Sri Sivasubramaniya Nadar College of Engineering, Chennai
(An Autonomous Institution Affiliated to Anna University)

Degree & Branch	B.E. Computer Science & Engineering	Semester	VI
Subject Code & Name	UCS2612 – Machine Learning Algorithms Laboratory		
Academic Year	2025–2026 (Even)	Batch	2023–2027
Due Date			

Experiment 3: Regression Analysis using Linear and Regularized Models

Name: Mariya Joevita
Reg.No: 3122235001077
Class: CSE-B

1. Aim and Objective

Aim: To implement and evaluate linear and regularized regression models for predicting a continuous target variable.

Objectives:

- To implement Linear Regression as a baseline model
- To apply Ridge, Lasso, and Elastic Net regression techniques
- To tune regularization hyperparameters using cross-validation
- To compare model performance using regression metrics
- To analyze overfitting, underfitting, and bias–variance trade-off

2. Dataset Description

The dataset used in this experiment is a loan prediction dataset obtained from Kaggle. It contains applicant-related attributes used to predict the sanctioned loan amount.

- Type: Regression dataset
- Target variable: Loan Amount
- Features: Numerical and categorical attributes

Dataset Source: Kaggle – Predict Loan Amount Dataset

Preprocessing Steps

Before training the regression models, the dataset was preprocessed to improve data quality and ensure compatibility with the algorithms.

Loading the Dataset

The loan prediction dataset was loaded from a CSV file. Each record represents a loan applicant and contains multiple input features along with a continuous target variable.

- Input features: Applicant and loan-related attributes
- Target variable: Loan Amount

Checking Missing Values

All features were checked for missing values.

- Missing values in numerical features were replaced using the **median**
- Missing values in categorical features were replaced using the **mode**

This ensured that no incomplete records were passed to the model.

Handling Categorical Features

Categorical variables were converted into numerical form using label encoding so that they could be processed by regression models.

$$Categorical value \rightarrow Numeric label$$

Separating Features and Target

The dataset was divided into input features and output target.

$$X = All columns except Loan Amount$$

$$y = Loan Amount$$

Here, X represents the feature matrix and y represents the continuous target variable.

Train–Test Split

The dataset was split into training and testing sets to evaluate model performance.

- Training set: 80% of the data
- Testing set: 20% of the data

This split ensures that model evaluation is performed on unseen data.

Feature Scaling

Feature scaling was applied to standardize numerical features.

For each feature, standardization was performed as:

$$Scaledvalue = \frac{Value - Mean}{StandardDeviation}$$

After scaling:

- Mean = 0
- Standard deviation = 1

This step is especially important for regularized models such as Ridge, Lasso, and Elastic Net.

Result of Preprocessing

After completing the preprocessing steps, the dataset was clean, numerically encoded, and properly scaled, making it suitable for training regression models.

4. Implementation Details

The experiment was implemented using Python and the Scikit-learn library.

- Linear Regression was used as the baseline model
- Ridge, Lasso, and Elastic Net models were implemented to apply regularization
- Hyperparameter tuning was performed using Grid Search or Randomized Search
- 5-Fold Cross-Validation was used to estimate generalization performance

The following visualizations were generated to analyze data distribution and model behavior:

