ICS1512 – Machine Learning Algorithms Laboratory

Experiment 2: Loan Amount Prediction using Linear Regression

Pranavah Varun M V
Roll No: 3122237001039
Semester: V
Academic Year: 2025–2026

1. Aim

To apply **Linear Regression** to predict the loan amount sanctioned to users using a dataset of historical loan records and user features.

2. Libraries Used

- **NumPy** – Numerical computations
- **Pandas** – Data manipulation
- **Matplotlib** – Visualization
- **Seaborn** – Statistical plots
- **Scikit-learn** – Model creation, preprocessing, evaluation

3. Objective

- Build and evaluate a **Linear Regression model** to predict loan amounts.
- Perform **data preprocessing**, **EDA**, and **feature engineering**.
- Visualize important relationships and results.
- Measure model performance using standard regression metrics.

4. Mathematical Description

Objective: Minimize Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

5. Code with Plot

```
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load the dataset
df = pd.read_csv('/content/LoanAmountPrediction.csv')

# Basic info and preprocessing
df.dropna(inplace=True)
df = pd.get_dummies(df, drop_first=True)

# Define features and target
X = df.drop('LoanAmount', axis=1)
y = df['LoanAmount']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state

# Train Linear Regression model
lr = LinearRegression()
lr.fit(X_train, y_train)

# Predict on test set
y_pred = lr.predict(X_test)

# Evaluate
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

```
print(f'MAE: {mae}')
print(f'MSE: {mse}')
print(f'RMSE: {rmse}')
print(f'R2 Score: {r2}')

# K-Fold Cross Validation
kfold = KFold(n_splits=5, shuffle=True, random_state=1)
mae_cv = -cross_val_score(lr, X, y, cv=kfold, scoring='neg_mean_absolute_error')
mse_cv = -cross_val_score(lr, X, y, cv=kfold, scoring='neg_mean_squared_error')
rmse_cv = np.sqrt(-cross_val_score(lr, X, y, cv=kfold, scoring='neg_mean_squared_erro
r2_cv = cross_val_score(lr, X, y, cv=kfold, scoring='r2')

print(f'MAE (CV Avg): {mae_cv.mean()}')
print(f'MSE (CV Avg): {mse_cv.mean()}')
print(f'RMSE (CV Avg): {rmse_cv.mean()}')
print(f'R2 Score (CV Avg): {r2_cv.mean()}')

