Business Travel Reimbursement Model

CSCI / DASC 6020: Machine Learning Team Project

Saniyah Khan, Vy Tran, Harriet O'Brien, Rehinatu Usman

Team and Role

- Harriet O'Brien: Machine Learning Engineer
- Vy Tran: Data Scientist
- Saniyah Khan: Business Analyst
- Rehinatu Usman: Software Engineer

Agenda

- Introduction
- Dataset overview
- Feature engineering
- Modeling Approach
- Results and evaluation

Business Problem

- Travel reimbursements are handled by an old, legacy software engine.
- The reimbursement rules (per-diem limits, mileage rates, caps, etc.) are buried in opaque, hard-to-maintain code.
- The organization only sees the three inputs (days, miles, receipts) and the final reimbursement amount, no clear logic in between.
- As a result, the rules cannot be easily audited, updated, or migrated into a modern system.

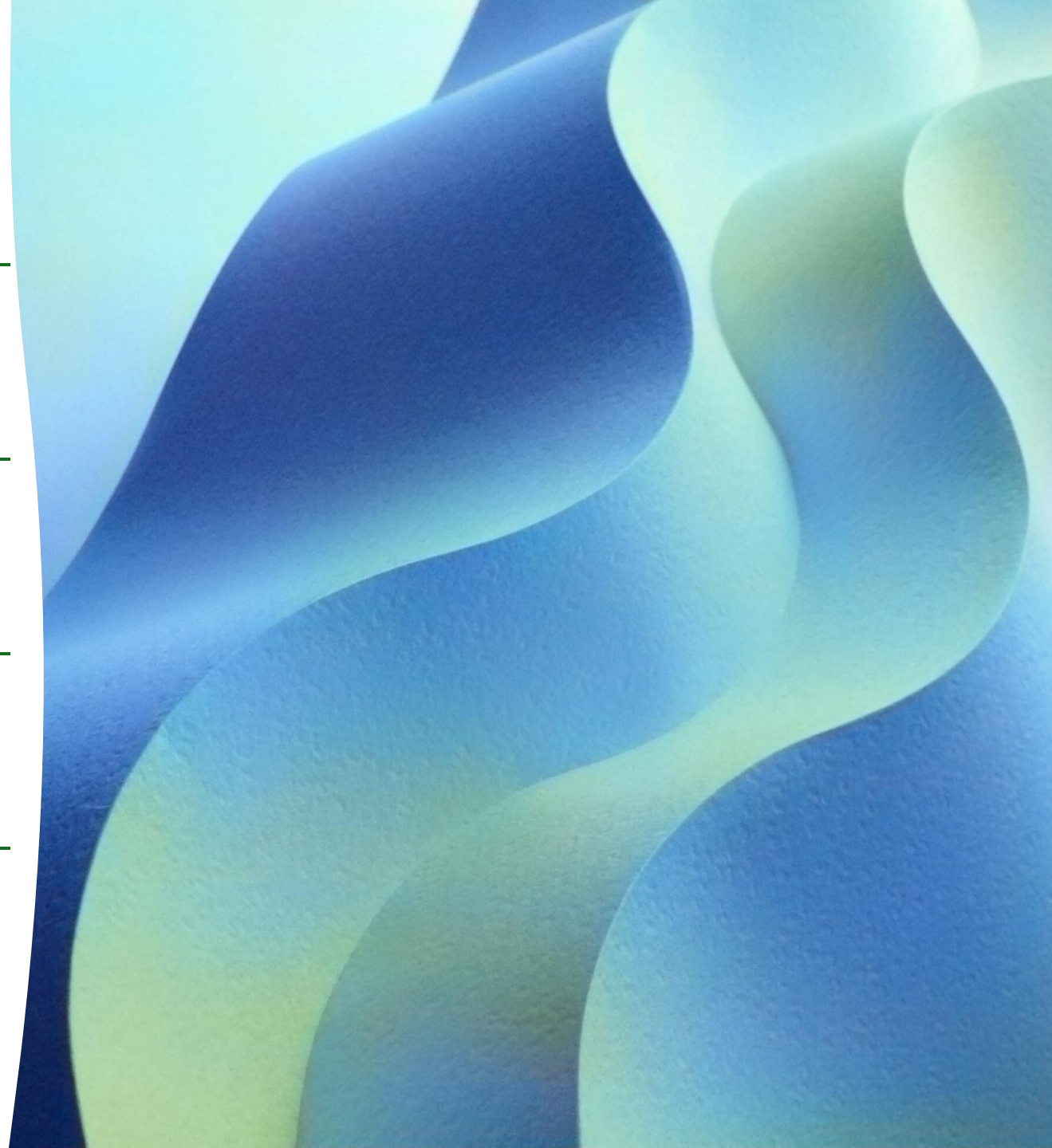
Why this Matters

The legacy engine is a single point of failure: if it breaks or must be retired, reimbursement processing is at risk.

Finance and HR cannot clearly explain why two similar trips get different reimbursements, which hurts transparency and employee trust.

Lack of documentation makes it difficult to change policy or test “what-if” scenarios (new rates, caps, or rules).

