ML Implementing Challenge

Technical Report – AI Engineering Hiring

Andres Fernando Gomez Rojas

Data Scientist

February 26, 2026

Contents

Introduction

This report presents the complete solution to the Wizeline ML Implementing Challenge. The objective is to train a regression model to predict a continuous target variable from 20 numerical features, generate predictions on a 200-sample blind test set, and propose a deployable solution for regular batch predictions.

The dataset consists of 800 training samples with 20 features (`feature_0` through `feature_19`) and a continuous target column. All features are numeric (float64) with no null values or outliers, as stated in the challenge specification.

The methodology follows two main phases: (1) a comprehensive exploratory data analysis (EDA) to understand the data structure and inform modeling decisions, and (2) a systematic model comparison, hyperparameter tuning, and final prediction pipeline.

Exploratory Data Analysis

Descriptive Statistics

All 20 features are continuous floating-point values. The dataset contains exactly 800 samples with zero null values across all columns. Table 1 summarizes the key statistics.

Table 1: Descriptive statistics for selected features and target.

Variable	Mean	Std	Min	25%	75%	Max
feature_0	468.18	270.80	0.92	239.33	704.65	940.77
feature_2	317.13	176.50	0.17	167.52	474.48	614.27
feature_9	196.00	111.04	0.26	95.05	291.55	384.92
feature_11	3.37	1.99	0.004	1.63	5.02	6.86
feature_13	100.11	59.15	0.19	48.13	149.97	203.12
target	14.63	5.09	0.28	10.88	18.22	27.36

The features exhibit very different scales: `feature_11` ranges from 0 to approximately 7, while `feature_12` ranges from 0 to nearly 980. This scale disparity is relevant for distance-based and regularized models.

Distributions: Histograms and Boxplots

Histograms and boxplots were generated for all 21 variables (20 features + target). Most features follow approximately uniform distributions across their respective ranges, consistent with the absence of outliers noted in the challenge specification.

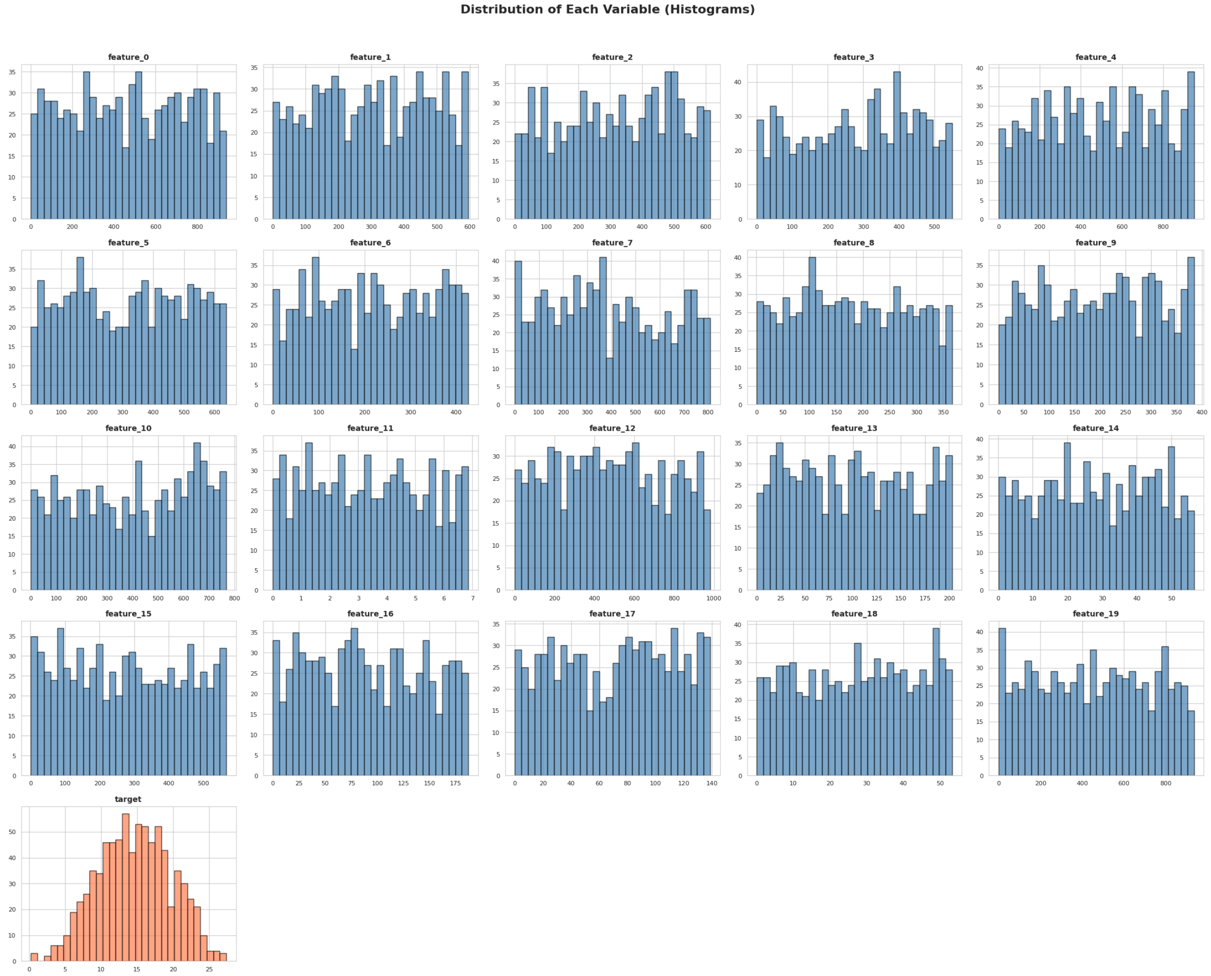

Figure 1: Histograms of all 20 features and the target variable.

