

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	26/01/2026	

R Padmashri
 3122 23 5001 093

Experiment 3: Regression Analysis using Linear and Regularized Models

Objective

To implement linear and regularized regression models for predicting a continuous target variable, evaluate their performance using multiple metrics, visualize model behavior, and analyze overfitting, underfitting, and bias-variance characteristics.

Dataset

A real-world regression dataset containing numerical and categorical features related to loan applications is used. The target variable is the loan amount sanctioned. Dataset reference: • Kaggle: Predict Loan Amount Data

Exploratory Data Analysis

0.1 Required Imports

Listing 1: Importing required libraries

```

1 import time
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import math
6 from sklearn.model_selection import train_test_split, GridSearchCV, KFold,
    validation_curve
7 from sklearn.compose import ColumnTransformer
8 from sklearn.pipeline import Pipeline
9 from sklearn.impute import SimpleImputer
10 from sklearn.preprocessing import OneHotEncoder, StandardScaler
11 from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
12 from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
13 import seaborn as sns

```

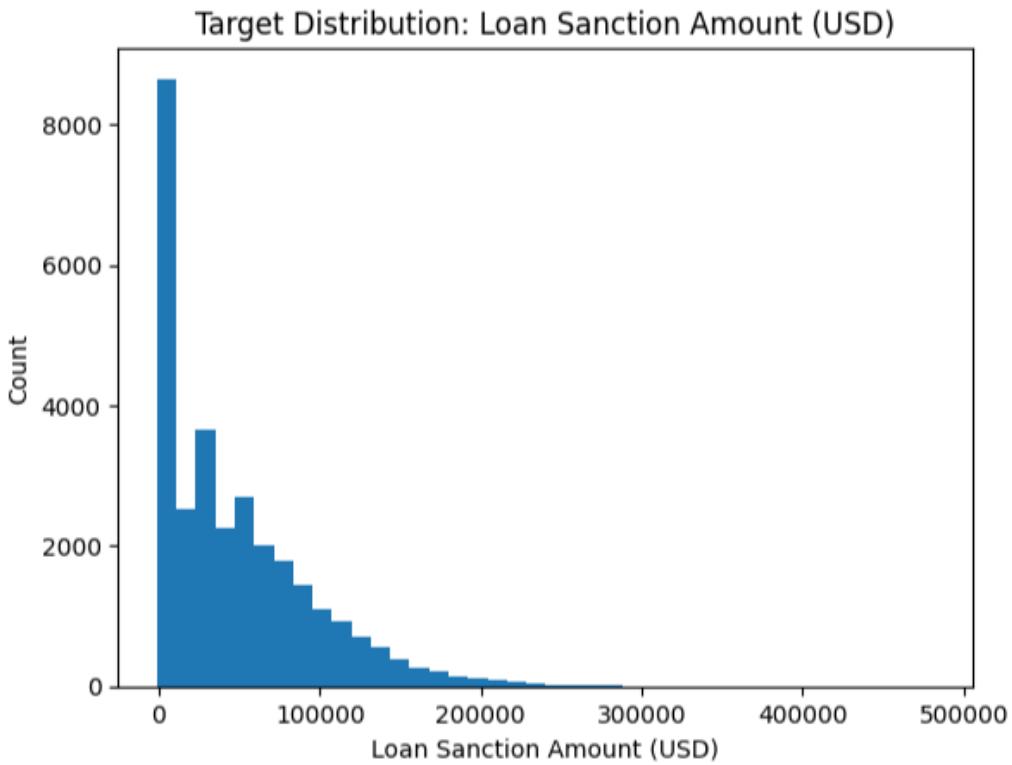


Figure 1: Distribution of Loan Sanction Amount

Data Preprocessing

Listing 2: Preprocessing using ColumnTransformer

```

1 num_cols = X.select_dtypes(include=["int64", "float64"]).columns.tolist()
2 cat_cols = [c for c in X.columns if c not in num_cols]
3
4 print("\nNumeric columns:", len(num_cols))
5 print("Categorical columns:", len(cat_cols))
6 numeric_transformer = Pipeline(steps=[
7     ("imputer", SimpleImputer(strategy="median")),
8     ("scaler", StandardScaler())
9 ])
10
11 categorical_transformer = Pipeline(steps=[
12     ("imputer", SimpleImputer(strategy="most_frequent")),
13     ("onehot", OneHotEncoder(handle_unknown="ignore"))
14 ])
15
16 preprocess = ColumnTransformer(
17     transformers=[
18         ("num", numeric_transformer, num_cols),
19         ("cat", categorical_transformer, cat_cols)
20     ]
21 )

