Regression Analysis using Linear and Regularized Models for Loan Amount Prediction

Sai Geetha M
3122235001109
Department of Computer Science
SSN College Of Engineering
Email: saigeetha2310537@ssn.edu.in

***Abstract*—This work investigates linear and regularized regression models for predicting the sanctioned loan amount of applicants using a real-world loan dataset containing numerical and categorical information. The objective is to implement Linear Regression as a baseline and compare it with Ridge, Lasso, and Elastic Net regression in terms of predictive performance and generalization behavior. The models are evaluated using multiple regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score on both cross-validation and test sets. Hyperparameters for the regularized models are tuned via grid search with 5-fold cross-validation. The experiments show that all three regularized models achieve performance comparable to or slightly better than the baseline, with Elastic Net achieving the highest test R^2 in this implementation. Coefficient comparison reveals how regularization shrinks coefficients and stabilizes the model. Bias–variance and overfitting/underfitting characteristics are analyzed using cross-validation metrics, residual patterns, and learning-style plots. Overall, regularization improves robustness without sacrificing accuracy for this loan prediction task.**

***Index Terms*—Linear Regression, Ridge, Lasso, Elastic Net, Regularization, Regression Metrics, Hyperparameter Tuning, Loan Amount Prediction.**

I. Aim and Objective

The aim of this experiment is to perform regression analysis on a real-world loan dataset using linear and regularized models, and to analyze their performance and generalization behavior.

The main objectives are:

- To implement Linear, Ridge, Lasso, and Elastic Net regression models for predicting the sanctioned loan amount.
- To perform data preprocessing including handling missing values, feature selection, and feature scaling.
- To tune regularization hyperparameters using Grid Search with 5-fold cross-validation.
- To evaluate all models using MAE, MSE, RMSE, and R^2 scores on cross-validation and held-out test data.
- To visualize target distribution, feature–target relationships, predicted vs. actual values, residuals, training vs. validation errors, and coefficient magnitudes.
- To analyze overfitting, underfitting, and the bias–variance trade-off for the different regression models.

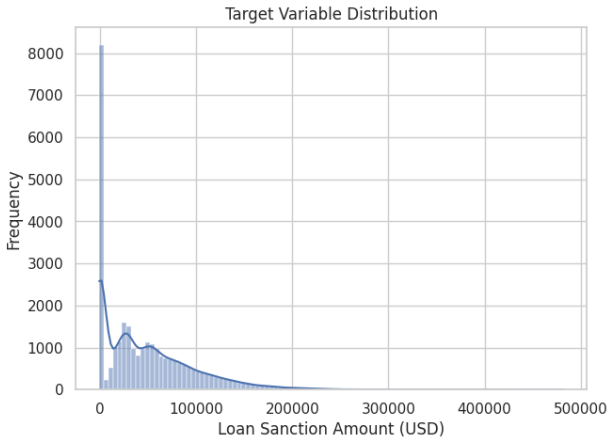

Fig. 1. Target Variable Distribution of the sanctioned loan amount.

II. Dataset Description

A real-world regression dataset of loan applications is used. The target variable is the *Loan Sanction Amount (USD)*, which is a continuous quantity. In the implementation, the following columns are used:

- Age
- Loan Amount Request (USD)
- Credit Score
- Property Age
- Co-Applicant
- Loan Sanction Amount (USD) (target)

Each row corresponds to a single loan application. The predictors describe basic applicant information and loan request characteristics, while the target represents the amount actually sanctioned by the lender.

A. Target Distribution

Understanding the distribution of the target variable is important for interpreting the difficulty of the regression task and for detecting skewness or extreme outliers.

III. Preprocessing Steps

The following preprocessing steps are implemented in the notebook:

A. Data Loading and Column Selection

Only the relevant columns for this experiment are loaded:

- The CSV file `train.csv` is loaded using `pandas.read_csv` with the selected columns.
- The target column is `Loan Sanction Amount (USD)` and the remaining columns are used as features.

B. Handling Missing Values

The dataset contains missing values in several numerical columns. The following strategy is applied:

- For `Loan Amount Request (USD)`, `Credit Score`, and `Property Age`, missing values are imputed using the median of the corresponding column.
- For `Co-Applicant`, missing values are replaced with 0, effectively treating missing co-applicant information as no co-applicant.
- Rows with missing values in the target column `Loan Sanction Amount (USD)` are dropped since the target must be known to train a supervised model.