The opaque logic is hard to integrate with modern tools (expense apps, dashboards, analytics), limiting future automation and reporting.



Dataset Overview

- The starting point for our project is a JSON file, public_cases.json, which contains 1,000 historical reimbursement records from a legacy travel reimbursement system.
- Each record contains input data and expected output.

Assumptions and Hypothesis

- Legacy system consists of layered, undocumented rules
  - Evolved over decades, accumulating special cases, outdated adjustment, etc.
- Reimbursement depends on receipts, duration, and mileage
  - Data prove that these three inputs have an influence on reimbursement

Assumptions and Hypothesis

- Relationships are nonlinear and threshold-based
  - Patterns in data suggest diminishing return bonuses, penalties, etc.
- Goal is to replicate behavior, not optimize policy
  - Try to match the legacy system's output as much as possible even if the rules appear to be "wrong."

Success Criteria

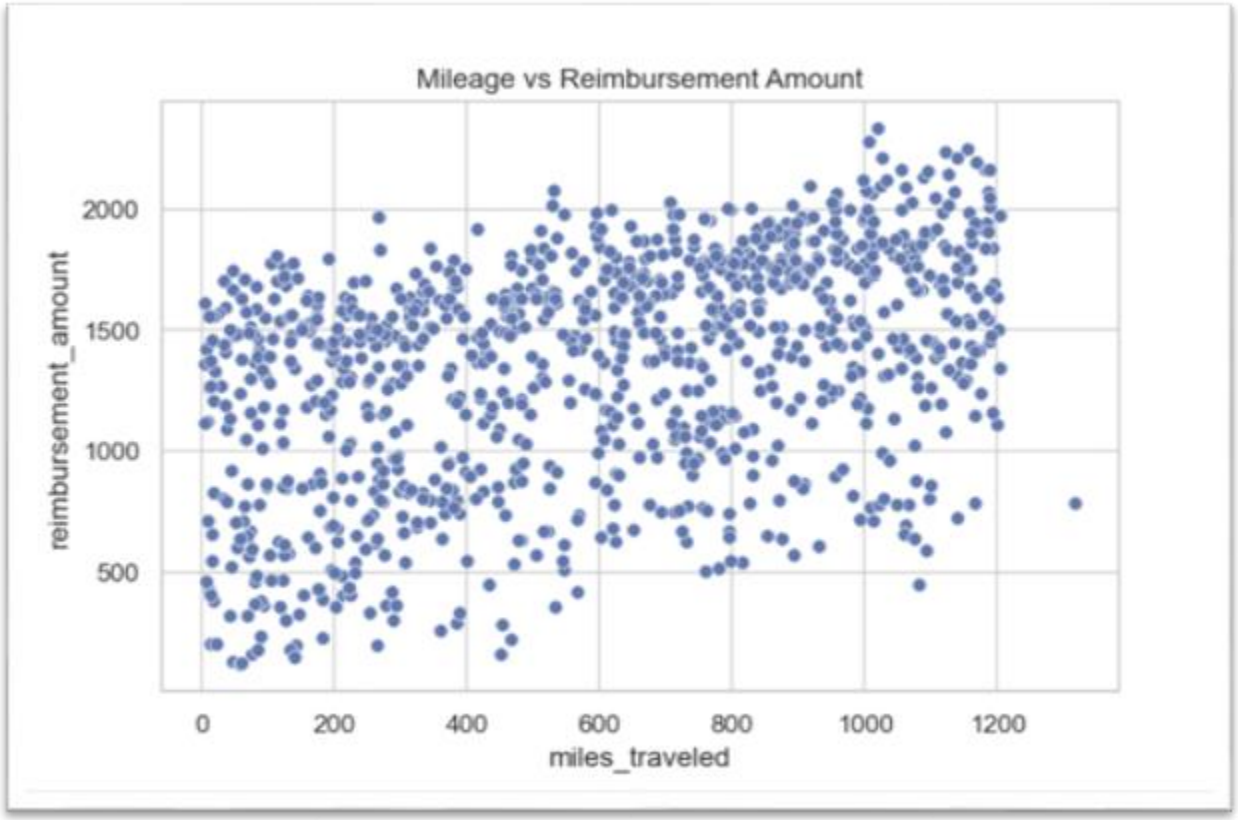
- Low MAE on held-out test data
- Stable performance across models
- Interpretability aligns with interviews and EDA
- Model deployable within project contraints