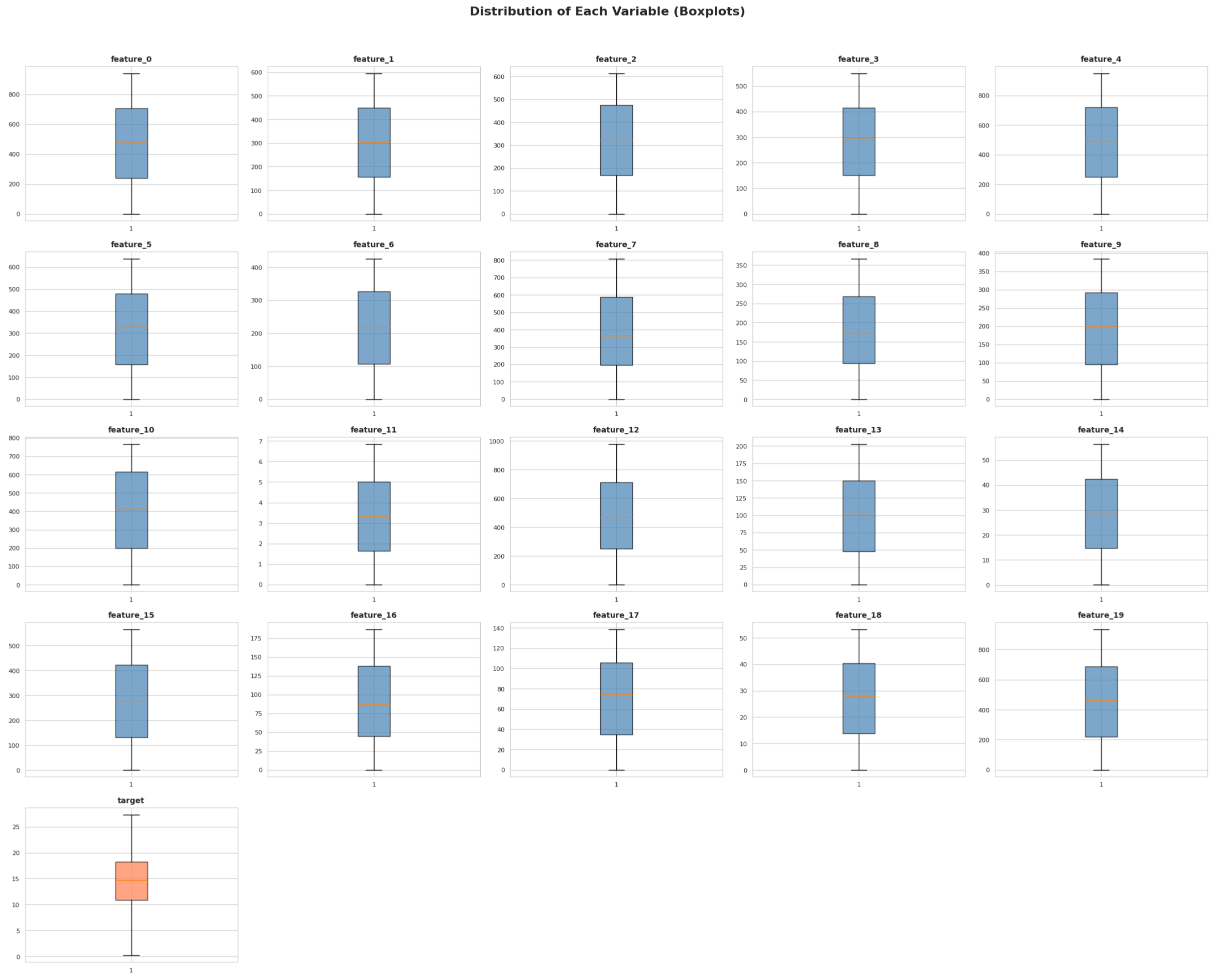

Figure 2: Boxplots of all 20 features and the target variable.

Target Normality

The target variable was assessed for normality to determine whether transformations might be beneficial for linear models.

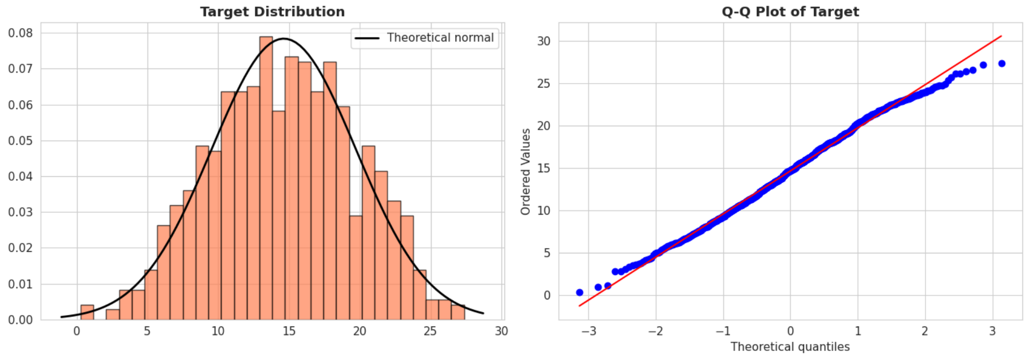

Figure 3: Target distribution with theoretical normal overlay (left) and Q-Q plot (right).

Table 2: Target normality statistics.

Statistic	Value
Skewness	-0.0398
Kurtosis	-0.5402
Shapiro-Wilk statistic	0.9949
Shapiro-Wilk p-value	0.0299

The skewness is very close to zero (-0.04), indicating near-perfect symmetry. The kurtosis of -0.54 suggests a slightly platykurtic (flatter than normal) distribution. While the Shapiro-Wilk test rejects exact normality at $\alpha = 0.05$, the distribution is sufficiently close to normal that no transformation (log, Box-Cox) is warranted.

Pearson Correlation Analysis

The Pearson correlation matrix reveals the linear relationships between all variables.

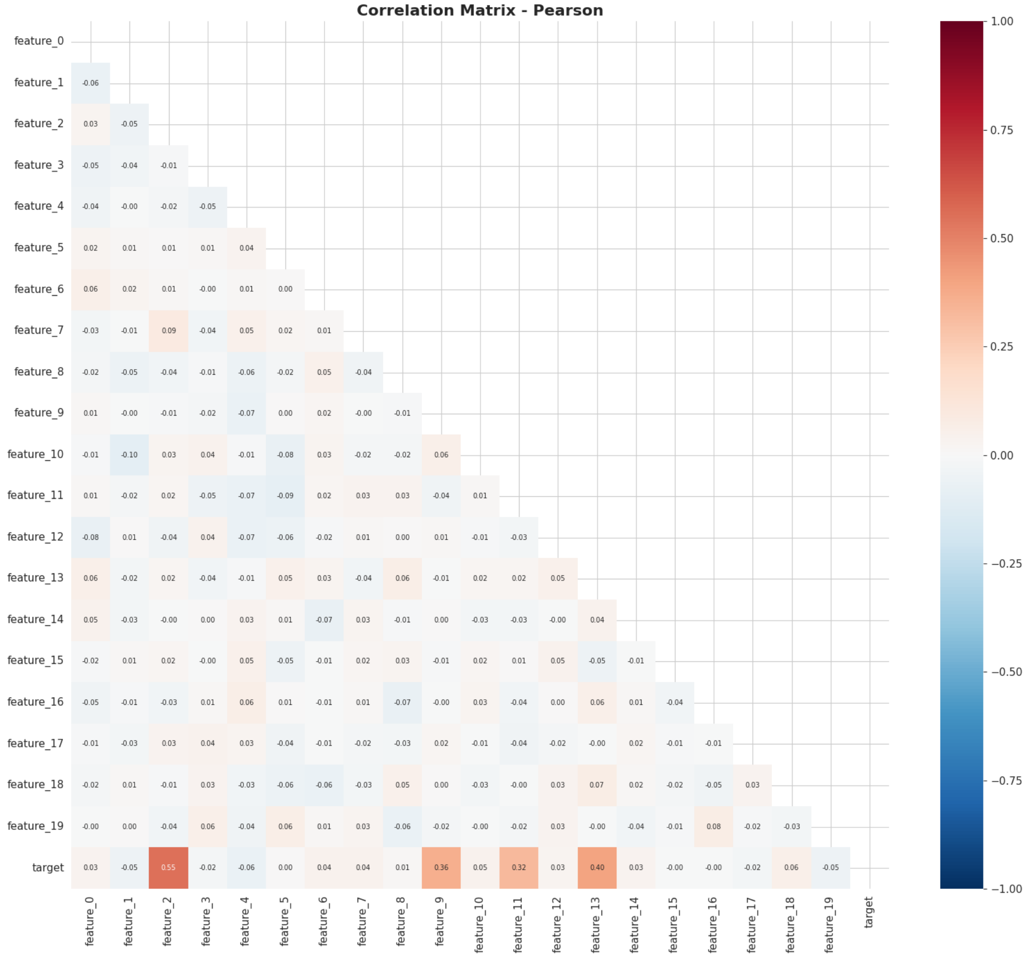

Figure 4: Pearson correlation matrix for all features and target.