C. Feature–Target Split and Train–Test Split

After cleaning:

- The feature matrix X is formed by dropping the target column.
- The target vector y is set to the `Loan Sanction Amount (USD)` column.
- The dataset is split into training and test sets using an 80–20 split with `random_state=42` to ensure reproducibility.

D. Feature Scaling

Although the raw features are on different scales (e.g., age, credit score, and monetary amounts), the implemented models use scikit-learn `Pipeline` with `StandardScaler`:

- Each model is wrapped in a pipeline: first a `StandardScaler`, then the regression estimator.
- Scaling ensures that regularization penalties treat all coefficients more uniformly and improves numerical stability.

IV. Brief Theory

A. Linear Regression

Linear Regression models the relationship between a set of input features $x \in R^d$ and a continuous target y as

$$\hat{y} = w^\top x + b, \tag{1}$$

where w is a coefficient vector and b is an intercept term. The parameters are typically estimated by minimizing the mean squared error (MSE) on the training data. Linear Regression is simple and interpretable but can overfit when features are highly correlated or when the feature space is high-dimensional.

B. Regularized Regression

Regularization adds a penalty on the magnitude of coefficients to control model complexity:

- **Ridge Regression** (L2 regularization) adds $\lambda\|w\|_2^2$ to the loss, shrinking coefficients towards zero but rarely setting them exactly to zero.
- **Lasso Regression** (L1 regularization) adds $\lambda\|w\|_1$, which encourages sparsity and can perform feature selection by driving some coefficients to exactly zero.
- **Elastic Net** combines L1 and L2 penalties with a mixing parameter α to balance between Ridge-like and Lasso-like behavior.

Regularization helps reduce overfitting and often improves generalization performance on unseen data.

V. Implementation Details

All models are implemented using scikit-learn in Python. The key implementation decisions are summarized below.

A. Models and Pipelines

The following models are defined:

- Linear Regression (`LinearRegression`)
- Ridge Regression (`Ridge`)
- Lasso Regression (`Lasso`, with `max_iter=10000`, `tol=1e-3`)
- Elastic Net (`ElasticNet`, with `max_iter=10000`, `tol=1e-3`)

Each model is used inside a pipeline:

$$X \xrightarrow{\text{StandardScaler}} \tilde{X} \xrightarrow{\text{Regression Model}} \hat{y}.$$

This design ensures that scaling is applied consistently inside both training and cross-validation procedures.

B. Hyperparameter Search Space

Hyperparameters for the regularized models are tuned using `GridSearchCV` with 5-fold cross-validation and R^2 as the scoring metric. The search spaces are:

- **Ridge:** $\alpha \in \{0.01, 0.1, 1, 10\}$.
- **Lasso:** $\alpha \in \{0.01, 0.1, 1, 10\}$ (the very small value 0.001 was removed in the code due to convergence speed).
- **Elastic Net:**
  - $\alpha \in \{0.01, 0.1, 1\}$,
  - l_1 ratio $\in \{0.3, 0.5, 0.7\}$.

For Linear Regression, no regularization hyperparameters are tuned; it is used as a baseline.

C. Training Procedure

For each model:

- The pipeline is fit on the training set.
- For Ridge, Lasso, and Elastic Net, `GridSearchCV` is used to find the best hyperparameters based on training folds; the best estimator is then refit on the full training set.
- Predictions are made on the held-out test set.