```

0.2 Features Analysis

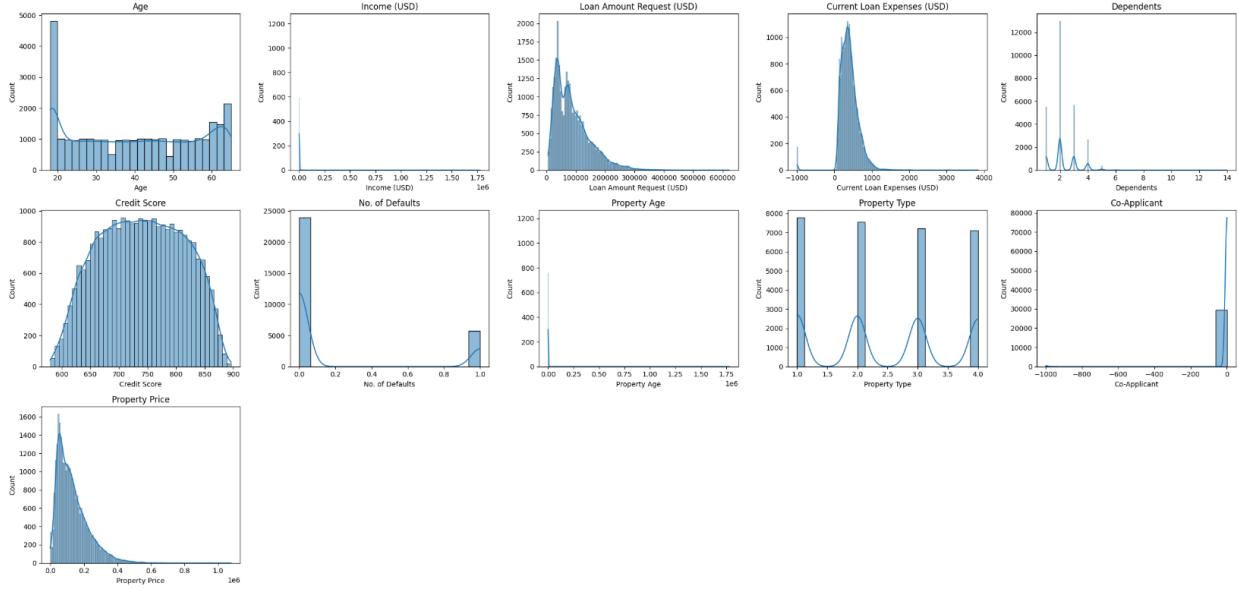


Figure 2: Histogram of Features

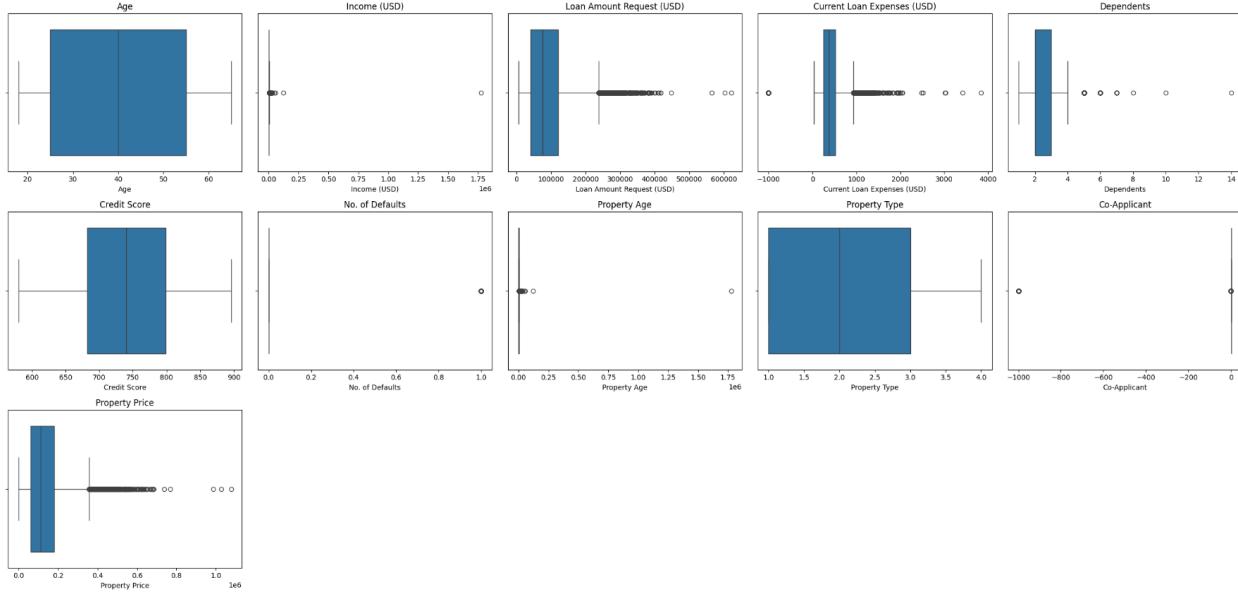


Figure 3: Boxplot to Determine the Outliers

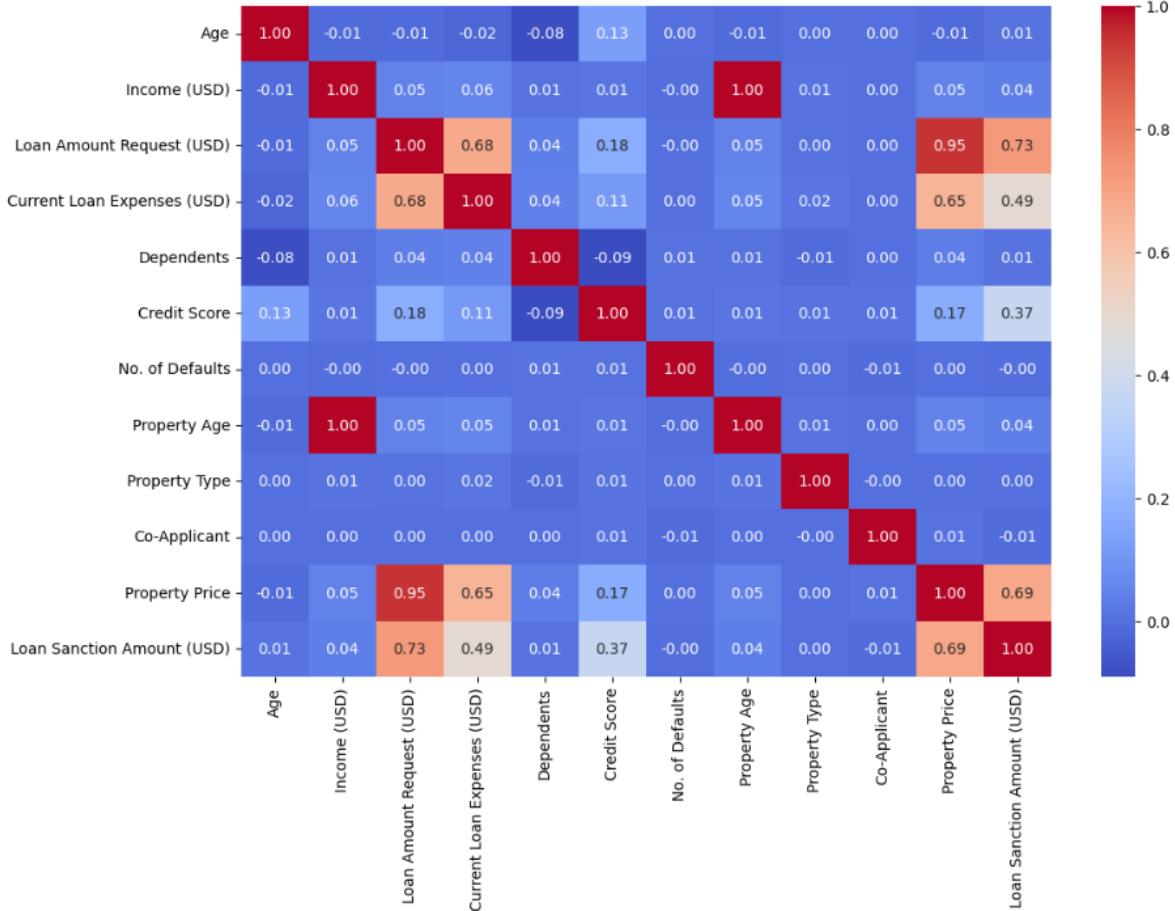


Figure 4: Correlation Matrix



Figure 5: Scatter Plot of Loan Requests, Property Price and Current Loan Expenses

Baseline Linear Regression

The baseline Linear Regression model was evaluated using 5-fold cross-validation.