Table 3: Pearson correlation with target (sorted by absolute value).

Feature	r	Feature	r
feature_2	0.5518	feature_19	-0.0535
feature_13	0.4047	feature_1	-0.0472
feature_9	0.3619	feature_10	0.0455
feature_11	0.3228	feature_6	0.0444
feature_4	-0.0627	feature_7	0.0411
feature_18	0.0601	*others*	$\|r\| < 0.04$

Only four features show meaningful linear correlation with the target: `feature_2` ($r = 0.55$), `feature_13` ($r = 0.40$), `feature_9` ($r = 0.36$), and `feature_11` ($r = 0.32$). The remaining 16 features have near-zero linear correlation.

Spearman Correlation Analysis

Spearman rank correlation was computed to detect monotonic non-linear relationships.

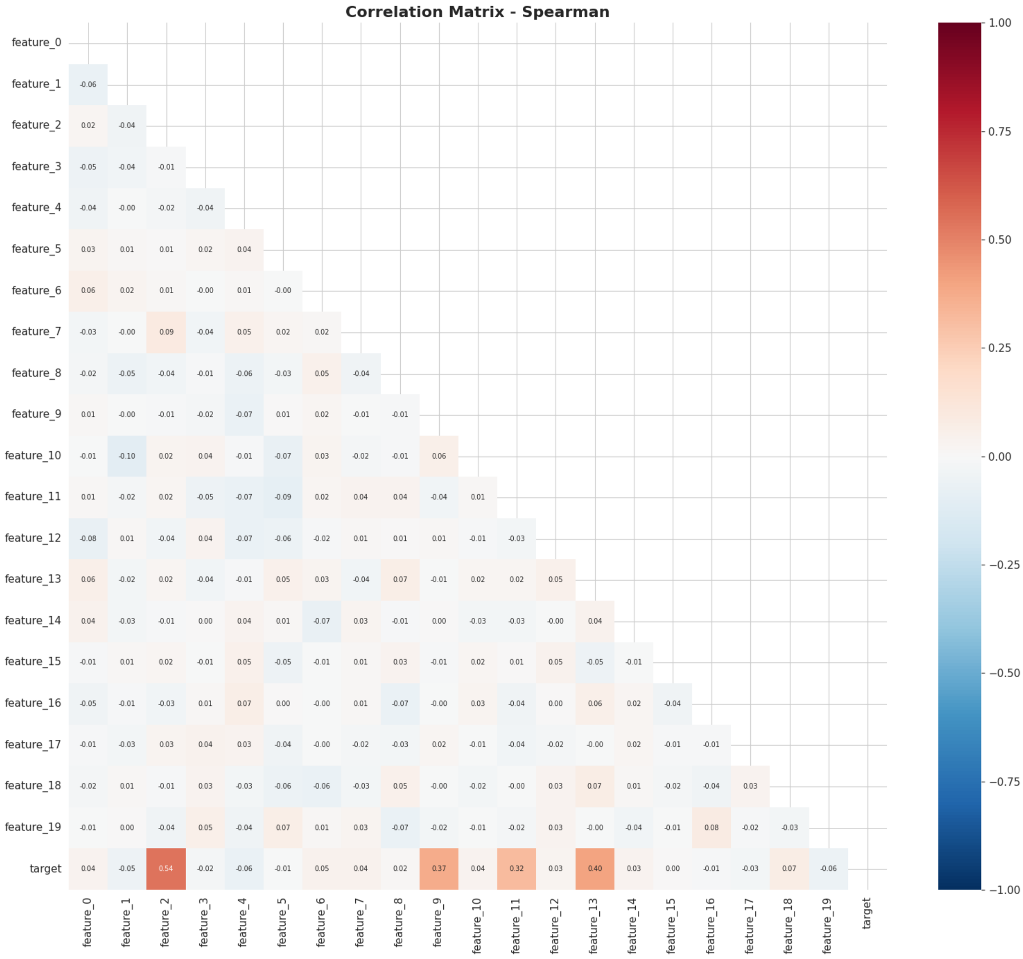

Figure 5: Spearman correlation matrix.

The Spearman results closely mirror the Pearson results, indicating that the relationships between features and target are predominantly linear rather than monotonically non-linear.

Scatter Plots: Top Features vs Target

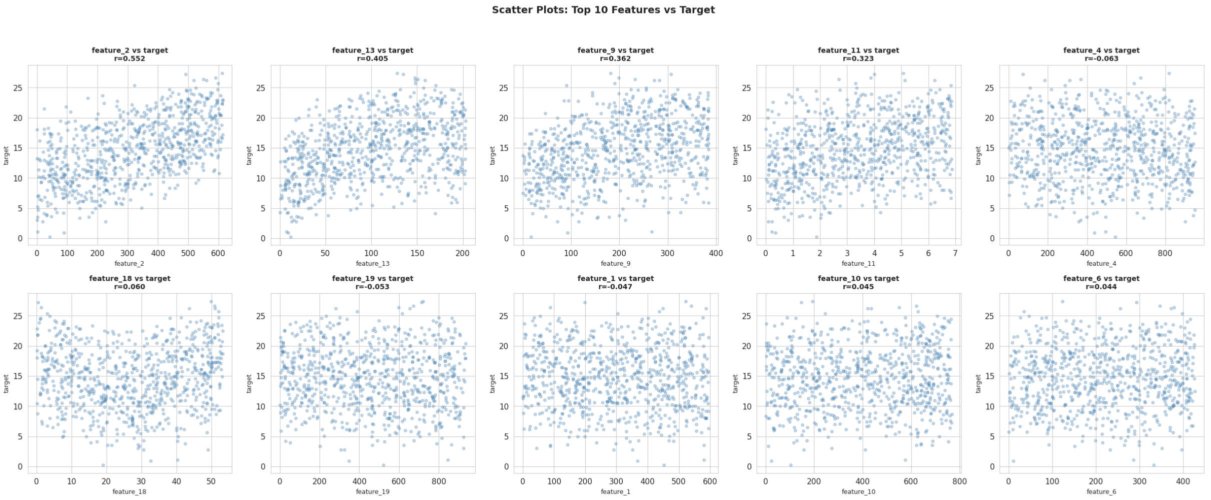

Figure 6: Scatter plots of the top 10 correlated features against the target.

The scatter plots confirm the linear trends for the top four features and the lack of visible pattern for the weakly correlated features.

Variance Inflation Factor (VIF)

VIF was computed for all 20 features to assess multivariate collinearity. For each feature, a linear regression was fit using all other features as predictors, and VIF was calculated as VIF $= 1/(1 - R^2)$.

