auto-insurance

customer lifetime value (CLV) prediction

Table Of Contents

project overview.

Background Overview

Customer Lifetime Value (CLV) quantifies the total revenue a customer generates for a business throughout their engagement. In the insurance sector, this metric is critical due to the industry's unique risk-pooling model: insurers aggregate risks from policyholders and reinvest collected premiums into income-generating assets (e.g., bonds, Treasury securities). Profitability hinges on two levers:

1. Premium income from clients
2. Investment returns on pooled funds

Insurers use CLV to segment customers and optimize risk-adjusted pricing, as well as to identify high-value customers and tailor retention strategies, directly linking premium stability to profitability.

Business Problems

Retention Strategy for High-Risk, High-Value Customers

- This matters as it is one of the industry's pain points.
- How can the insurance company optimize risk-adjusted pricing by balancing premium income and risk exposure to improve profitability?

Optimizing Risk-Adjusted Pricing Using Claims and Premium Metrics

- How can the insurance company optimize risk-adjusted pricing by balancing premium income and risk exposure to improve profitability?

Key Stakeholders

Actuaries

Risk assessment

Marketing

Retention campaigns

Executives

Profitability strategy

Goals of The Project.

Prioritize High-Value Customers

Identify customers with the highest predicted CLV to focus retention efforts.

Optimize Marketing Spend

Allocate budgets to acquire customers similar to high-CLV profiles (e.g., married, multi-policy holders).

Dynamic Pricing & Risk Adjustment

Adjust premiums for customers likely to yield long-term profitability (e.g., safe drivers with high CLV).

Analytical Approach

- Applied a combination of linear models (Ridge, Lasso, ElasticNet) and tree-based models (Random Forest, XGBoost, LightGBM, etc.) for regression.
- Performed feature engineering to improve model performance:
  - Applied log transformation to reduce skewness in the target variable.
  - Created interaction terms to capture nonlinear relationships between features.
- Evaluated models using cross-validation and compared performance before and after hyperparameter tuning.

Metric Evaluation

To evaluate model performance, several metrics were used:

MAE and RMSE

for measuring average and large errors.

MAPE

for understanding error in percentage terms.

R^2 and Adjusted R^2

to assess variance explained and model fit.

All metrics were computed using cross-validation to ensure robust and generalizable results.

insights deep dive.

Dataset Overview

Column Name	Description	Type
Vehicle Class	Vehicle class of each policy holders.	Categorical
Coverage	Coverage of policy (Basic, Extended, Premium)	Categorical
Renew Offer Type	Offer1, Offer2, Offer3	Categorical
EmploymentStatus	Retired, Employed, Disabled, Medical Leave, Unemployed	Categorical
Marital Status	Single, Married, Divorce	Categorical
Education	High School or Below, College, Bachelor, Master, Doctor	Categorical

Column Name	Description	Type
Number of Policies	Policies held by each customer	Float
Monthly Premium Auto	Recurring revenue per customer	Float
Total Claim Amount	Historical claims cost per customer	Float
Income	Annual income per customer	Float
Customer Lifetime Value	Net profit from the customer over the period of time	Float