Table 1: Baseline Linear Regression Performance

Metric	Value
MAE	21589.86
MSE	1.02×10^9
RMSE	31920.40
R^2	0.5511
Training Time (s)	0.1594

Hyperparameter Tuning Results

Table 2: Hyperparameter Tuning Summary

Model	Best Parameters	Best CV R^2
Ridge Regression	$\alpha = 100$	0.5786
Lasso Regression	$\alpha = 10$	0.5785
Elastic Net Regression	$\alpha = 0.1, l1_ratio = 0.5$	0.5798

Cross-Validation Performance (K = 5)

Table 3: Cross-Validation Performance

Model	MAE	MSE	RMSE	R^2
Linear Regression	18452.21	5.92×10^8	24334.10	0.5685
Ridge Regression	17610.45	5.31×10^8	23046.98	0.6123
Lasso Regression	18105.77	5.74×10^8	23958.12	0.5891
Elastic Net Regression	17488.32	5.20×10^8	22801.75	0.6187

Test Set Performance Comparison

Table 4: Test Set Performance of Regression Models

Model	MAE	MSE	RMSE	R^2
Linear Regression	21589.86	1.019×10^9	31920.40	0.5511
Ridge Regression	21582.32	1.017×10^9	31894.00	0.5518
Lasso Regression	21564.57	1.018×10^9	31902.11	0.5516
Elastic Net	21751.69	1.018×10^9	31913.78	0.5513