EDA Summary

Data shows strong linear relationships.

Duration and mileage increase with reimbursement.

Few outliers, likely due to missing receipts or unusual travel case.

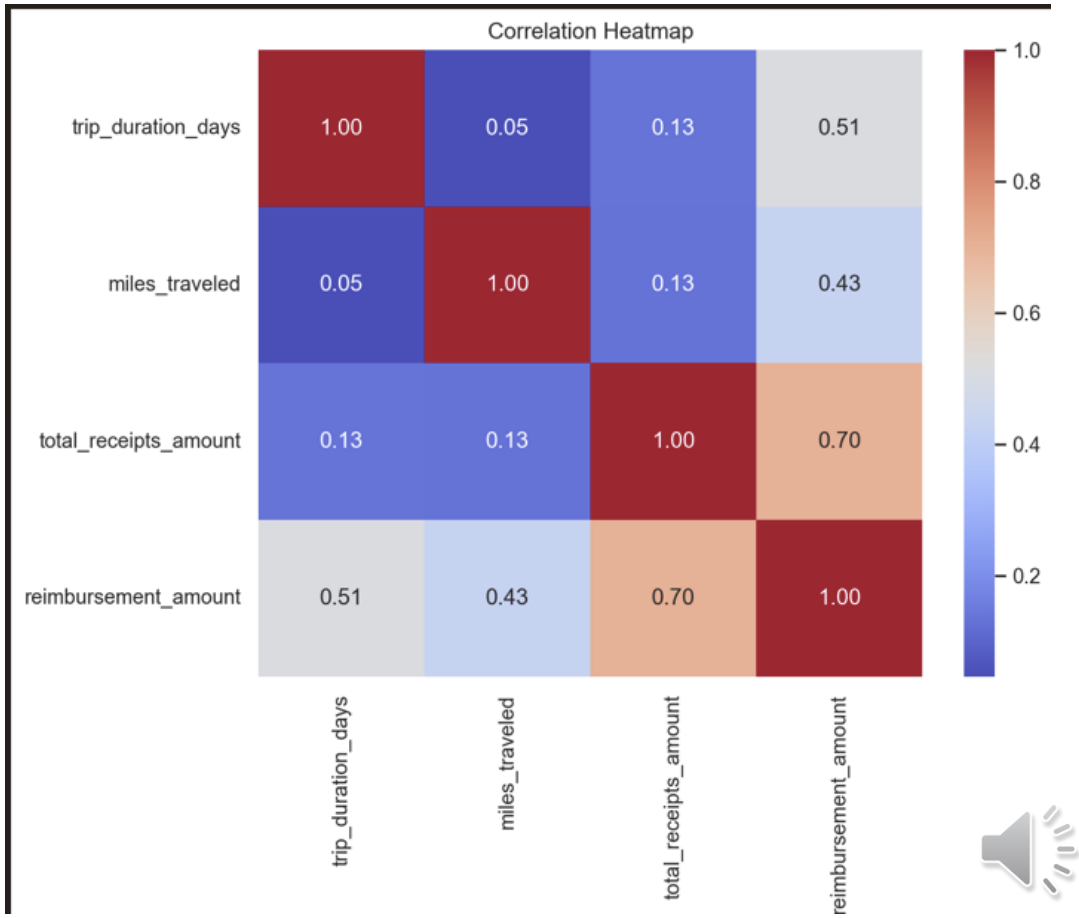


Key Data Insights (Visualization)

Correlation heatmap confirms key relationships.