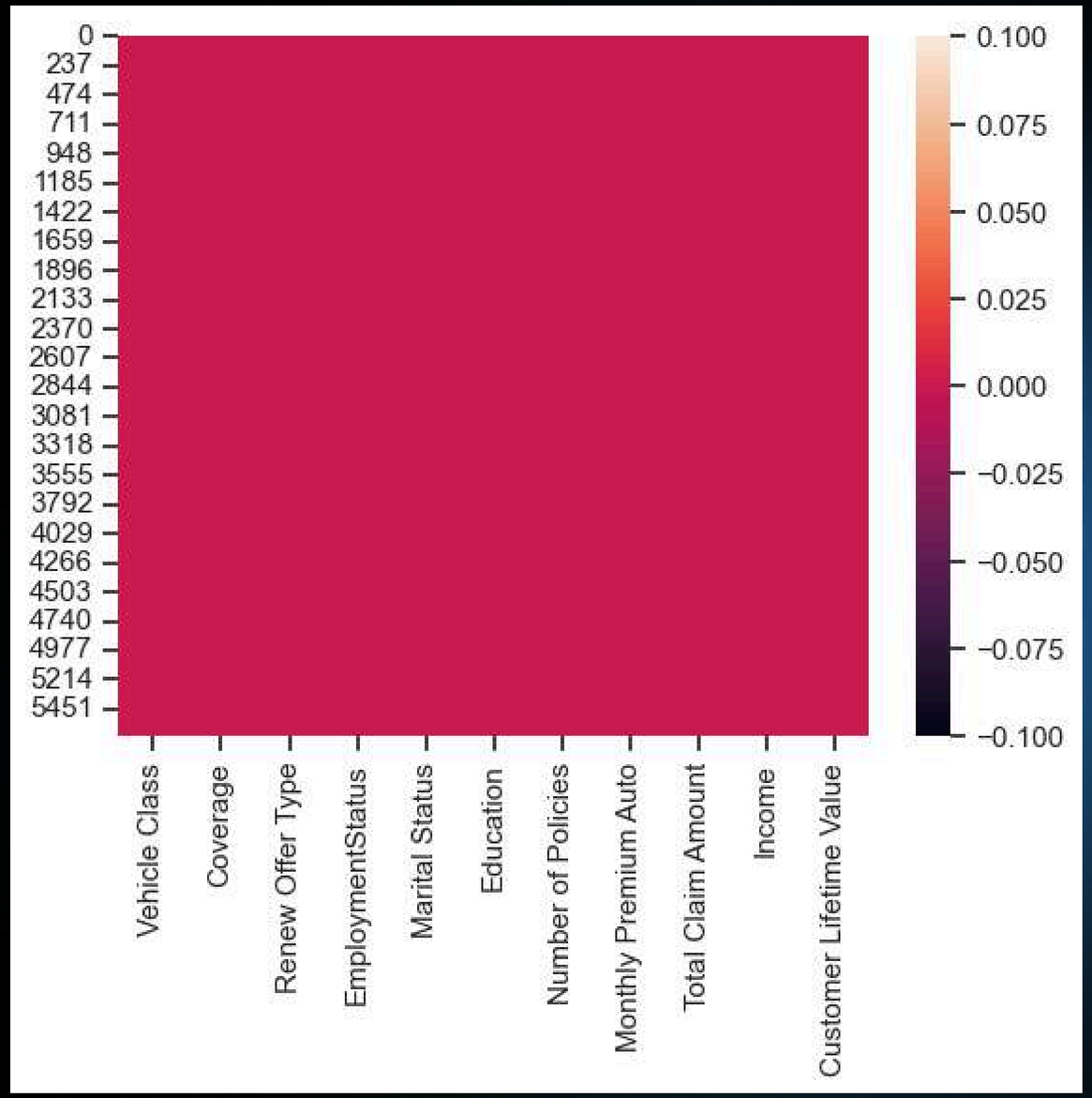
Target Variable

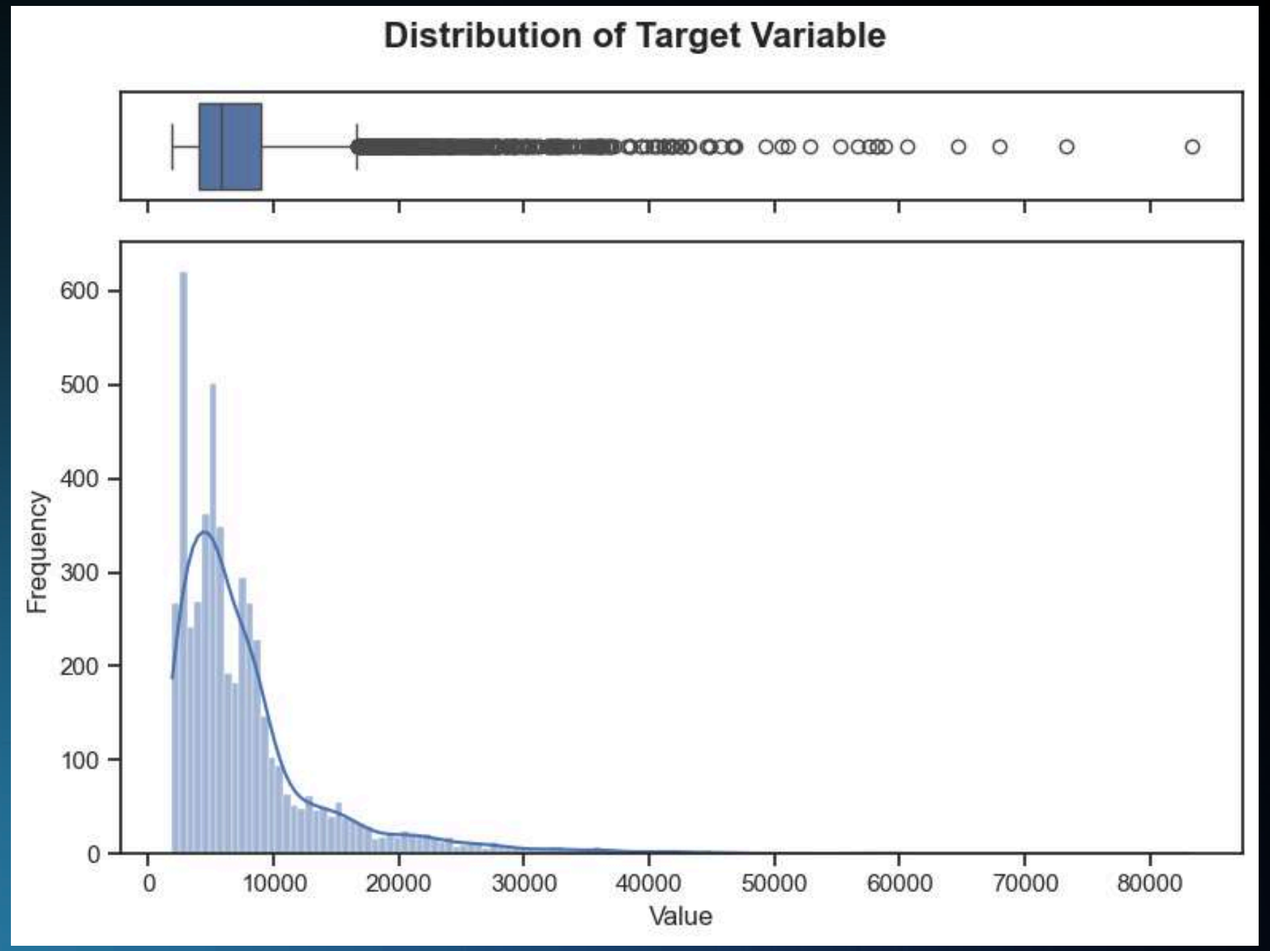
Missing & Duplicate Values Handling

Data was throughly checked for quality issues.

- **Missing values:** None were found across features.
- **Duplicated entries: 620 rows** of duplicated entries were identified and removed to prevent data leakage and model bias.

This ensured a clean, reliable dataset for model training and evaluation.



