ACME Legacy Reimbursement - Project Summary

Species: **default species**

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

Executive Overview

We are reverse-engineering ACME's legacy reimbursement engine to match and explain its outputs. This report summarizes progress across three phases: initial discovery and behavioral hypotheses, feature engineering with baseline models, and correlation-driven insights to guide next modeling steps. The aim is to balance technical rigor with business clarity while preserving legacy artifacts (tiering, rounding quirks) that stakeholders care about.

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.

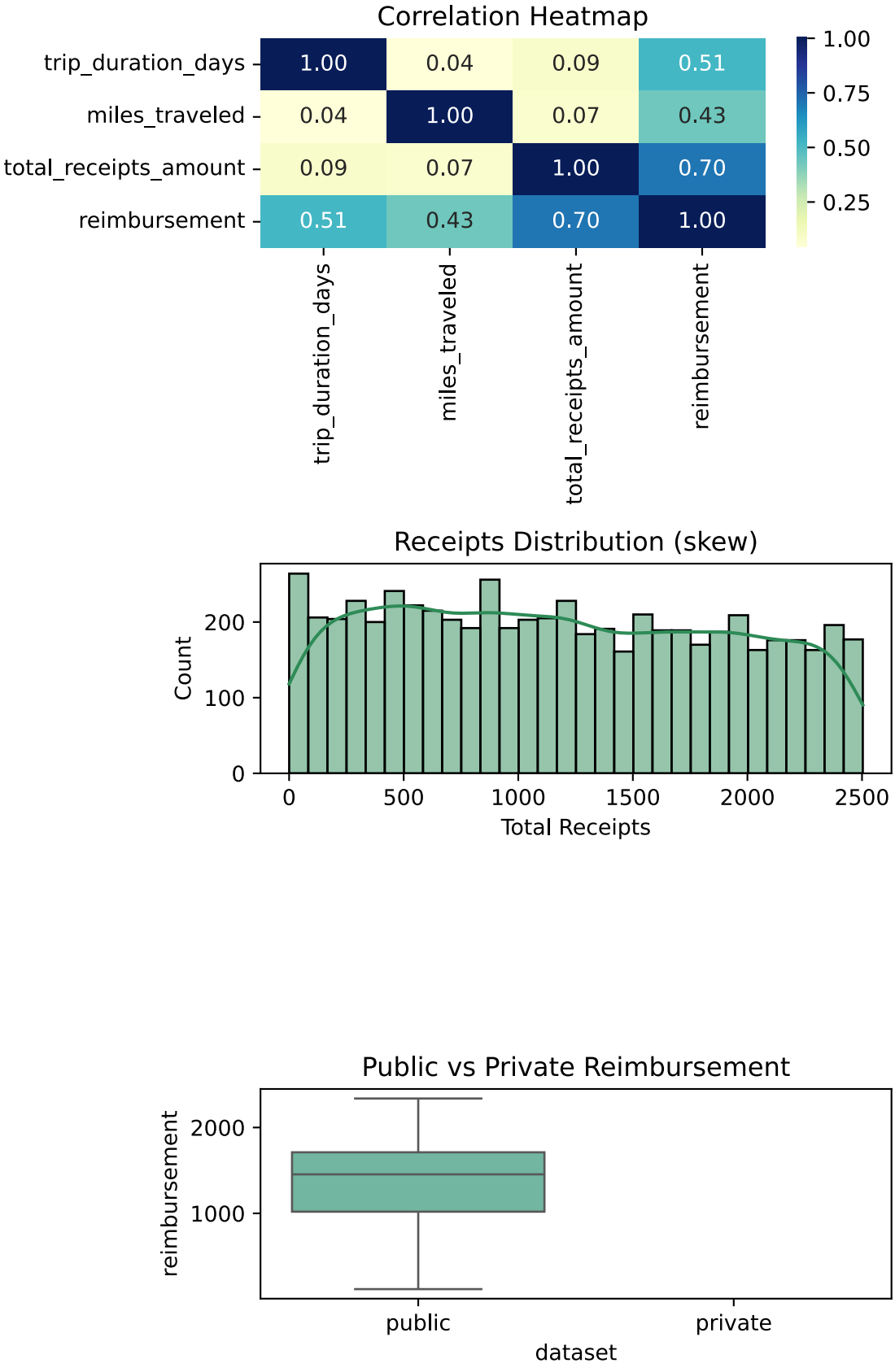

Figure 1: Phase 1 visuals: correlation structure, receipts skew, and public vs private alignment.

Phase 2 - Feature Engineering and Baseline Modeling

Feature design (behavioral intent) - cost_per_day: Lodging/meal intensity per day; flags expensive daily burn. - cost_per_mile: Spend per mile; highlights route efficiency. - miles_per_day: Pace of itinerary; proxies route type and overnight needs. - cost_ratio (cost_per_day / cost_per_mile): Balance between lodging vs. mileage costs; captures spend discipline.

Data handling - Guarded divides by zero, replaced inf/NaN, and retained rows after cleaning. IQR checks on original and derived features required no exclusions. Baseline split used a 75/25 train/test partition.

Models evaluated - Linear Regression (baseline interpretability). - Ridge and Lasso (regularized linear) to stabilize coefficients. - Polynomial Regression (degree 2) to capture curvature and interactions among duration, miles, receipts, and efficiency terms.

Findings - Polynomial regression delivered the strongest fit (highest R2, lowest error), reinforcing the need for non-linear terms to mimic tiered mileage and diminishing spend effects. - Ridge/Lasso performed close to the linear baseline with improved stability, suggesting modest collinearity but no severe overfitting. - Engineered efficiency features retained signal value after cleaning, supporting their inclusion in interaction-rich models. - The tapering and interaction patterns visible in Phase 1 visuals (skewed receipts, mileage tiering) motivate moving beyond linear models to methods that learn non-linear boundaries directly.

Risks and gaps - Single train/test split only; no cross-validation or holdout robustness yet. - Tree/ensemble models have not been run; hyperparameters remain at defaults. - Business thresholds (within $0.01 / within $1.00) not yet reported.

Phase 2 takeaway Linear structure explains much of the variance, but non-linear terms are needed to emulate the legacy system's tiering and diminishing-return behavior. Regularization helps stability; richer non-linear models are the logical next step.

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).

Next Steps and Recommendations

- Broaden models: add tree/ensemble approaches (random forest, gradient boosting) alongside polynomial baselines.
- Strengthen evaluation: introduce cross-validation and report MAE/RMSE plus business thresholds (percent within +/-$0.01 and +/-$1.00).
- Encode business effects: implement explicit long-trip penalties and spend-discipline rules (piecewise mileage rates, capped daily spend, monotonic constraints).
- Interpretability: apply SHAP/partial dependence to translate model behavior into stakeholder-friendly rules; document rounding/post-processing to preserve legacy artifacts.
- Production readiness: package the pipeline into the required 3-argument script, ensure <5s runtime, remove external dependencies, and fix seeds/versioning for deterministic outputs.