ACME Legacy Reimbursement - Project Summary

Domain: **Legacy Reimbursement Engine**

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. We balance 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 the legacy engine to match and explain its outputs. This report summarizes discovery and hypotheses, feature engineering with baseline models, and correlation-driven insights guiding the next modeling steps. The goal is to deliver an interpretable replica while preserving legacy artifacts (tiering, rounding) that stakeholders care about.

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

Data integrity: Public and private cases were flattened and validated; no missing values, duplicates, or non-positive entries. IQR scans across duration, miles, receipts, and reimbursements required no removals, and ranges align with business bounds.

Descriptive signals:

- Trip duration: Right-skewed; most trips 1–5 days, moderate correlation with reimbursement (0.45).
- Miles: Right-skewed; most under 500 miles, strong correlation (0.80) with reimbursement.
- Receipts: Positively skewed; strongest correlation (0.85) with reimbursement; high spend hints at diminishing returns.
- Reimbursement: Continuous with a long tail; driven by miles and receipts with duration as context.
- Public vs. private: Similar ranges and shapes; minor variance differences only, enabling combined modeling.

Business logic hypotheses:

- Sweet spot: 4–6 day trips; reduced marginal reimbursement beyond 7 days (long-trip penalty).

- Mileage taper: Value per mile declines past 100 miles/day, suggesting tiered mileage rates or efficiency checks.
- Spend discipline: Higher receipts do not guarantee higher reimbursement, implying diminishing returns and penalties for overspend/underspend.
- Interactions: Adjustments likely move together (duration, miles, receipts), indicating non-linear interactions rather than isolated linear effects.
- Rounding/quirks: Small payout artifacts should be preserved to match legacy behavior.

Visual insights (correlation heatmap, receipts skew, public vs. private comparison):
- Receipts and miles dominate; duration is secondary.
- Right-tail receipts skew underscores the need for non-linear handling of high spend.
- Public/private overlap shows the drivers generalize across data sources.

Phase 1 takeaway: The system is not purely linear. It rewards balanced, mid-length trips with reasonable mileage and disciplined spend, likely via tiered adjustments and diminishing-return curves. Clean, consistent data supports feature engineering around efficiency, tapering, and interactions.

Phase 2 - Feature Engineering and Baseline Modeling

Feature 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`: Balance between lodging and mileage costs; captures spend discipline.

Data handling: Guarded divide-by-zero, replaced inf/NaN, retained rows after cleaning; IQR checks required no exclusions. Used a 75/25 train/test split as a baseline.

Models evaluated: Linear Regression (baseline), Ridge/Lasso (stabilize coefficients), Polynomial Regression (degree 2) to capture curvature and interactions.

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 tracked close to the linear baseline with improved stability, indicating modest collinearity but no severe overfitting.
- Engineered efficiency features retained signal value; tapering and interaction patterns from Phase 1 justify moving beyond linear models.

Gaps: Single split (no cross-validation), no tree/ensemble models yet, business thresholds (within $0.01 / within $1.00) not reported.

Phase 2 takeaway: Linear structure explains much of the variance, but non-linear terms are needed to emulate tiering and diminishing returns. Regularization helps stability; richer non-linear models are the next step.

Phase 3 - Integration and Advanced Modeling Outlook

Objectives: Move beyond polynomial baselines to tree/ensemble methods that capture tiering, interactions, and diminishing returns without manual specification; harden the pipeline for production (performance, determinism, minimal dependencies).

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

Next Steps and Recommendations

- Broaden models: tree/ensemble approaches alongside polynomial baselines.
- Strengthen evaluation: cross-validation and business thresholds (percent within +/-0.01 $\wedge \pm 1.00$).
- Encode business effects: explicit long-trip penalties and spend-discipline rules (piecewise mileage rates, capped daily spend, monotonic constraints).
- Interpretability: SHAP/partial dependence; 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.