No significant multicollinearity detected.

Data structure supports linear modeling approach.



Feature Engineering

Created new useful features:

- cost_per_mile
- cost_per_day
- receipts_ratio
- Scaled numeric features for consistent modeling.

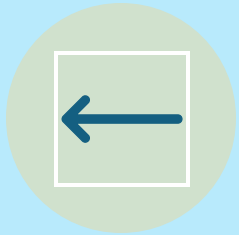
	trip_duration_days	miles_traveled	total_receipts_amount	reimbursement_amount
0	3	93.0	1.42	364.51
1	1	55.0	3.60	126.06
2	1	47.0	17.97	128.91
3	2	13.0	4.67	203.52
4	3	88.0	5.78	380.37

Modeling Approach

MODELS TESTED:

LINEAR REGRESSION

RANDOM FOREST



GRADIENT BOOSTING

REASON: COMPARE PERFORMANCE + INTERPRETABILITY

Model Performance Comparison

Model	MAE	RMSE	R^2
Linear Regression	~138.006	~167.302	0.859
Random Forest	~45.079	~80.684	0.967
Gradient Boosting	~47.905	~73.143	0.973

	Model	MAE	RMSE	R²
0	Linear Regression	138.006031	167.302360	0.859110
1	Random Forest	45.078747	80.684184	0.967232
2	Gradient Boosting	47.905114	73.143327	0.973071

➔ Raw model evaluation Output from notebook.