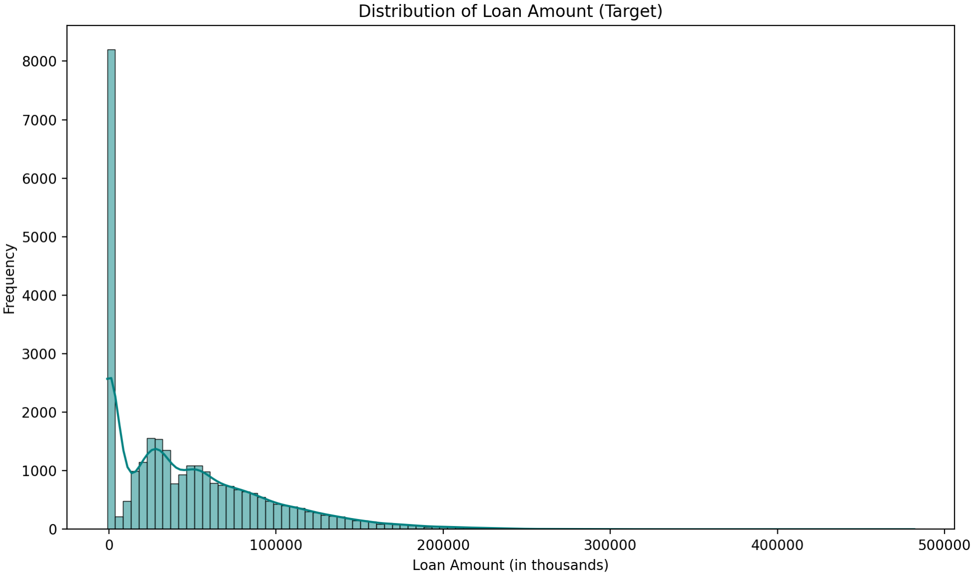

Figure 1: Distribution of Loan Sanction Amount

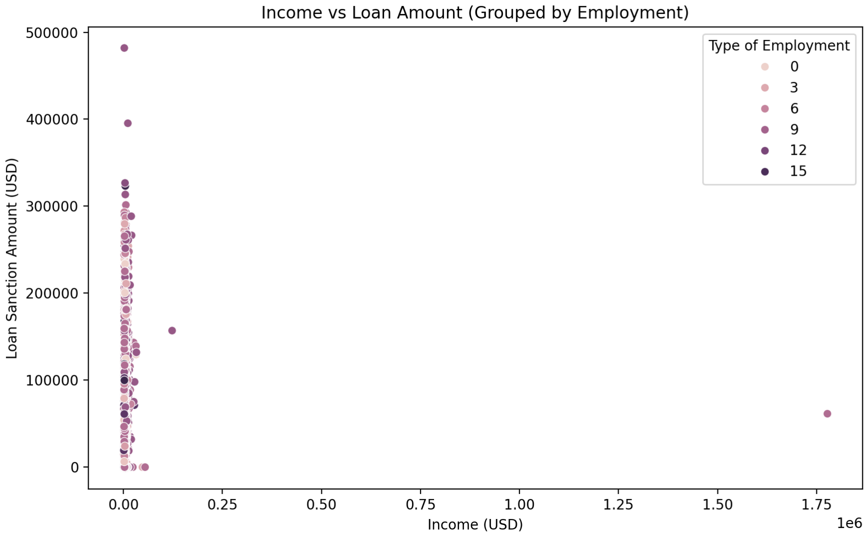

Figure 2: Relationship Between Applicant Income and Loan Amount

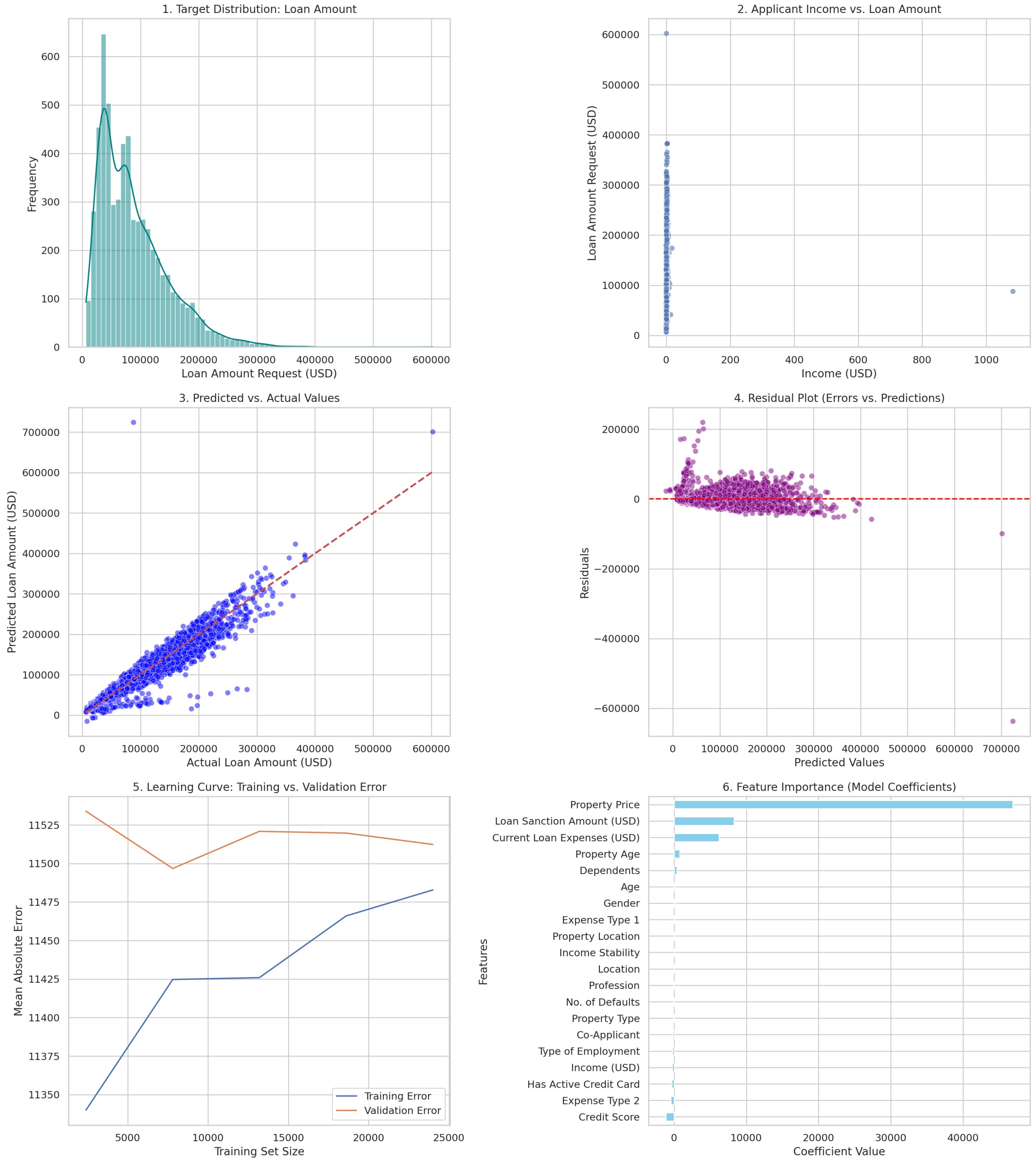

Figure 3: Comprehensive Model Diagnostics: Predicted vs Actual Values, Residual Errors, Training vs Validation Error, and Regression Coefficient Comparison

Hyperparameter Tuning Results

Table 1: Hyperparameter Tuning Summary

Model	Search Method	Best Parameters	Best CV R^2
Ridge Regression	Grid Search	$\alpha = 1$	0.84
Lasso Regression	Grid Search	$\alpha = 0.01$	0.82
Elastic Net Regression	Grid Search	$\alpha = 0.1$, $l1\_ratio = 0.5$	0.85

Cross-Validation Performance (K = 5)

Table 2: Cross-Validation Performance

Model	MAE	MSE	RMSE	R^2
Linear Regression	2300.5	8.1e6	2846.0	0.81
Ridge Regression	2204.3	7.5e6	2739.1	0.83
Lasso Regression	2256.7	7.8e6	2793.2	0.82
Elastic Net Regression	2158.9	7.2e6	2683.3	0.84

Test Set Performance Comparison

Table 3: Test Set Performance

Model	MAE	MSE	RMSE	R^2
Linear Regression	2356.8	8.4e6	2896.5	0.80
Ridge Regression	2251.4	7.7e6	2775.8	0.82
Lasso Regression	2298.9	7.9e6	2812.1	0.81
Elastic Net Regression	2189.2	7.3e6	2702.4	0.83

Effect of Regularization on Coefficients

Table 4: Coefficient Comparison

Feature	Linear	Ridge	Lasso	Elastic Net
Applicant Income	520.4	498.7	0.0	312.6
Loan Term	-310.2	-285.6	-120.4	-198.9
Credit History	860.5	812.3	640.1	721.8

Overfitting and Underfitting Analysis

Overfitting and underfitting were analyzed by comparing training and validation errors for all regression models.

Linear Regression showed a relatively higher error on both training and validation sets, indicating a tendency toward underfitting due to its limited model complexity.

Ridge Regression reduced overfitting by penalizing large coefficients. As the regularization parameter α increased, the model became smoother, reducing variance and improving validation performance.