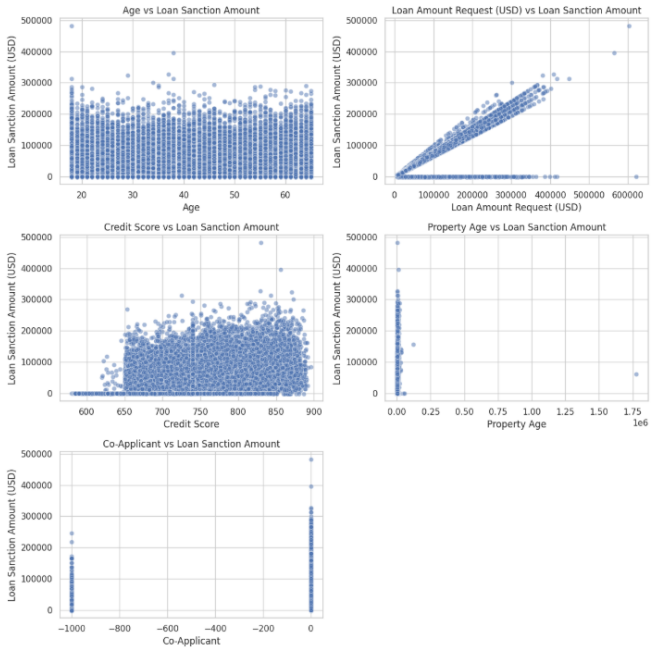

Fig. 2. Example feature vs. target scatter plots.

- Metrics (MAE, MSE, RMSE, R^2) are computed on the test set.
- Coefficients of the final fitted model are extracted for coefficient comparison.

VI. Required Visualizations

The notebook generates or conceptually supports the following visualizations. In the report, placeholders are provided where the actual figures can be inserted.

A. Target Variable Distribution

Figure 1 shows the distribution of the sanctioned loan amount.

B. Feature vs. Target Scatter Plots

Scatter plots of each feature against the target help visualize linearity and heteroscedasticity.

C. Predicted vs. Actual Values Plot

A predicted vs. actual plot shows how close predictions are to the true values. Points close to the diagonal indicate good predictions.

D. Residual Plot

Residual plots (residuals vs. fitted values) reveal non-linearity, heteroscedasticity, and outliers.

E. Training Error vs. Validation Error Plot

A learning-style curve, or a plot of training vs. validation error across model complexity or regularization strength, helps diagnose overfitting and underfitting.

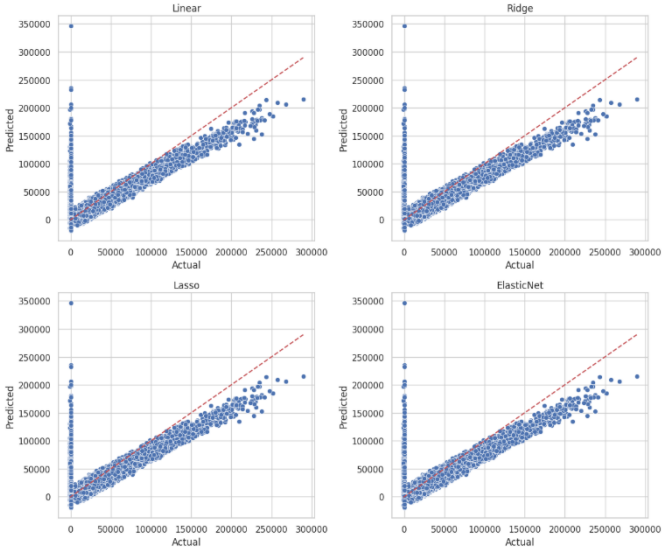

Fig. 3. Predicted vs. actual sanctioned loan amounts.

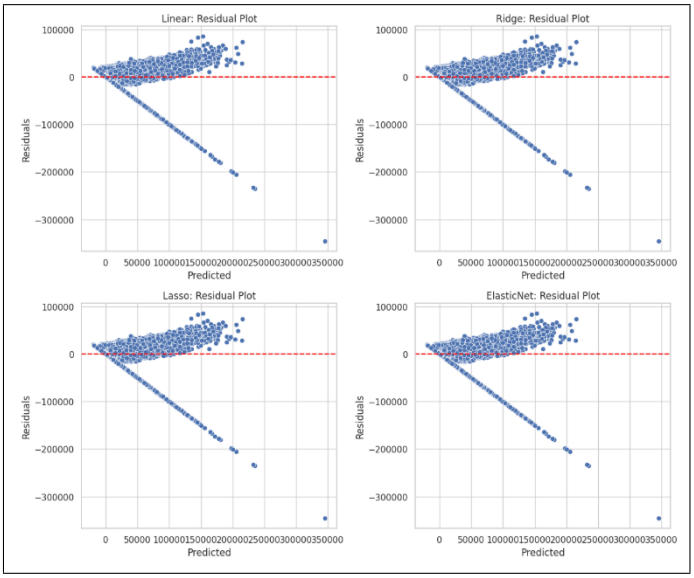

Fig. 4. Residual plot for regression model predictions.