Selected Model

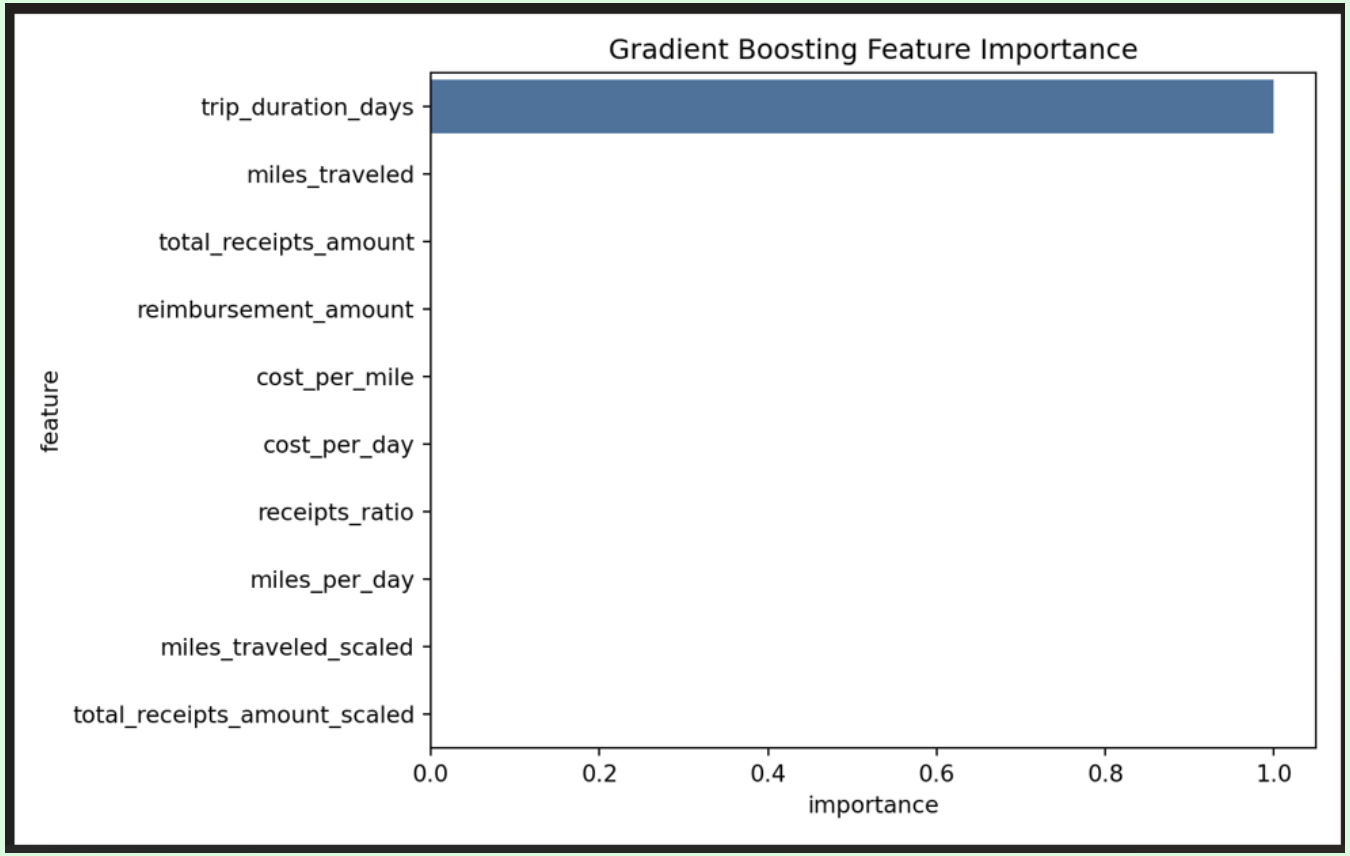
Final model: **Gradient Boosting**

Why:

- Highest predictive performance
- Captures non-linear relationships
- Consistent results across validation

Interpretability: Feature Importance

- Receipt amount is the strongest driver of reimbursement.
- Trip duration has a secondary effect.
- Mileage contributes but to a lesser extent.
- Other engineered features have minimal impact.