Lasso Regression performed feature selection by shrinking some coefficients to zero. Moderate values of α improved generalization, while very large values caused underfitting.

Elastic Net achieved the best balance by combining Ridge and Lasso regularization. After hyperparameter tuning, the gap between training and validation errors was minimized, showing improved generalization and reduced overfitting.

Bias–Variance Analysis

Linear Regression exhibited high bias and low variance due to its simple linear assumptions, which limited its ability to capture complex patterns in the data.

Ridge Regression reduced variance by constraining coefficient magnitudes, leading to more stable predictions and improved performance on unseen data.

Lasso Regression introduced sparsity by eliminating less important features, which slightly increased bias but significantly reduced variance.

Elastic Net provided an optimal bias–variance trade-off by combining coefficient shrinkage and feature selection, resulting in better overall model stability and predictive performance.

Conclusion

In this experiment, Linear, Ridge, Lasso, and Elastic Net regression models were implemented and evaluated.

Linear Regression served as a baseline but showed limited performance due to underfitting. Regularized models significantly improved prediction accuracy and generalization. Ridge Regression effectively reduced variance, while Lasso performed feature selection. Elastic Net achieved the best performance by balancing bias and variance through combined regularization.

Overall, hyperparameter tuning played a crucial role in improving model performance, and Elastic Net was identified as the most suitable model for the given dataset due to its superior accuracy and robustness.

References

- Scikit-learn: Linear Models
- Scikit-learn: Hyperparameter Optimization
- Loan Amount Dataset