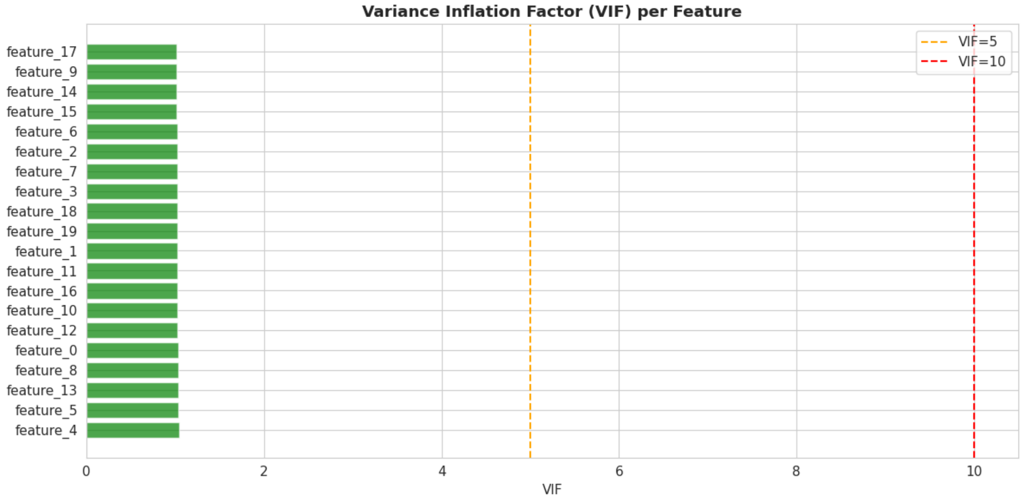

Figure 7: Variance Inflation Factor per feature.

All VIF values fall below 1.05 (the highest being 1.040 for `feature_4`), which is well below the standard thresholds of 5 (moderate) and 10 (severe). This confirms that the 20 features are practically independent of each other, and no feature removal is needed for collinearity reasons.

Principal Component Analysis (PCA)

PCA was performed on standardized features to assess dimensionality reduction potential and visualize the data structure.

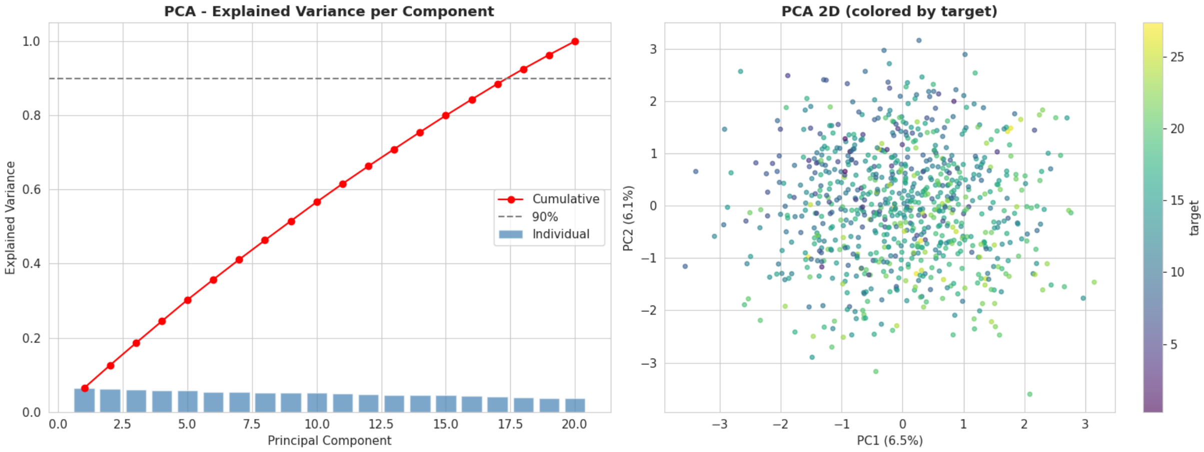

Figure 8: PCA explained variance per component (left) and 2D projection colored by target (right).

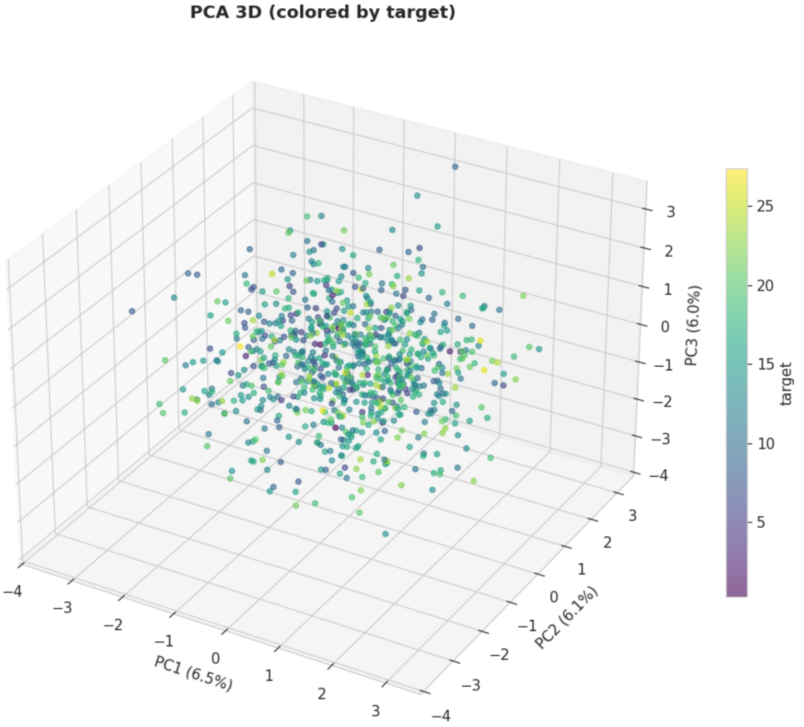

Figure 9: PCA 3D projection colored by target value.

Cumulative explained variance reaches 30.2% with 5 components and 56.6% with 10 components. Approximately 18 components are needed to explain 90% of the variance. This is consistent with the low VIF values: the features are largely independent, so PCA cannot achieve meaningful dimensionality reduction. The 2D and 3D projections show a gradient in target values, confirming that some structure is present even in the first few components.

Exploratory Clustering

K-Means clustering was applied to the standardized data to detect natural subgroups.

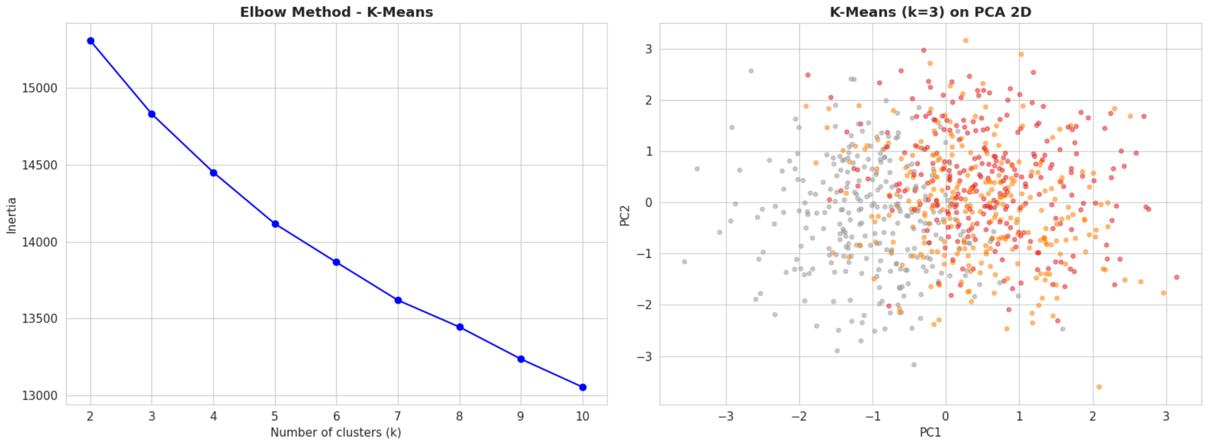

Figure 10: Elbow method (left) and K-Means clusters projected on PCA 2D (right).