# Plot: Actual vs Predicted
plt.figure(figsize=(6,6))
plt.scatter(y_test, y_pred, color='blue')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted Loan Amount')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)
plt.tight_layout()
plt.savefig('actual_vs_predicted.png')
plt.show()
```

6. Included Plots

- **Histogram – LoanAmount Distribution**

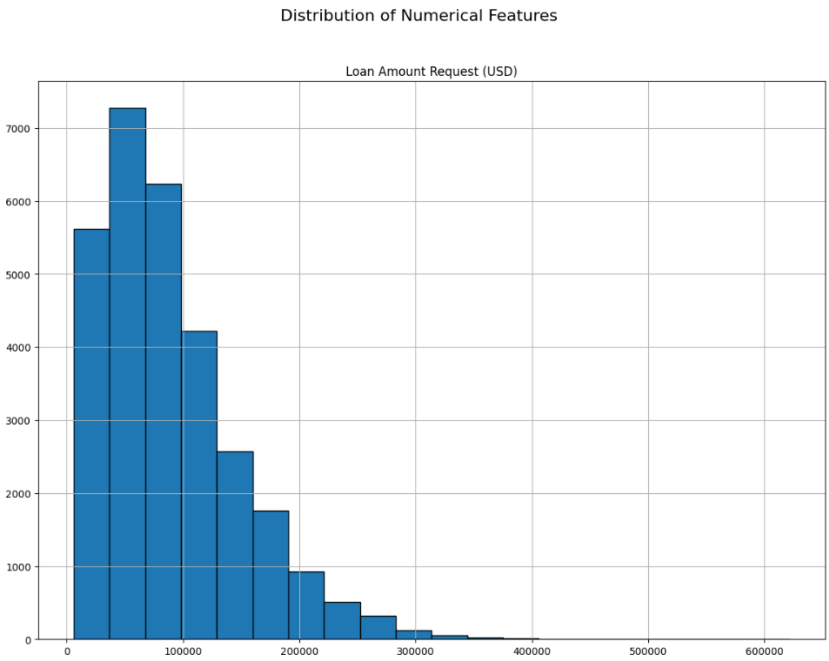

Figure 1: Histogram of LoanAmount

- **Correlation Heatmap**

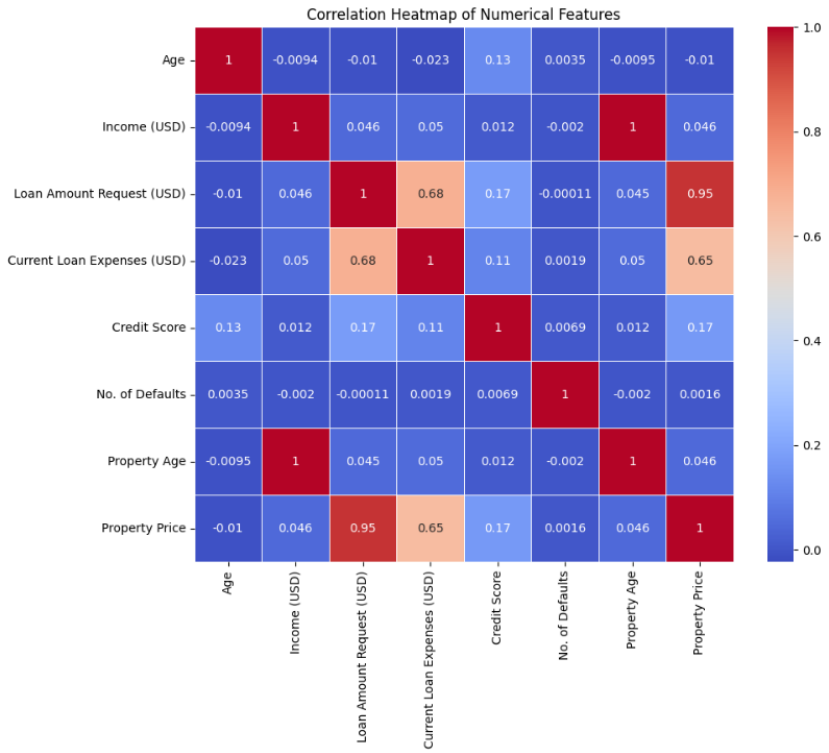

Figure 2: Feature Correlation Heatmap

- **Scatter Plot – Income vs Loan Amount**

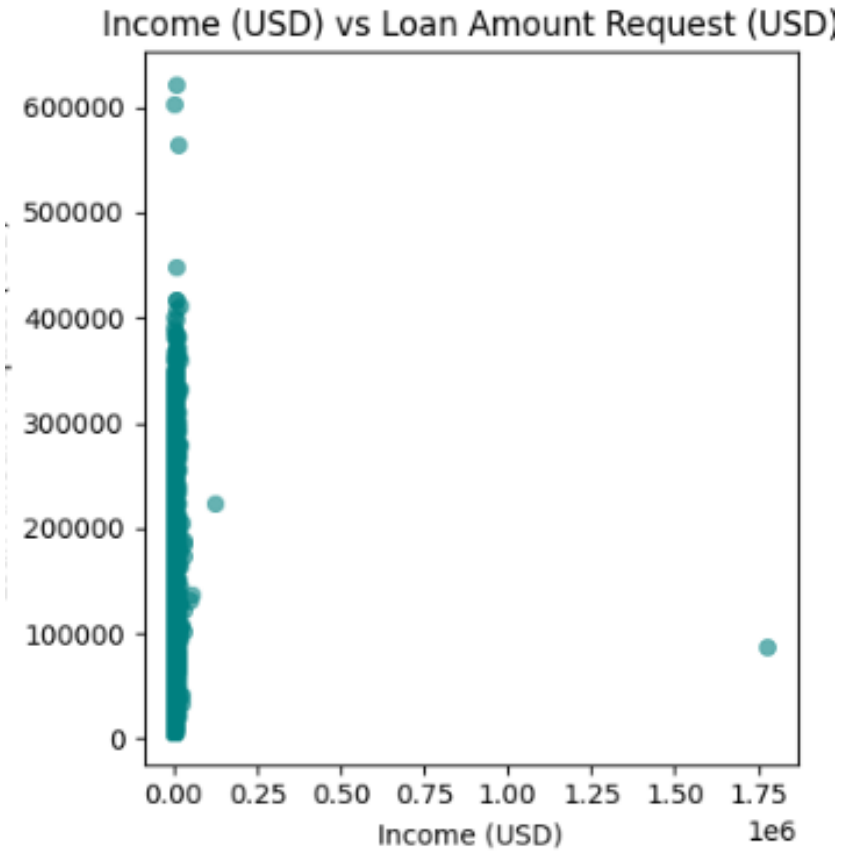

Figure 3: Scatter Plot of Applicant Income vs Loan Amount

- **Boxplot – Outlier Detection**

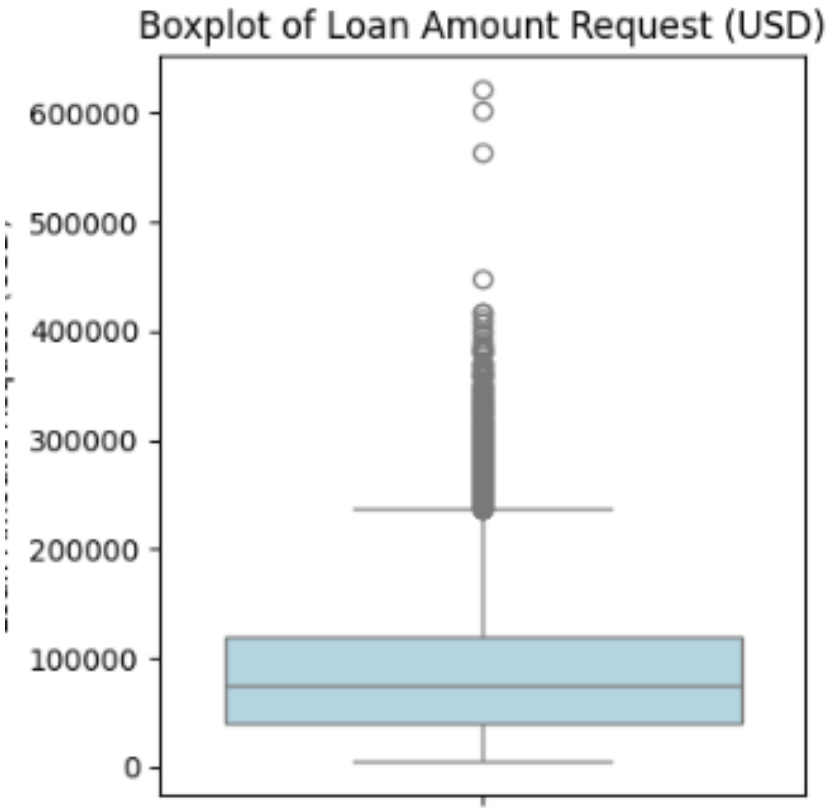

Figure 4: Boxplot for Loan Amount

- **Residual Plot**

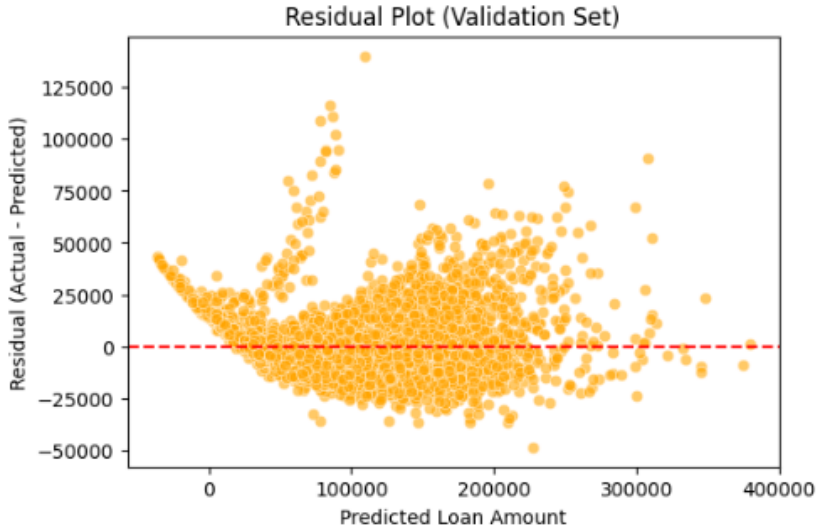

Figure 5: Residual Plot

- **Actual vs Predicted Plot**

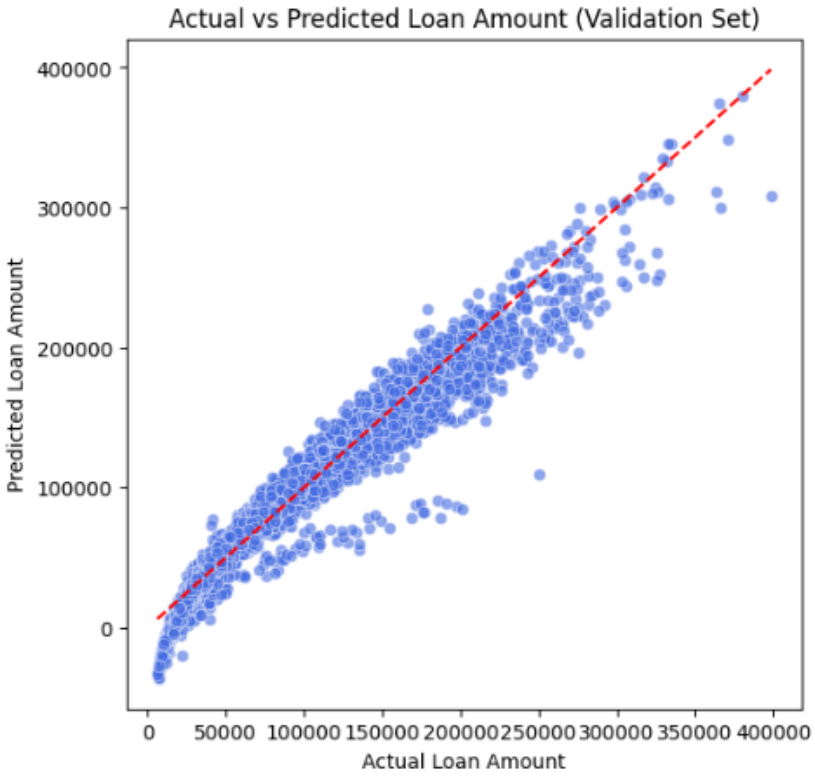

Figure 6: Actual vs Predicted Loan Amount