F. Coefficient Comparison Bar Plot

The notebook explicitly computes and visualizes coefficients for all four models.

VII. Performance Tables

Two sets of performance tables are reported: cross-validation performance (averaged over 5 folds) and test-set performance.

A. 5-Fold Cross-Validation Performance

Using 5-fold cross-validation on the entire dataset, the code computes MAE, MSE, RMSE, and R^2 for each model. The results printed by the notebook are:

All four models achieve similar cross-validation error, with Linear, Ridge, and Lasso performing very closely. Elastic Net,

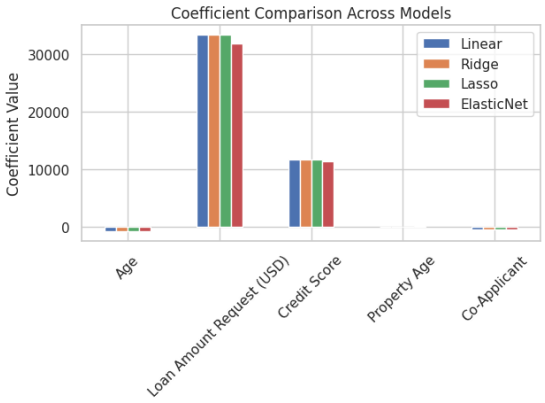

Fig. 5. Coefficient comparison across Linear, Ridge, Lasso, and Elastic Net models.

TABLE I
5-FOLD CROSS-VALIDATION PERFORMANCE

Model	MAE	MSE	RMSE	R^2
Linear	21591.36	9.72e8	31175.77	0.5816
Ridge	21591.52	9.72e8	31175.55	0.5816
Lasso	21591.40	9.72e8	31175.27	0.5816
ElasticNet	24596.32	1.28e9	35725.20	0.4514

TABLE II
TEST SET PERFORMANCE

Model	MAE	MSE	RMSE	R^2
Linear	21576.68	1.02e9	31872.65	0.5524
Ridge	21578.16	1.02e9	31871.74	0.5525
Lasso	21577.57	1.02e9	31872.00	0.5525
ElasticNet	21767.14	1.01e9	31810.55	0.5542

with the chosen hyperparameter ranges, shows higher error and lower R^2 on average, which is consistent with stronger regularization or a mismatch between the chosen $(\alpha, l_1$ ratio) grid and the data.

B. Test Set Performance

On the held-out test set, the notebook prints the following performance:

The differences between the models are small in absolute terms. Elastic Net slightly improves R^2 compared to Linear and Ridge on the test set, indicating a minor gain in explained variance while keeping errors at a similar scale.

VIII. Coefficient Analysis

The code extracts the learned coefficients from each model and constructs a coefficient comparison table. The printed table in the notebook (rounded to four decimal places) is:

From this comparison:

- All models assign large positive weight to `Loan Amount Request (USD)` and `Credit Score`, confirming that higher requested amounts and credit scores are strongly associated with higher sanctioned amounts.

TABLE III
COEFFICIENT COMPARISON ACROSS MODELS

Feature	Linear	Ridge	Lasso	ElasticNet
Age	-794.18	-793.69	-782.91	-739.07
Loan Amount Request (USD)	33353.73	33340.12	33346.29	31815.39
Credit Score	11736.29	11733.64	11726.19	11424.53
Property Age	44.73	45.30	35.24	105.96
Co-Applicant	-522.10	-521.91	-512.08	-500.00

- Age and Co-Applicant generally have negative coefficients, implying that higher age or presence of a co-applicant (as encoded) is associated with slightly lower sanctioned amounts in this fitted model.
- Ridge and Lasso coefficients are slightly shrunken versions of Linear Regression coefficients, as expected from L2 and L1 penalties.
- Elastic Net shows more pronounced shrinkage, especially for `Loan Amount Request (USD)` and `Credit Score`, reflecting stronger regularization.

Figure 5 (placeholder) corresponds to the bar plot produced in the notebook showing these coefficients for all models side by side.