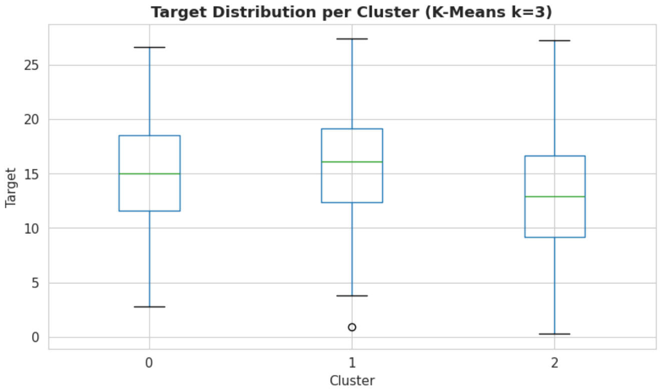

Figure 11: Target distribution per K-Means cluster (k=3).

The elbow curve shows no sharp inflection point, suggesting there are no strongly separated natural clusters in the data. The target distributions across the three clusters overlap substantially, indicating that a single global model is appropriate rather than cluster-specific models.

Feature Interactions

Products of the top 5 features (by Pearson correlation) were computed, and their correlation with the target was measured.

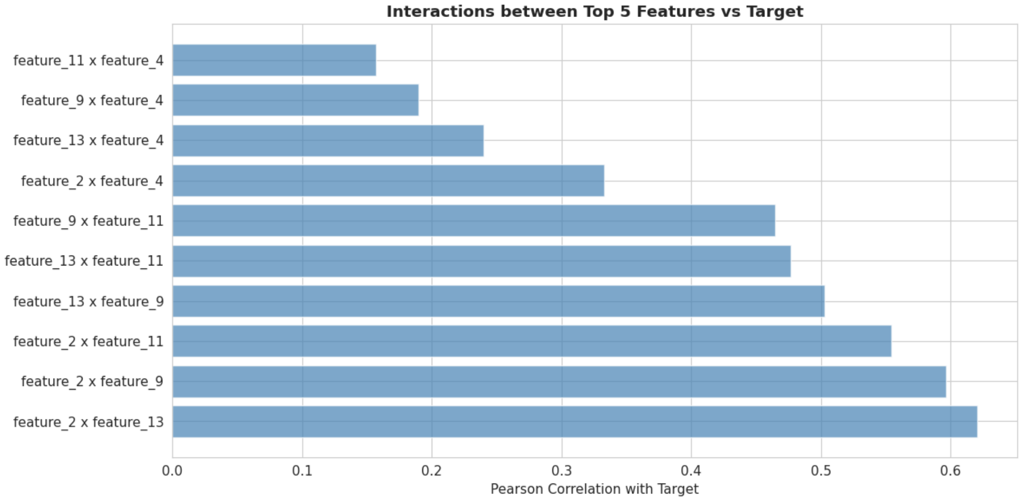

Figure 12: Pearson correlation of feature interactions (products) with the target.

Table 4: Top feature interactions vs target correlation.

Interaction	Pearson	Spearman
feature_2 × feature_13	0.6203	0.6444
feature_2 × feature_9	0.5962	0.6242
feature_2 × feature_11	0.5546	0.5687
feature_13 × feature_9	0.5028	0.5954
feature_13 × feature_11	0.4767	0.5085
feature_9 × feature_11	0.4646	0.4961

This is a critical finding: the interaction `feature_2` × `feature_13` achieves $r = 0.62$ with the target, which is higher than any individual feature ($r = 0.55$ for `feature_2` alone). This reveals that the target depends on multiplicative combinations of features, not just individual features additively. This motivates the use of interaction features in modeling.

Mutual Information

Mutual Information (MI) was computed between each feature and the target using `sklearn.feature_selection` Unlike correlation, MI captures any statistical dependency (linear and non-linear).

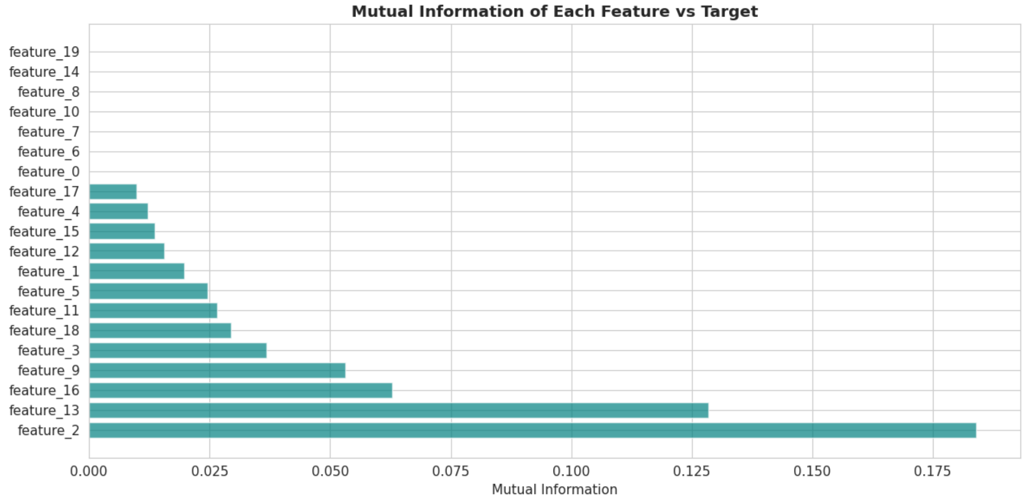

Figure 13: Mutual Information of each feature with the target.

Table 5: Mutual Information scores (top features and zero-MI features).

Feature	MI	Feature	MI
feature_2	0.1839	feature_0	0.0000
feature_13	0.1285	feature_6	0.0000
feature_16	0.0628	feature_7	0.0000
feature_9	0.0532	feature_8	0.0000
feature_3	0.0367	feature_10	0.0000
feature_18	0.0295	feature_14	0.0000
feature_11	0.0266	feature_19	0.0000

Notably, `feature_16` ranks third in MI (0.063) but has near-zero Pearson correlation ($r = -0.004$), indicating a non-linear relationship with the target that linear metrics miss. Seven features have MI $= 0$, suggesting they contribute no information to predicting the target.

Feature Scaling

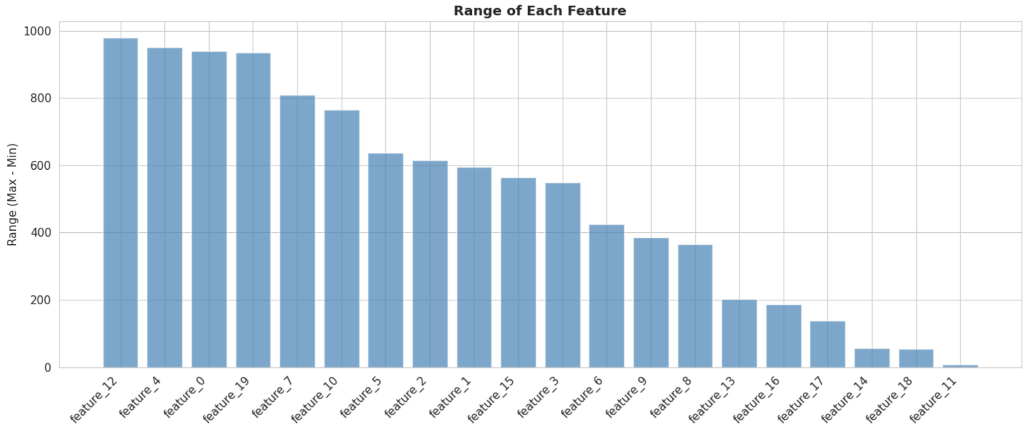

Figure 14: Range of each feature (Max − Min).

