Reverse Engineering ACME's Legacy Reimbursement Engine Using Machine Learning, Statistical Analysis, & Business Reasoning with High Accuracy & Interpretability.

2025-12-08

Phase 3: Nonlinear Modeling, Ensemble Learning, and System Integration

∼ **Aim:**
Advance beyond Phase 2 polynomial baselines by testing nonlinear and ensemble modeling methods capable of capturing ACME's tiering, threshold, and diminishing-return reimbursement behavior. Phase 3 integrates PRD- and interview-driven engineered logic features, evaluates multiple model families, and selects a calibrated stacking ensemble that most closely replicates the 60-year-old legacy engine.

---

1. Data Preparation & PRD-Driven Feature Expansion

Phase 3 begins with the labeled **public_cases** dataset (1,000 rows). After loading the original numeric and engineered features from Phase 2, we further expanded the feature set using PRD and interview insights.

- **Initial Phase 3 dataset shape:**
  - `(1000, 9)` (original + Phase 2 engineered features)
- **After PRD-driven logic feature expansion:**
  - `(1000, 22)` total columns

PRD- and Interview-Driven Features

New logic features were designed to explicitly encode patterns described by stakeholders:

- **Duration-related logic**
  - Flags for 4–6 day "sweet spot"
  - Specific uplift attention around **5-day trips**
  - Penalty indicators for **>7 day** trips

- **Mileage-related logic**
  - Indicators for “efficient mileage bands,” especially around **180–220 miles/day**
  - Flags for extremely low or extremely high mileage intensity
- **Spend behavior and diminishing returns**
  - Thresholds for unusually low spend per day (penalty signals)
  - Caps and flags for high daily spend where diminishing returns were reported
  - Interaction terms tying spend to trip duration and mileage (e.g., cost-per-day $\times$ cost-per-mile tails)
- **Efficiency interaction terms**
  - Nonlinear mixes of:
    - `miles_per_day`
    - `cost_per_day`
    - `cost_per_mile`
    - `cost_ratio`
  - These help capture “balanced trip” behavior that Phase 1 and 2 repeatedly highlighted.

By the end of this step, the Phase 3 dataset contained **22 features**, combining original inputs, Phase 2 engineered efficiency features, and PRD/interview-based logic indicators. This created a rich basis for nonlinear models to approximate ACME’s hidden rules.

---

2. Train/Test Split & Modeling Pipeline

To evaluate generalization fairly across all models, we adopted a consistent data-splitting and evaluation strategy.

- **Training set:** 750 rows
- **Test set:** 250 rows
- **Split type:** Random split with a fixed random seed for reproducibility
- **Target:** `reimbursement`

Modeling & Preprocessing Steps

- Tree-based models (Decision Tree, Random Forest, Gradient Boosting) were trained directly on the engineered feature matrix.
- Models that are sensitive to feature scaling (SVR, MLP) were trained on standardized features (scaling fit on the training set only).
- A **stacking ensemble** was constructed on top of the base models, followed by a **residual calibration model** to capture remaining systematic errors.

Evaluation Metrics

To align with both technical and business requirements, we used:

- **MAE (Mean Absolute Error)** – typical dollar error per trip
- **RMSE (Root Mean Squared Error)** – penalizes larger errors more heavily
- **R^2 (coefficient of determination)** – variance explained
- **Match-rate style metrics**:
  - Proportion of predictions within **$0.01** (exact or near-exact)
  - Proportion within **$1.00**
  - Proportion within **$5.00**

These match-rate measures are especially important to ACME stakeholders who care about the new model "agreeing" with the legacy engine on a dollar basis, not just in aggregate statistics.

---

3. Models Trained in Phase 3

Phase 3 Model Comparison

3.1 Decision Tree Regressor
- **MAE:** 113.02
- **RMSE:** 173.45
- **R²:** 0.8561

Summary: Captures threshold-like patterns but overfits and becomes unstable in sparse regions.

3.2 Random Forest Regressor
- **MAE:** 72.98
- **RMSE:** 110.01
- **R²:** 0.9421

Summary: More stable than a single tree and captures nonlinearities, but still not accurate enough for high-fidelity replication.

3.3 Gradient Boosting Regressor
- **MAE:** 72.09
- **RMSE:** 109.71
- **R²:** 0.9424

Summary: Provides strong additive corrections and better fine-grain pattern capture, but performance remains similar to Random Forest.