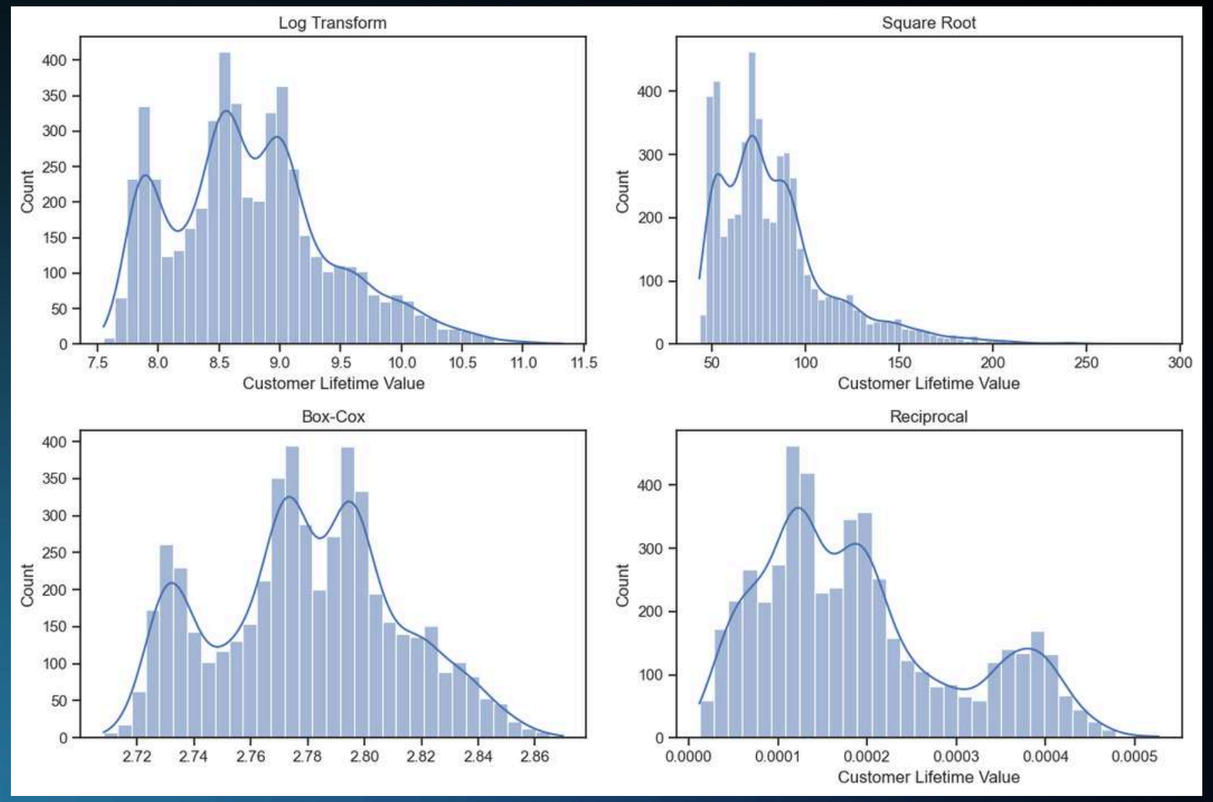


Distribution of Target Variable

The target variable shows a **right-skewed distribution** with clear presence of **outliers**.

- The **mean** is significantly higher than the median, indicating skewness.
- The **maximum value** is over **10× larger** than the 75th percentile.

To address this, log transformation was applied to normalize the distribution and reduce outlier impact on regression models.



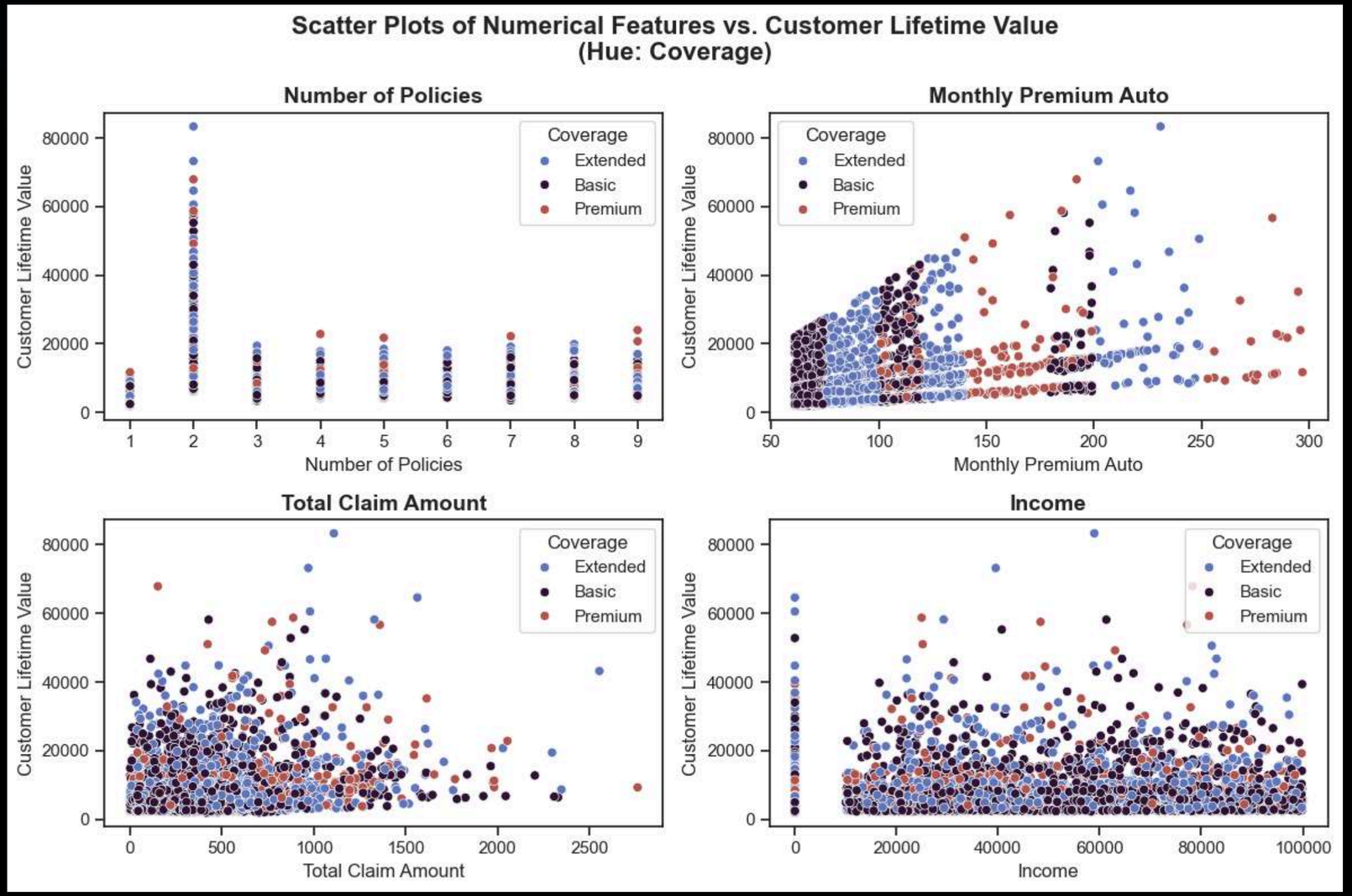
Target Normalization & Skewness Reduction