The scale ratio between the largest range (`feature_4`: ≈ 950) and smallest range (`feature_11`: ≈ 7) exceeds 135×. Standardization is therefore essential for distance-based models (SVR) and regularized models (Ridge, Lasso).

EDA Summary

The key findings from the exploratory analysis are:

- No null values, no outliers, no multicollinearity (all VIF < 1.05).
- Four features dominate the linear relationship with the target: `feature_2`, `feature_13`, `feature_9`, `feature_11`.
- Feature interactions (products) yield stronger correlations than individual features, with `feature_2` × `feature_13` reaching $r = 0.62$.
- Seven features have zero Mutual Information and appear to be noise.
- The target is approximately normally distributed; no transformation needed.
- Feature scales vary by $> 100\times$; standardization required for certain models.

Modeling

Methodology

Based on the EDA findings, the following modeling strategy was adopted:

1. **Data split**: 80/20 hold-out (640 train / 160 test), with a fixed random seed for reproducibility.
2. **Interaction features**: Six interaction columns were created from the top 4 features: `feature_2` × `feature_13`, `feature_2` × `feature_9`, `feature_2` × `feature_11`, `feature_13` × `feature_9`, `feature_13` × `feature_11`, and `feature_9` × `feature_11`.
3. **Model comparison**: Each model was tested with and without interactions, using 5-fold cross-validation on the training set.
4. **Hyperparameter tuning**: Top 3 models were tuned using RandomizedSearchCV (50 iterations, 5-fold CV).
5. **Final retraining**: The best model was retrained on all 800 samples.
6. **Persistence**: The model was saved as `.pkl` and experiments were logged with MLflow.

Models Evaluated

Six model families were evaluated, each in two variants (with and without interaction features), yielding 12 configurations:

1. **Linear Regression**: Baseline with no regularization.
2. **Ridge**: L2 regularization with standardization pipeline.
3. **Lasso**: L1 regularization with standardization pipeline.
4. **Random Forest**: Ensemble of 200 decision trees.
5. **XGBoost**: Gradient boosting with 200 estimators.
6. **SVR (RBF kernel)**: Support Vector Regression with standardization pipeline.

First Round: Default Parameters

All 12 model configurations were evaluated with default hyperparameters.

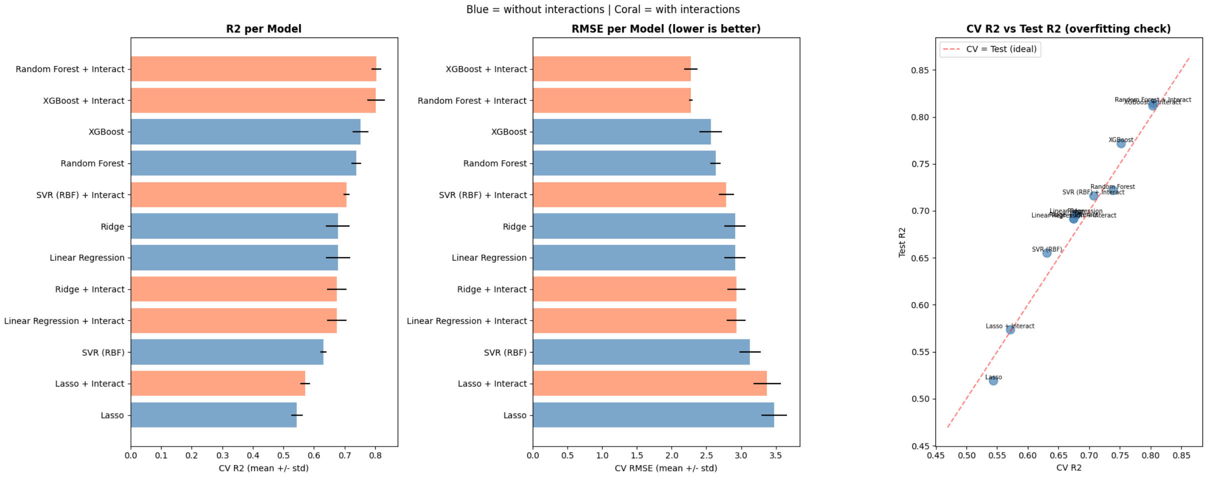

Figure 15: Model comparison: R^2 (left), RMSE (center), and overfitting check (right). Blue = without interactions; Coral = with interactions.

Table 6: First-round results ranked by CV R^2 (5-fold).

Model	CV R^2	±	CV RMSE	Test R^2	Test RMSE	Test MAE
Random Forest + Interact	0.8037	0.0164	2.278	0.8146	2.040	1.674
XGBoost + Interact	0.8030	0.0284	2.276	0.8123	2.053	1.604
XGBoost	0.7518	0.0260	2.564	0.7717	2.264	1.788
Random Forest	0.7383	0.0166	2.633	0.7219	2.499	2.000
SVR (RBF) + Interact	0.7069	0.0106	2.789	0.7162	2.524	2.081
Ridge	0.6786	0.0389	2.913	0.6961	2.612	2.110
Linear Regression	0.6785	0.0391	2.913	0.6962	2.612	2.109
Ridge + Interact	0.6747	0.0321	2.933	0.6920	2.630	2.097
Lin. Reg. + Interact	0.6747	0.0322	2.933	0.6916	2.631	2.100
SVR (RBF)	0.6311	0.0102	3.130	0.6556	2.781	2.333
Lasso + Interact	0.5714	0.0151	3.375	0.5737	3.094	2.540
Lasso	0.5439	0.0191	3.481	0.5193	3.285	2.766

Key observations from the first round:

- Tree-based models (Random Forest, XGBoost) significantly outperform linear models.
- Adding interaction features consistently improves tree-based models: RF improves from $R^2 = 0.74$ to 0.80; XGBoost from 0.75 to 0.80.
- Interaction features do not improve linear models (Ridge, Linear Regression), likely because the interactions are multiplicative while these models are additive.
- Lasso performs worst, likely because its L1 penalty aggressively shrinks coefficients that are small but collectively informative.
- The overfitting check (CV R^2 vs Test R^2) shows most models generalize well, with test scores slightly above CV scores.

Hyperparameter Tuning

The top 3 models were selected for hyperparameter optimization using RandomizedSearchCV with 50 iterations and 5-fold cross-validation (250 fits per model).

Table 7: Hyperparameter search spaces.