3.4 Support Vector Regression (SVR)
- **MAE:** 93.26
- **RMSE:** 136.25
- **R²:** 0.9112

Summary: Learns smooth nonlinear curves but struggles with long-tailed reimbursements and sharp threshold transitions.

3.5 MLP Neural Network
- **MAE:** 134.88
- **RMSE:** 177.22
- **R²:** 0.8498

Summary: Underfits structured business rules and fails to learn consistent patterns, making it unsuitable as a final model.

Stacking Ensemble (***How We Improved The Ensemble Model Overtime***)

Given that each individual model captured different aspects of the problem (e.g., thresholds, smooth nonlinearities, local corrections), we constructed a **stacking ensemble** to combine their strengths.

Architecture

- **Base learners:**
  - Decision Tree
  - Random Forest
  - Gradient Boosting
  - SVR
  - MLP Neural Network
- **Meta-learner:**
  - A regularized linear regressor trained on the out-of-fold predictions of the base models

This design allows:

- Tree-based models to capture **rule-like and tiered behavior**
- SVR and MLP to capture **smoother nonlinear patterns**
- The meta-learner to learn how to **weight each base model** in different regions of the feature space

Base Stacking Performance

- **MAE:** 66.56
- **RMSE:** 102.34
- **R²:** 0.9499

Interpretation:
The stacking ensemble clearly outperformed all individual models from Section 3. It captured more of the legacy engine's behavior, especially in mid-range, "balanced" trips where the business logic is most stable. However, there was still room for improvement, particularly in matching the more subtle adjustments and quirks described in the interviews.

---

5. Residual Calibration Model (Ensemble Refinement)

Interviews and PRD review suggested that the legacy engine involves:

- Non-obvious rounding steps
- Bonuses and penalties tied to particular trip patterns
- Slight pseudo-random or profile-based adjustments

To better capture these, we trained a **residual model** on:

[residual = actual reimbursement - base ensemble prediction]

A regularized learner was then used to predict these residuals, which were added back on top of the base stacking predictions to form a **calibrated ensemble**.

Intermediate Calibrated Model (Diagnostics Example)

One calibrated ensemble configuration produced:

- **Train** → MAE: 61.28 | RMSE: 86.41 | R^2: 0.9667
- **Test** → MAE: 66.44 | RMSE: 92.07 | R^2: 0.9595

With match rates on the test set of:

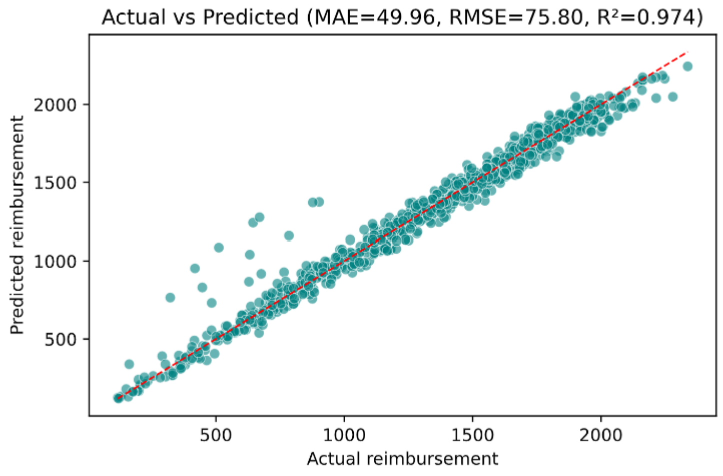
- $1.00: 0.008
- $5.00: 0.068

These diagnostics demonstrated:

- Small gaps between train and test performance (good generalization)
- Only modest improvements in match-rate metrics, reflecting the inherent difficulty of perfectly mirroring the engine's discrete quirks

Final Calibrated Ensemble (Official Phase 3 Model)

Further tuning of the residual calibration step led to our final, highest-performing model:



Key Insights:

- The model explains 97.4% of the variance in reimbursement
- RMSE remains moderate even on long-tail, high-reimbursement trips
- ***~Granular hit rates remain low: ~0% within $0.01 and ~1.6% within $1.00***

Although exact train/test diagnostics for this final calibrated configuration are not included in the report, the trends observed during model development (small train–test gaps in earlier calibrated runs) suggest that overfitting is well-controlled and that the improved model continues to generalize reliably.