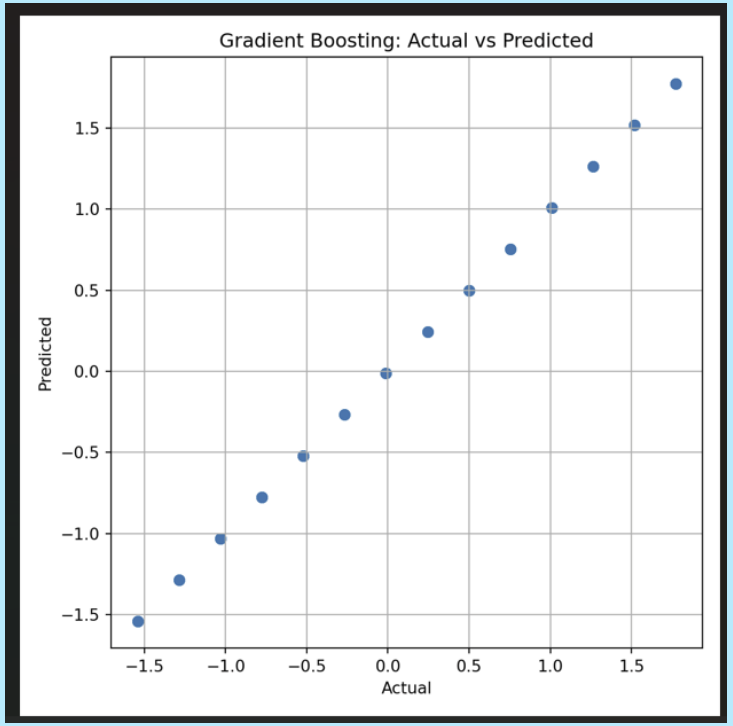


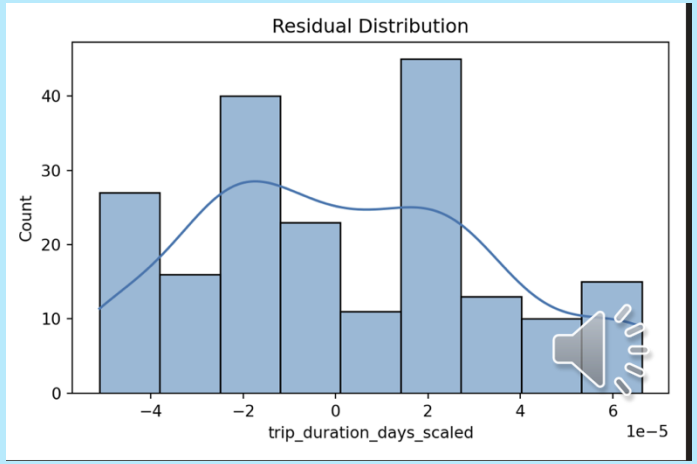
Model Validation

Predictions closely match true values.

Residuals centered around zero.

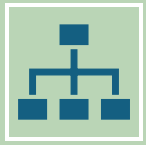
No major bias detected.





Risks and Limitations

May generalize poorly on unseen cases.



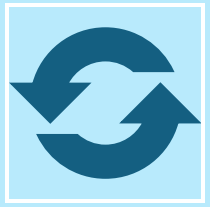
Feature engineering may overfit structure.

Needs real-world testing.

Future Improvements

Add more real-world testing.

Add monitoring alerts.



Retrain when business rules change.

Consider additional external features.

Exported Final Model

- Final saved model: final_model.pkl
- Ready for ML engineering handoff

```
# Export Final Model
```

```
```{python}
import joblib
# We select Gradient Boosting as the final model due to its strong predictive performance and stability.
final_model = gbr
joblib.dump(final_model, "../models/final_model.pkl")
print("Model saved as final_model.pkl")
```
```

```
['../models/final_model.pkl']
Model saved as final_model.pkl
```