Parameter	Values
Random Forest	
n_estimators	{100, 200, 300, 500}
max_depth	{None, 10, 15, 20, 30}
min_samples_split	{2, 5, 10}
min_samples_leaf	{1, 2, 4}
max_features	{sqrt, log2, None}
XGBoost	
n_estimators	{100, 200, 300, 500}
max_depth	{3, 5, 7, 10}
learning_rate	{0.01, 0.05, 0.1, 0.2}
subsample	{0.7, 0.8, 1.0}
colsample_bytree	{0.7, 0.8, 1.0}
min_child_weight	{1, 3, 5}

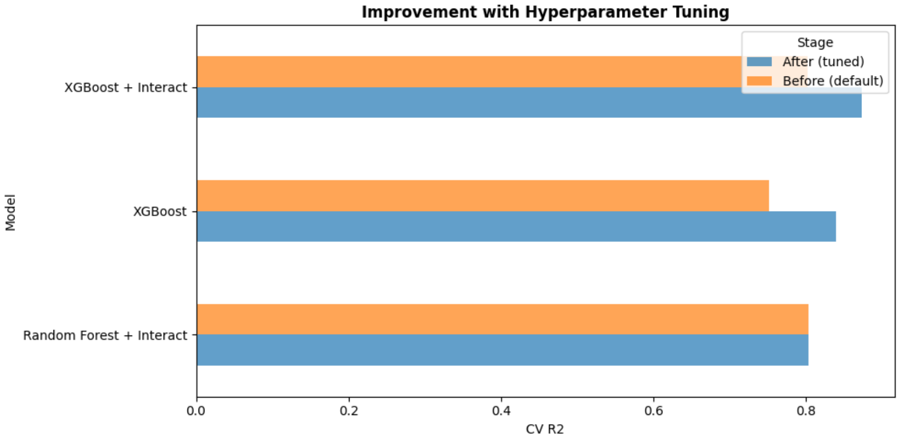

Figure 16: Improvement from hyperparameter tuning (default vs tuned CV R^2).

Table 8: Results after hyperparameter tuning.

Model	Best CV R^2	Test R^2	Test RMSE	Test MAE	Time (s)
XGBoost + Interact	0.8736	0.8838	1.615	1.304	317.5
XGBoost	0.8407	0.8438	1.873	1.552	260.2
Random Forest + Interact	0.8039	0.8168	2.028	1.660	466.0

XGBoost + Interact is the clear winner after tuning, with a substantial improvement from $R^2 = 0.80$ (default) to $R^2 = 0.87$ (tuned) in cross-validation, and $R^2 = 0.88$ on the held-out test set. The optimal hyperparameters are:

Table 9: Best hyperparameters for XGBoost + Interact.

Parameter	Value
n_estimators	200
max_depth	3
learning_rate	0.05
subsample	0.8
colsample_bytree	0.8
min_child_weight	3

The shallow tree depth (3) combined with conservative learning rate (0.05) and subsampling (0.8) indicates that the model benefits from gentle, regularized boosting rather than deep, aggressive fitting.

Residual Analysis

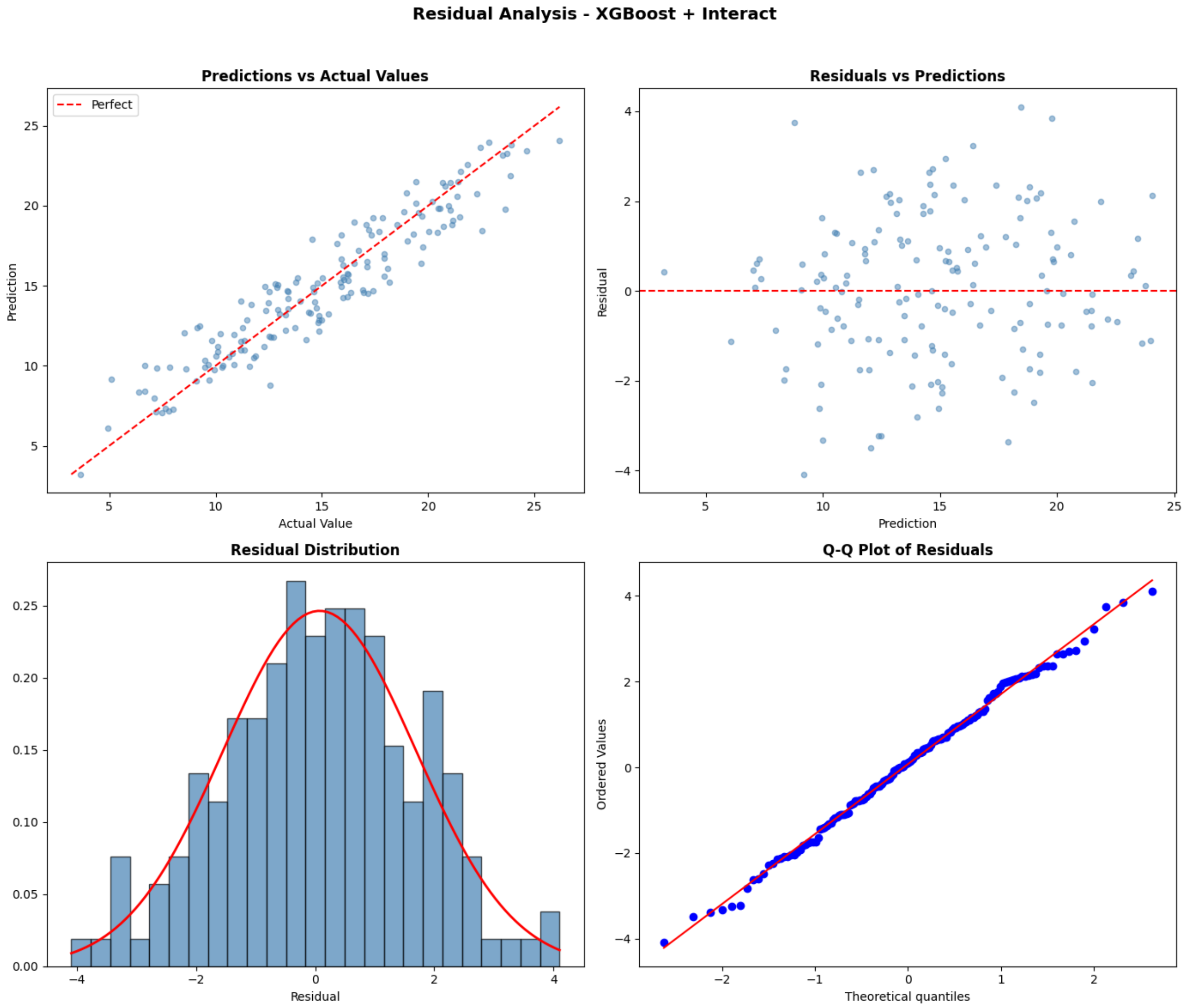

Figure 17: Residual analysis of XGBoost + Interact: predictions vs actual (top-left), residuals vs predictions (top-right), residual distribution (bottom-left), and Q-Q plot (bottom-right).

Table 10: Residual statistics for the best model.

Statistic	Value
Mean	0.0751
Standard Deviation	1.6184
Skewness	−0.0690
Kurtosis	−0.3019

The residual analysis confirms that the model behaves well:

- The predictions vs actual plot follows the diagonal closely, with no systematic bias.
- Residuals vs predictions show no heteroscedasticity or pattern, distributed symmetrically around zero.
- The residual distribution is approximately normal (skewness ≈ -0.07, kurtosis ≈ -0.30).
- The Q-Q plot follows the theoretical normal line closely, with minor deviations at the tails.
- The near-zero mean (0.075) indicates minimal systematic bias.