- Skewness measures distribution asymmetry. Ideally, | skewness | < 1 for regression.
- Transformations tested to reduce right-skew:
  - Log: Skewness reduced to 0.56 (acceptable for regression).
  - Square Root: Skewness = 1.597 (still moderate).
  - Box-Cox: Best result with 0.045 skewness (near normal).
  - Reciprocal: Skewness = 0.69 (improved but less interpretable).
- KS Test: All transformations failed strict normality test (p = 0.0), but skewness reduction is sufficient for regression.

Final Choice: Log Transformation — balances performance & interpretability, making it easier to explain to non-technical stakeholders.

Numerical Features vs. Customer Lifetime Value

- Monthly Premium Auto and Total Claim Amount show a positive relationship with Customer Lifetime Value (CLV), especially among customers with Extended and Premium coverage.
- Number of Policies peaks at 2 policies but shows no strong linear trend beyond that.
- Income has no clear relationship with CLV, suggesting low direct predictive power.
- Visual patterns help prioritize features for transformation or interaction in modeling.



Outliers Analysis

Why transform instead of remove?

- Outliers retained as they represent real customer behaviors (e.g., high CLV = valuable customers).
- Removing outliers risks losing insights and harms model generalization.

Transformations address skewness while preserving data:

1. Reduce skewness
   - Right-skewed distributions (e.g., CLV, claims) are normalized via log/sqrt transforms.
2. Preserve all data points
   - Retain high-value/high-risk customers critical for accurate predictions.
3. Improve model fit
   - Meets regression assumptions (linearity, homoscedasticity).
4. Align with business logic
   - Log/sqrt reflect insurance dynamics:
     - Large CLV/claims don't scale linearly.
     - Compresses extremes while retaining trends.

Feature	Total Outliers	Extreme Outliers
Number of Policies	228	0
Monthly Premium Auto	251	50
Total Claim Amount	216	51
Income	0	51
Customer Lifetime Value	449	51

Feature Construction

New features derived from existing features are:

Premium to Claim Ratio

Ratio of the monthly premium paid for auto insurance to the total claim amount paid over a specific period for a customer.

$$PCR = \frac{Monthly\ Premium\ Auto}{Total\ Claim\ Amount}$$

Premium to Income Ratio

The proportion of a customer's income that is allocated to their monthy auto insurance premium.

$$PIR = \frac{Monthly\ Premium\ Auto}{Income}$$

Claims per Policy

Calculates the average claim amount per policy held by a customer
Highlight the claim burden per policy.

$$CPP = \frac{Total\ Claim\ Amount}{Number\ of\ Policies}$$

Income per Policy

Measures the average income per policy held by a customer
Reflect the financial capacity supporting each policy.

$$IPP = \frac{Income}{Number\ of\ Policies}$$

Feature Engineering

Feature engineering is the process of creating, transforming, or selecting features from raw data to improve machine learning model performance.