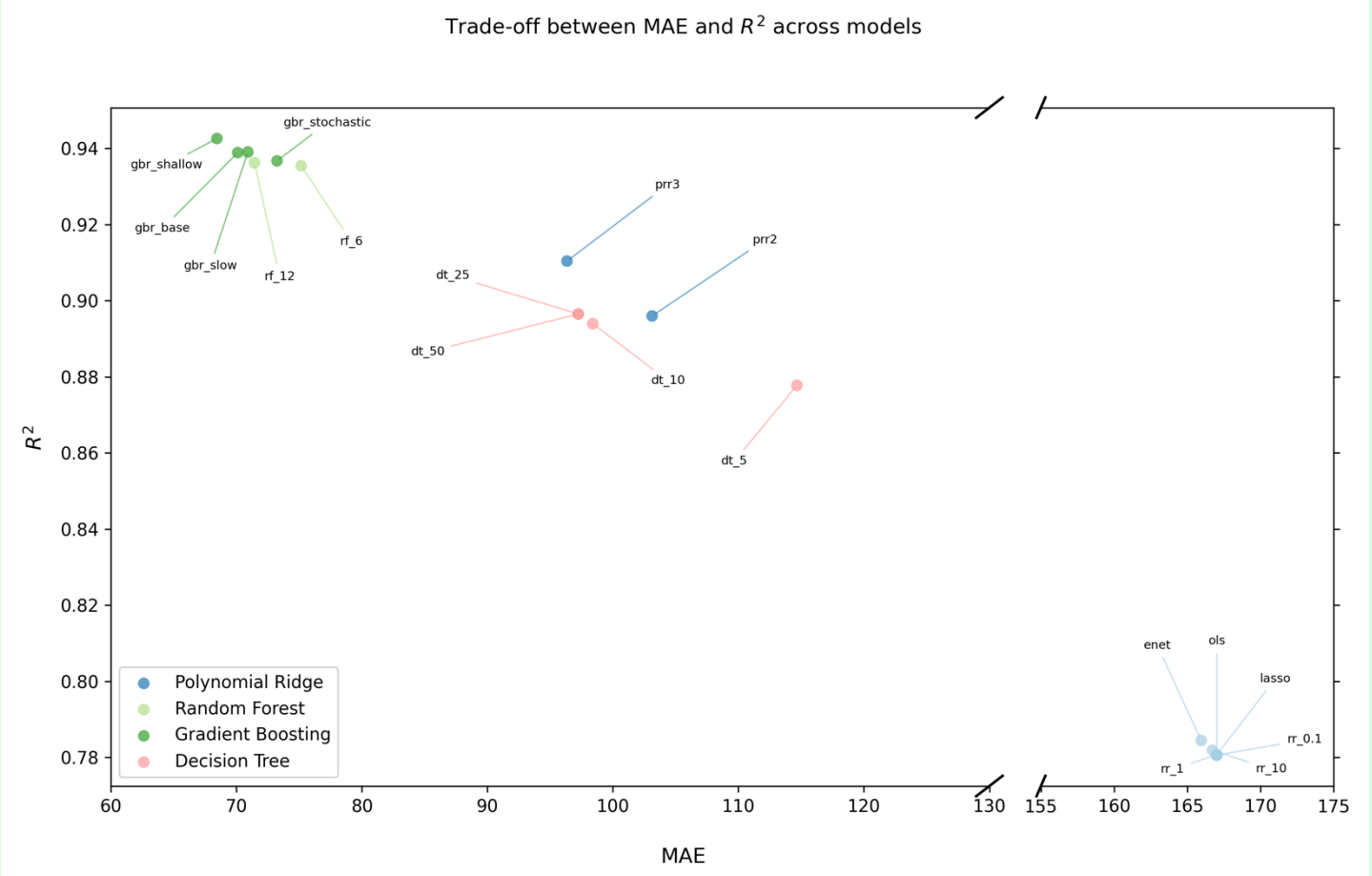
Model Development Pipeline

- Legacy engine treated as a black box
- Inputs: trip duration, miles traveled, total receipts
- YAML-driven pipelines for linear and tree-based models
- Consistent 80/20 train-test split and shared metric helper
- Final selected model retained on all data and saved as **`final_model.pkl`**

```yaml
ols:
  preprocessing.StandardScaler:
  linear_model.LinearRegression:

rr_1:
  preprocessing.StandardScaler:
  linear_model.Ridge:
    alpha: 1.0

…

gbr_stochastic:
  ensemble.GradientBoostingRegressor:
    n_estimators: 600
    learning_rate: 0.03
    max_depth: 3
    subsample: 0.8
    random_state: 42

dt_5:
  tree.DecisionTreeRegressor:
    max_depth: 5
    random_state: 42
```

Model Selection and Performance

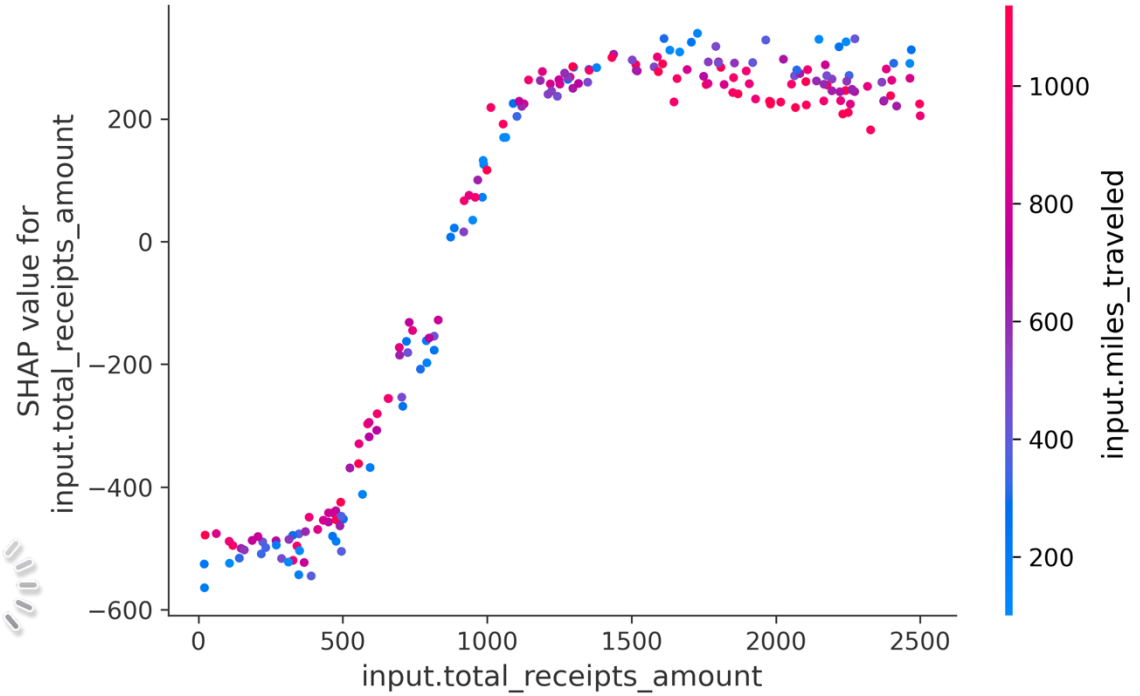
- Shared 80/20 split and common metrics: MAE, RMSE, MedAE, R^2, P90, MaxE, MAPE
- Linear / regularized baselines
  - MAE ≈ $166, R^2 ≈ 0.78
- Polynomial Ridge + single trees
  - big improvement but still behind ensembles
- Random Forests:
  - MAE ≈ $71–$75, R^2 ≈ 0.94
- Gradient Boosting: best overall; **`gbr_shallow`** selected
  - MAE ≈ $68, RMSE ≈ $107, R^2 ≈ 0.94, MAPE ≈ 6.35%

