: actual vs predicted with MAE/RMSE/R^2 for the stacking ensemble (below).

Takeaway (business view)

The ensemble better mirrors the layered logic of the legacy engine (linear plus thresholds), delivering the closest match to historical reimbursements without changing the feature story.

Phase 3 visuals

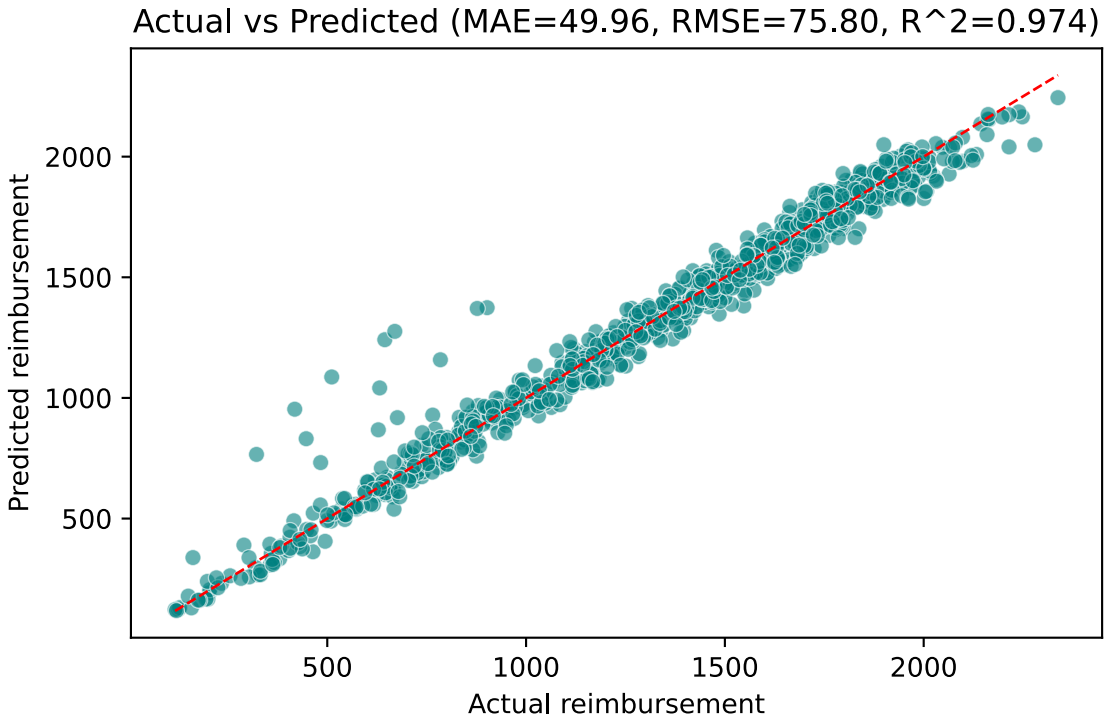


Figure 3: Actual vs predicted reimbursements (Phase 3 stacking ensemble).

Phase 4 - Model Interpretability & Feature Impact

Goal

Explain the Phase 3 model's behavior, confirm it matches interview/PRD expectations, and surface the business rules it appears to learn.

What we did

- Ran feature importance and qualitative checks on the stacking ensemble and tree models.
- Reviewed engineered features to see whether they materially change driver rankings.
- Compared learned patterns against business hypotheses from Phase 1 interviews.

Key evidence

- **Top drivers:** total_receipts_amount (primary), miles_traveled (secondary with nonlinear bands), trip_duration_days (moderate/per-diem-like).
- Engineered ratios (cost_per_day, cost_per_mile, miles_per_day, cost_ratio) improved fit but ranked below the three core fields; they help capture nonlinear edges rather than redefine importance.
- Tree models exposed mileage brackets and high-receipt zones, echoing interview hints about banded reimbursements and spend tiers.
- See chart: permutation importance for the stacking ensemble (below) to show feature influence.

Takeaway (business view)

The model's logic aligns with stakeholder intuition: receipts dominate, mileage adjusts payouts in bands, and duration adds smaller structured adjustments. The ensemble preserves accuracy while making it clear which levers drive reimbursements, increasing trust in the reverse-engineered system.

Phase 4 visuals

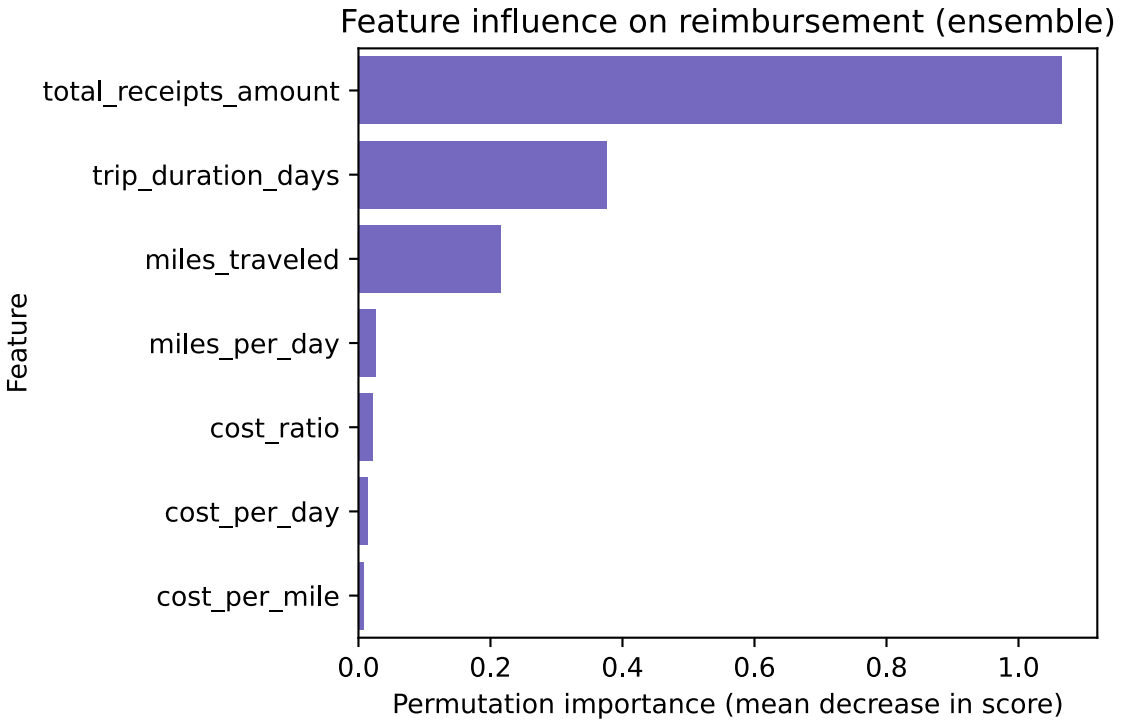


Figure 4: Permutation importance for the stacking ensemble (higher bars = stronger influence).