ACME Legacy Reimbursement -- Final Model Execution Guide

Purpose

This document explains how to execute and evaluate the final reimbursement model used in this project.

The repository contains several notebooks and exploratory experiments. However, only one trained model and one execution path produce the metrics reported in Phase 3 and Phase 4 of the project summary. This guide exists to make that distinction clear and to ensure results can be reproduced consistently.

The final model

The final model is stored as a serialized Python object at:

`src/final_model.pkl`

This file contains a fully trained stacking ensemble model. It is a binary artifact saved using Python tooling and is not intended to be opened or viewed as text. Seeing unreadable characters when opening the file in an editor is expected and does not indicate an error.

All Phase 3 metrics referenced in the report are derived from evaluating this saved model. Retraining models inside notebooks or scripts will produce different, typically weaker, results and should not be used for final reporting.

How Phase 3 results are reproduced

The authoritative way to reproduce the Phase 3 metrics is to run the evaluation script:

`script/phase3_performance_metrics.py`

This script evaluates the saved final model. It does not retrain a new one.

Running the evaluation

From the project root directory (the folder containing `src/`, `data/`, `scripts/`, and `reports/`), run:

```
python scripts/phase3_performance_metrics.py
```

What this script does

The script performs the following steps:

- Loads the trained model from `src/final_model.pkl`
- Loads the engineered Phase 2 dataset from `data/phase2_features_baseline_models.csv`
- Applies a fixed 75/25 holdout split, using the last 25 percent of rows as test data
- Computes standard regression metrics and precision thresholds
- Writes prediction outputs to `data/phase3_predictions.csv` for audit and inspection

Expected output

When run correctly against the saved final model, the expected Phase 3 results are approximately:

- MAE around 62
- RMSE around 95

- R^2 around 0.95
- Exact matches (≤ $0.01): approximately 0 percent
- Close matches (≤ $1.00): approximately 1.6 percent
- Test set size: 250 rows

If materially different values appear, such as MAE near 79 or close match rates near 0.4 percent, this indicates that a different model was evaluated. The most common cause is retraining a lightweight baseline model instead of loading the saved final artifact.

Supporting files

The following files support analysis and inspection but are not the source of record for final metrics:

File	Purpose
`Notebooks/04_Ensemble_Model_Metrics.ipynb`	Interactive version of the Phase 3 evaluation that loads the final model
`scripts/phase3_performance_metrics.py`	Command-line evaluator for the tuned model; prints MAE/RMSE/R^2 and close/exact hit rates
`data/phase2_features_baseline_models.csv`	Engineered feature set used to evaluate the model
`src/predict.py`	Lightweight wrapper for generating predictions from inputs
`data/phase3_predictions.csv`	Saved predictions from the official Phase 3 evaluation
`reports/project_Summary.qmd`	Main report; Phase 3 scatter and Phase 4 permutation plots load the tuned model on the 75/25 holdout
`reports/postprocessing_tuning.qmd`	Technical log of post-processing variants and their metrics
`reports/phase3_phase4_addendum.qmd`	Narrative addendum for Phase 3/4 findings aligned to the tuned model

Notebooks that retrain models are useful for learning and experimentation, but they should not be used to report final performance.

Interpreting the results

The final model closely reproduces the overall payout structure of the legacy reimbursement engine. This is reflected in the strong R^2 and reduced MAE and RMSE relative to earlier baselines.

At the same time, the model rarely matches the legacy system at the penny or dollar level. This suggests the presence of step-based business logic in the legacy engine, such as tiering, caps, or rounding rules, that are difficult to infer purely from historical data.

As a result, the current model serves as a reliable benchmark for understanding legacy behavior and comparing systems. It is not yet a penny-accurate replacement without additional rule discovery or an agreed tolerance band.