Feature Importance

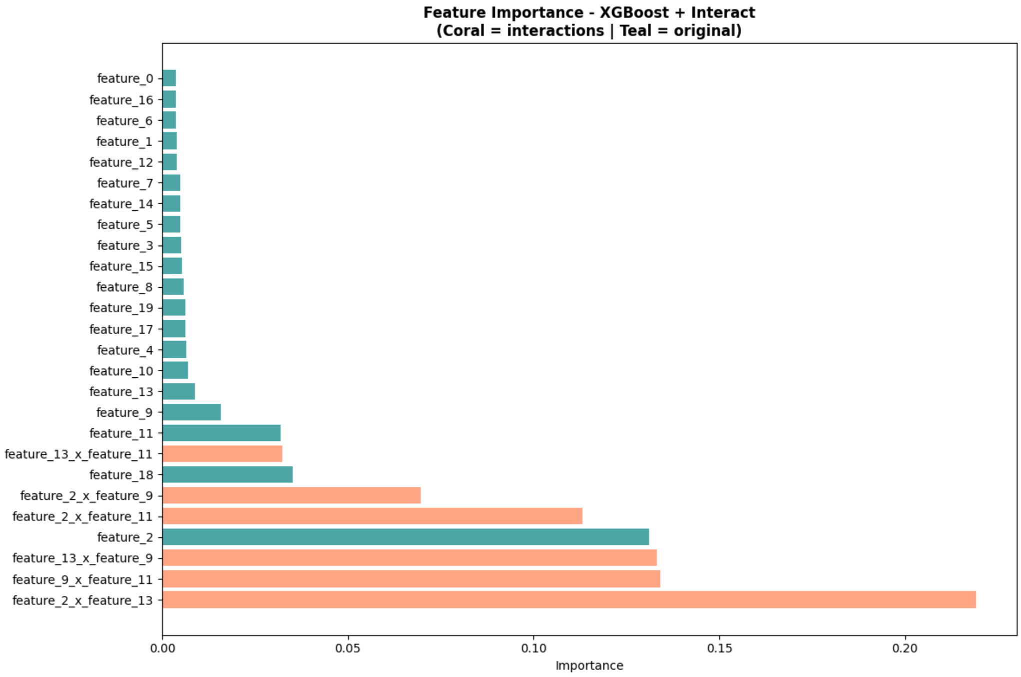

Figure 18: Feature importance from the tuned XGBoost + Interact model. Coral bars indicate interaction features; teal bars indicate original features.

Table 11: Top 10 features by XGBoost importance.

Feature	Importance
feature_2 × feature_13	0.2191
feature_9 × feature_11	0.1341
feature_13 × feature_9	0.1332
feature_2	0.1311
feature_2 × feature_11	0.1132
feature_2 × feature_9	0.0696
feature_18	0.0351
feature_13 × feature_11	0.0324
feature_11	0.0319
feature_9	0.0157

The feature importance analysis validates the EDA findings: the top feature is the interaction `feature_2` × `feature_13` (importance 0.219), confirming that the target has a strong multiplicative dependence on these features. Five of the top six features are interactions, collectively accounting for approximately 67% of the total importance. Among original features, only `feature_2` ranks in the top five.

Final Model and Blind Test Predictions

Final Retraining

The best model (XGBoost + Interact with tuned hyperparameters) was retrained on all 800 training samples to maximize the information available for prediction.

Table 12: Final model performance (retrained on 800 samples).

Metric	Value
CV R^2 (5-fold, 800 samples)	0.8842 ± 0.0127
Model type	XGBoost + Interactions
Number of features	26 (20 original + 6 interactions)
Model file size	0.22 MB

Blind Test Predictions

The trained model was applied to the 200-sample blind test set after adding the same 6 interaction features.

Table 13: Blind test prediction statistics.

Statistic	Value
Number of predictions	200
Mean	14.3110
Median	14.3471
Standard deviation	4.8628
Minimum	3.2953
Maximum	25.5090

Distribution Comparison: Training vs Predictions

To validate that the predictions are coherent, the distribution of blind test predictions was compared against the training target distribution using both histograms and Kernel Density Estimation (KDE).

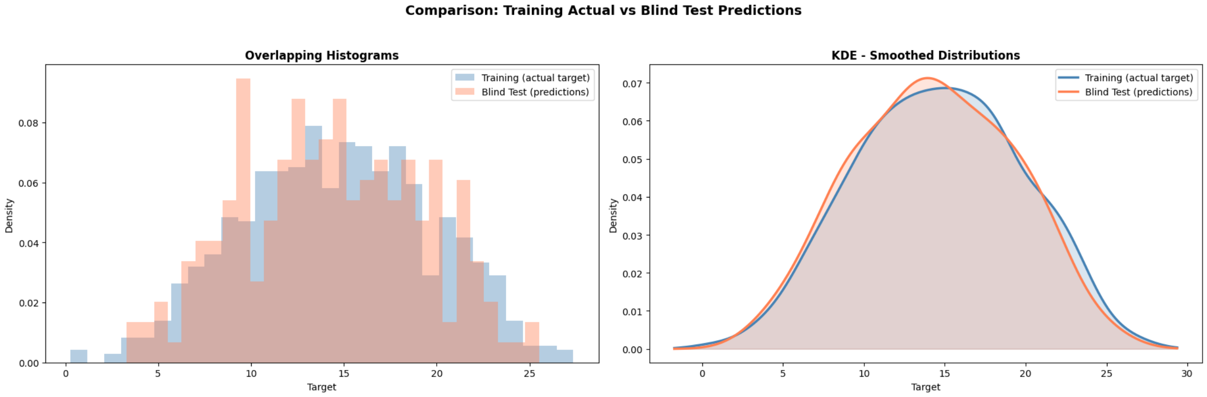

Figure 19: Overlapping histograms (left) and KDE smoothed distributions (right) comparing training target values and blind test predictions.

Table 14: Kolmogorov-Smirnov test: training target vs blind test predictions.

Statistic	Value
KS statistic	0.0500
p-value	0.8076
Conclusion	Distributions are similar (H_0 not rejected at $\alpha = 0.05$)

The KS test confirms that the predicted distribution is statistically indistinguishable from the training distribution ($p = 0.81$), providing confidence that the model generalizes correctly to the blind test data.

Model Persistence and Reproducibility

The final model and its configuration were saved for future use:

- `models/final_model.pkl`: Serialized XGBoost model (0.22 MB) saved with `joblib`.
- `models/model_config.pkl`: Configuration dictionary containing model name, interaction features, column names, best hyperparameters, and CV metrics.
- `predictions/blind_test_predictions.csv`: 200 predictions in the required format (single column `target_pred`).
- `mlruns/`: MLflow experiment directory with all runs, parameters, and metrics logged.

To generate predictions from the saved model:

```
import joblib, pandas as pd

model = joblib.load('models/final_model.pkl')
config = joblib.load('models/model_config.pkl')

new_data = pd.read_csv('new_data.csv')
# Add interaction features as needed
predictions = model.predict(new_data_processed)
```

Conclusion

This work presented a complete machine learning pipeline for a multivariate regression problem. The exploratory analysis revealed that multiplicative interactions between a small subset of features are the strongest predictors of the target variable. This insight directly informed the modeling strategy, leading to the selection of XGBoost with engineered interaction features as the optimal model.

The final model achieves a cross-validated R^2 of 0.884 ± 0.013 on the full training set and $R^2 = 0.884$ on the held-out test set. Residual analysis confirms no systematic bias, approximately normal residual distribution, and no heteroscedasticity. The Kolmogorov-Smirnov test validates that blind test predictions follow a distribution consistent with the training data.

The solution is packaged as a serialized `.pkl` model with MLflow experiment tracking, enabling straightforward deployment for regular batch predictions.