Purpose:

- Reducing skewness for better model fit.
- Creating derived metrics to capture insurance-specific insights.
- Handling outliers to retain valuable data without removal.
- Improving predictor-target relationships for regression accuracy.

Feature Transformation

Log Transformation is applied to right-skewed features to normalize distributions and reduces outlier impact while preserving data.

Square Root Transformation is applied to moderatly skewed features.

Encoding

Converted nominal categorical features with One-Hot Encoding and ordered categoris with Ordinal Encoding.

Scaling

Ensures features contribute equally to model training, especially scale is critical for distanced-based and gradient-descent algorithms.

Used StandardScaler() to all numeric features, to ensure penalty terms in regularization.

model summary.

Models Applied as Baseline

Regressors used for this project are:

Linear Models

1. **Linear Regression** - Multiple linear model assuming a linear relationship between features and target.
2. **Ridge** - Linear model with L2 regularization to handle multicollinearity and overfitting.
3. **Lasso** - Linear model with L1 regularization, useful for feature selection by shrinking less important coefficients to zero.
4. **ElasticNet** - Combines L1 and L2 regularization, balancing feature selection and multicollinearity handling.

Tree-Based Models

1. **Decision Tree** - Single tree-based model that splits data based on feature thresholds, capturing non-linear relationships.
2. **Random Forest** - Ensemble of decision trees using bagging, reducing variance and improving robustness.
3. **Gradient Boosting** - Ensemble method that builds trees sequentially to minimize errors, excelling with complex patterns.
4. **AdaBoost** - Boosting ensemble that adjusts weights of misclassified instances, enhancing focus on difficult cases.

Models Applied as Baseline

Regressors used for this project are:

Boosting Models (Advanced Ensembles)

1. **XGBoost** - Optimized gradient boosting framework with regularization and parallel processing, tuned for mean absolute error (MAE).
2. **LightGBM** - Gradient boosting with a focus on speed and efficiency using histogram-based learning, suitable for large datasets.

Kernel-Based Models

Support Vector Regression (SVR) - Uses kernel tricks (e.g., RBF) to capture non-linear relationships, effective with small-to-medium datasets but computationally intensive.

Instance-Based Models

K-Nearest Neighbors (KNN) - Predicts based on the average of k nearest neighbors, sensitive to feature scaling and distance metrics.

Baseline Model Summary

After applying Cross Validation to the baseline models, Gradient Boosting is the top model, suitable for the business problems while Decision Tree underperforms, emphasizing the value of ensemble methods within the top 5 models.

Regressor	MAE	RMSE	R^2
Gradient Boosting	1575.13	4013.33	0.670
Random Forest	1593.77	4071.86	0.661
LightGBM	1601.38	4079.24	0.659
XGBoost	1683.66	4191.38	0.640
Decision Tree	1971.26	5426.46	0.397

conclusion and recommendation.

Conclusion

Data

- DataEngineered features (e.g., Claims_per_Policy, log-transformed Monthly_Premium_Auto) reduced skewness (CLV from 3.06 to 0.56), enhancing model fit.
- Ordinal encoding (Coverage, Education) preserved domain hierarchies for segmentation.
- Residual analysis shows heteroskedasticity; unmodeled nonlinear effects persist.

Model

- Gradient Boosting (R^2: 0.670, MAE: $1575.13, RMSE: $4013.33) outperformed others (e.g., Random Forest: 0.661, Decision Tree: 0.397).
- Default settings beat tuned versions; cross-validation confirmed stability with low bias.
- Residuals centered at zero, but Q-Q plot deviations suggest nonlinearity or heteroskedasticity.

Business

- Model supports segmentation (high-value clients), pricing (MAE $1575), and retention (MAPE 10.7%) for profitability.
- Reliable CLV predictions enhance premium stability and risk management.
- Potential pricing errors ($1575) and residual issues limit precision.

Recommendations

Explore interaction terms or Box-Cox transformation (skewness to 0.045) for better data representation.

Tune Gradient Boosting (e.g., learning_rate, n_estimators) or ensemble with Random Forest, as well as try other ensemble methods.

Implement Gradient Boosting for segmentation/pricing; refine with feature selection and advanced ensembles for retention optimization.

Thank You

Links:

Github Repo

Dataset