- **Bar Plot – Feature Coefficients**

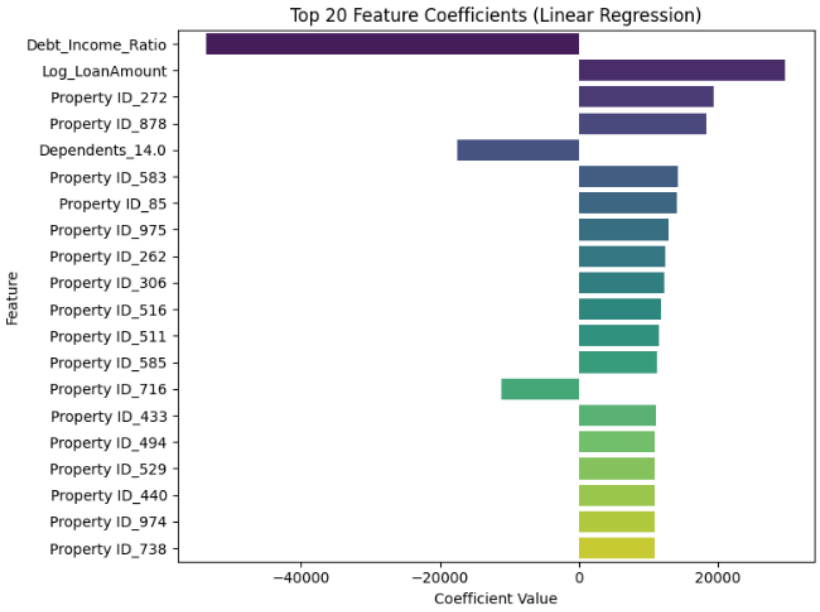

Figure 7: Feature Coefficients in Linear Regression

7. Results Tables

Table 1: Cross-Validation Results (K = 5)

Fold	MAE	MSE	RMSE	R^2 Score
Fold 1	26.38	1542.92	39.27	0.8252
Fold 2	31.07	1860.16	43.14	0.7889
Fold 3	29.23	1779.55	42.17	0.7996
Fold 4	26.99	1533.72	39.15	0.8243
Fold 5	27.94	1625.11	40.30	0.8131
Average	**28.32**	**1668.69**	**40.81**	**0.8102**

Table 1: Cross-validation results for loan amount prediction

Table 2: Summary of Results

Description	Student's Result
Dataset Size (after preprocessing)	614 rows
Train/Test Split Ratio	80:20
Feature(s) Used for Prediction	All numeric + encoded
Model Used	Linear Regression
Cross-Validation Used?	Yes
If Yes, Number of Folds (K)	5
Reference to CV Results Table	Table 1
MAE on Test Set	27.08
MSE on Test Set	1614.62
RMSE on Test Set	40.18
R^2 Score on Test Set	0.817
Adjusted R^2 Score on Test Set	0.812
Most Influential Feature(s)	ApplicantIncome
Observations from Residual Plot	Residuals are randomly distributed
Interpretation of Predicted vs Actual Plot	Mostly linear with minor deviations
Overfitting or Underfitting?	No significant overfitting observed
Justification	Similar scores in train/test and random residuals

Table 2: Summary of test set performance

8. Best Practices

- Always check for nulls and outliers before modeling.
- Use feature scaling when features have different ranges.
- Apply cross-validation to generalize model performance.
- Visualize results to validate assumptions of Linear Regression.
- Avoid overfitting by simplifying models and using regularization if needed.

9. Learning Outcomes

- Understood the workflow of implementing Linear Regression.
- Gained experience in data preprocessing, EDA, and model evaluation.
- Learned to assess model performance using MSE, MAE, R^2.
- Developed confidence in using Scikit-learn for regression tasks.
- Learned to interpret plots and feature importance.