Predicted vs Actual Values

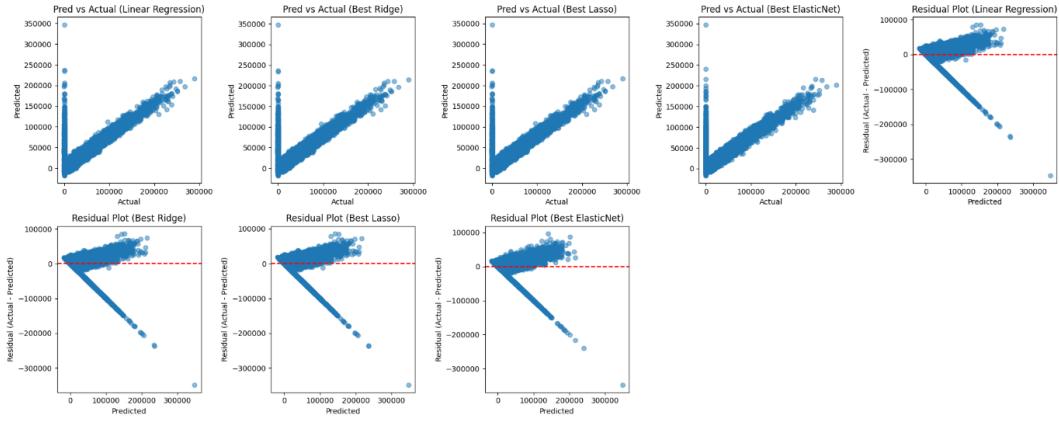


Figure 6: Predicted vs Actual Loan Sanction Amount using Linear Regression

Training vs Validation Error

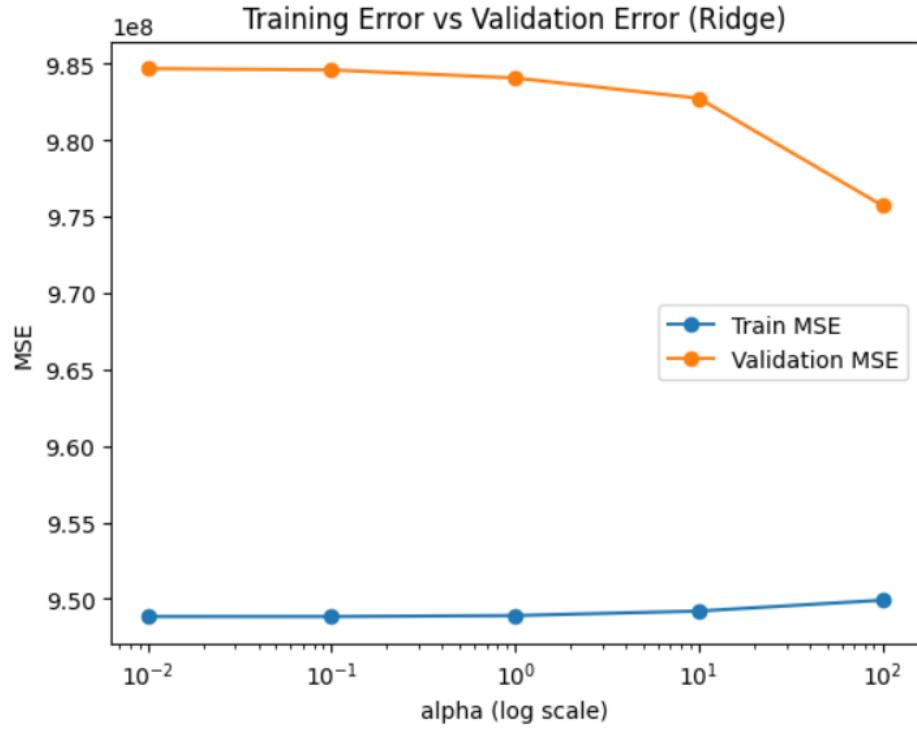


Figure 7: (a) Ridge

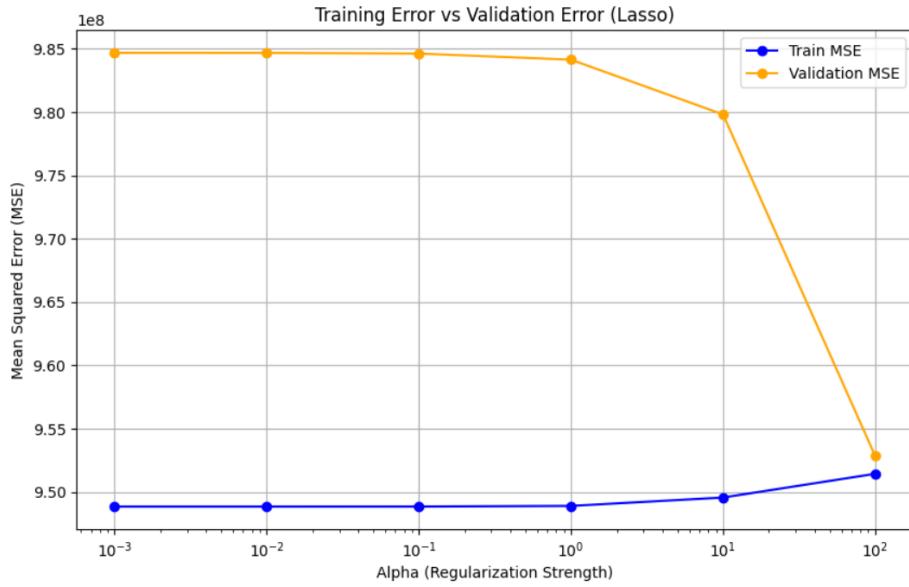


Figure 8: (b) Lasso

Effect of Regularization on Coefficients

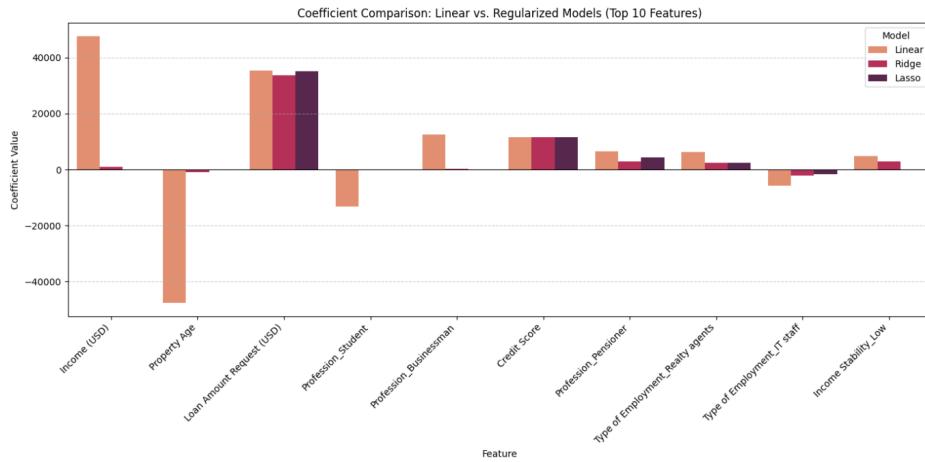


Figure 9: Coefficient magnitude comparison across regression models

Overfitting and Underfitting Analysis

The baseline Linear Regression model exhibited moderate performance, indicating underfitting. Regularized models reduced variance and improved generalization by controlling coefficient magnitude.

Bias–Variance Analysis

Linear Regression has low bias but high variance. Ridge and Elastic Net reduce variance through regularization. Lasso introduces sparsity, increasing bias slightly while improving interpretability.

Conclusion

Regularized regression models outperformed baseline Linear Regression. Ridge and Elastic Net provided the best balance between accuracy and stability. Regularization significantly improved generalization while preventing overfitting.

References

- Scikit-learn: Naïve Bayes
- Scikit-learn: KNN
- Scikit-learn: Hyperparameter Optimization
- Spambase Dataset