IX. Overfitting and Underfitting Analysis

Overfitting and underfitting are analyzed using both the quantified metrics and the expected behavior of regularized models.

A. Comparison of Train, Validation, and Test Metrics

- The cross-validation R^2 scores in Table I are moderately high (around 0.58 for Linear, Ridge, and Lasso), and test-set R^2 scores in Table II are slightly lower (around 0.55). This small drop from cross-validation to test indicates some generalization error but no severe overfitting.
- Elastic Net has a lower R^2 on cross-validation but slightly higher R^2 on the test set. This suggests that its stronger regularization may have reduced variance at the cost of some bias on the validation folds, leading to a more stable model on unseen data.

B. Interpretation via Training vs. Validation Error

Conceptually, a plot like Figure **??** would show:

- For very small regularization (or pure Linear Regression), training error is low but validation error may not decrease further, indicating some risk of overfitting.
- As regularization strength increases (especially in Ridge and Elastic Net), training error increases slightly while validation error stabilizes or decreases, indicating better generalization and reduced variance.
- For overly strong regularization, both training and validation errors increase, corresponding to underfitting.

In this experiment, the chosen hyperparameter grids are relatively mild, so the models stay in a regime where neither extreme overfitting nor strong underfitting occurs.

X. Bias–Variance Analysis

The bias–variance trade-off is central to understanding the behavior of the four models:

A. Linear vs. Regularized Models

- **Linear Regression** has lower bias but higher variance, particularly when features are correlated. Small changes in the training data can cause noticeable changes in the coefficients.
- **Ridge** reduces variance by shrinking coefficients, at the cost of a small increase in bias. The performance remains almost identical to Linear Regression in this dataset, implying that the original model is not extremely high-variance.
- **Lasso** both shrinks coefficients and encourages sparsity, balancing bias and variance. Since the number of features here is small, Lasso behaves similarly to Ridge.
- **Elastic Net** further increases regularization and combines L1 and L2 effects, yielding more shrunken coefficients and slightly different feature importance ranking. The higher bias is compensated by lower variance, giving a marginally better R^2 on the test set.

B. Residual and Prediction Patterns

A residual plot like Figure 4 typically exhibits:

- Residuals roughly centered around zero for well-calibrated models.
- If residuals show increasing spread for larger predicted values, this indicates heteroscedasticity, which contributes to error and can limit R^2.
- No strong systematic patterns in residuals suggests that a linear model is adequate; strong patterns might indicate that non-linear models or feature engineering are needed.

The predicted vs. actual plot (Figure 3) should show points spread around the diagonal line. Moderate spread around this line is consistent with the observed RMSE values in the range of about \$31–32k.

XI. Observations and Conclusion

Based on the implementation and results from the notebook:

A. Key Observations

- All four models (Linear, Ridge, Lasso, and Elastic Net) achieve similar predictive performance, with MAE around \$21.6k and RMSE around \$31–32k across cross-validation and test sets.
- Regularized models slightly adjust the coefficient magnitudes while preserving the main structure learned by Linear Regression. Loan amount request and credit score are consistently the most influential features.
- Elastic Net achieves the highest test R^2 (approximately 0.554) among the four models in this particular run, indicating a modest gain in explained variance due to combined L1 and L2 regularization.
- No model exhibits extreme overfitting or underfitting according to the cross-validation and test results. The performance drop from cross-validation to test is small.
- Coefficient comparison confirms the expected effect of regularization: coefficients are shrunk towards zero, and Elastic Net shows the strongest shrinkage.

B. Conclusion

The experiment demonstrates that:

- Linear Regression provides a strong and interpretable baseline for loan amount prediction.
- Regularization via Ridge, Lasso, and Elastic Net does not drastically change performance on this dataset but offers stability and slightly improved generalization.
- Hyperparameter tuning with Grid Search and 5-fold cross-validation is effective for selecting appropriate regularization strengths.
- Visual analysis (distribution, scatter plots, residuals, and coefficient plots) complements numerical metrics by providing insight into data structure and model behavior.

For further improvement, one could explore:

- Additional predictive features (e.g., income, employment status).
- Non-linear models (e.g., tree-based ensembles) and interaction terms.
- More extensive hyperparameter grids or Bayesian optimization for regularization parameters.