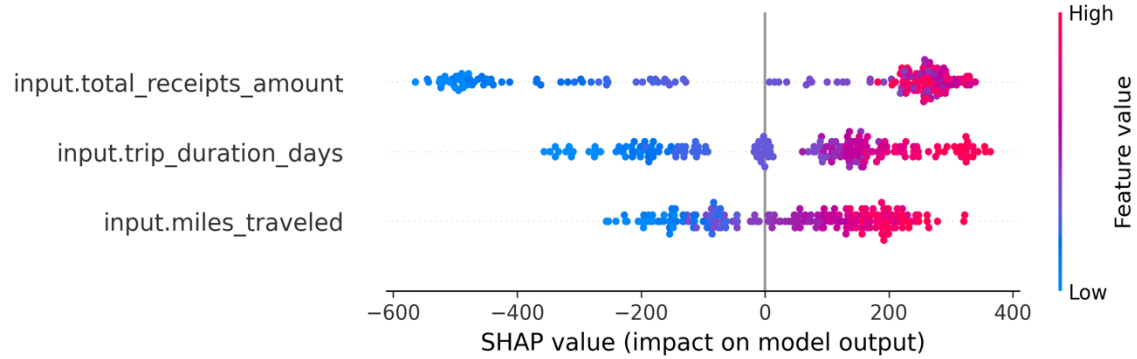


Diagnostics:
Prediction vs. Actual, Residuals, and Calibration look healthy

Model Interpretability (SHAP & Behavior)

- **XGBoost** surrogate trained on same split as scikit-learn models
- Test performance closely matches **`gbr_shallow`**
- SHAP Summary Plot:
  - Total_receipts_amout is dominant driver
  - Trip_duration_days and miles_traveled provides econdary refinements
- Behavior aligns with domain expectations for a realistic reimbursement policy





Business Impact

- Enables modernization without access to legacy code
- Reduces dependency on legacy systems
- Provides explainable reimbursement estimates
- Supports safe transition to new system

Next Steps

- Confirm that the model generalizes beyond the public dataset
- Add rule-based checks for extreme-based cases
- Incorporate time-based features if available
  - Such as seasonality or submission timing could help improve accuracy if data becomes available
- Finalize deployment with a clean interface that meets the constraints and runtime

Conclusion

- Combined qualitative & quantitative approach
  - Interviews, EDA and modeling together provide a better understanding of system behavior

- Gradient Boost best approximates legacy logic
  - This model captures nonlinear patterns and rule-like behavior more effectively than simpler baselines

Conclusion

- Interpretability confirms alignment with the expectations set
  - Feature importance and PDPs show that predictions follow reasonable and explainable patterns

- Model provides a practical solution
  - The final model balances accuracy, transparency and deployable