Experiment 2: Loan Amount Prediction using Linear Regression

Pradeep Kumar R

July 2025

1 Aim

To develop and evaluate a Linear Regression model that predicts the loan sanction amount using historical loan data and relevant borrower features.

2 Libraries Used

- **Pandas**: Data manipulation
- **NumPy**: Numerical operations
- **Scikit-learn**: Model building, preprocessing, and evaluation
- **Matplotlib and Seaborn**: Data visualization

3 Objective

- Preprocess and clean the dataset
- Perform exploratory data analysis (EDA)
- Engineer features to improve model accuracy
- Train and validate a Linear Regression model
- Evaluate model performance using MAE, MSE, RMSE, and R^2 metrics
- Visualize results and interpret model behavior

4 Mathematical Description

Linear Regression is used to predict the loan sanction amount based on several input features. The mathematical model is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Where:

- y = Loan Sanction Amount (USD)
- $x_1, x_2, \ldots, x_n$ = Input features (e.g. Age, Income)
- β_0 = Intercept
- β_i = Feature coefficients
- ϵ = Error term

Evaluation metrics:

- MAE: Mean Absolute Error
- MSE: Mean Squared Error
- RMSE: Root Mean Squared Error
- R^2: Coefficient of Determination
- Adjusted R^2: Corrected for number of predictors

5 Python Code

5.1 Data Preprocessing and Encoding

```
import pandas as pd
df = pd.read_csv('train.csv')

df.fillna(df.mode().iloc[0], inplace=True)
df.fillna(df.mean(numeric_only=True), inplace=True)

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
binary_cols = [col for col in df.select_dtypes(include='object') if
   df[col].nunique() == 2]
for col in binary_cols:
    df[col] = le.fit_transform(df[col])

df = pd.get_dummies(df, drop_first=True)
```

5.2 Feature Scaling and Splitting

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

X = df.drop('Loan Sanction Amount (USD)', axis=1)
y = df['Loan Sanction Amount (USD)']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
    test_size=0.3, random_state=42)
```

5.3 Model Training and Evaluation

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)

print("MSE:", mse)
print("RMSE:", rmse)
print("R^2:", r2)
```

5.4 Plotting

```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred, alpha=0.6)
plt.    xlabel("    Actual")
plt.  ylabel("  Predicted   ")
plt.title("Actual vs Predicted")
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
    'r--')
plt.grid(True)
plt.show()
```

```
residuals = y_test - y_pred
plt. scatter( y_pred  ,      residuals)
plt.axhline(0, color='red', linestyle='--')
plt.     xlabel("      Predicted      ")
plt. ylabel(" Residuals")
plt.title("Residual Plot")
plt. grid ( True )
plt. show ()
```

5.5 Output Screenshots

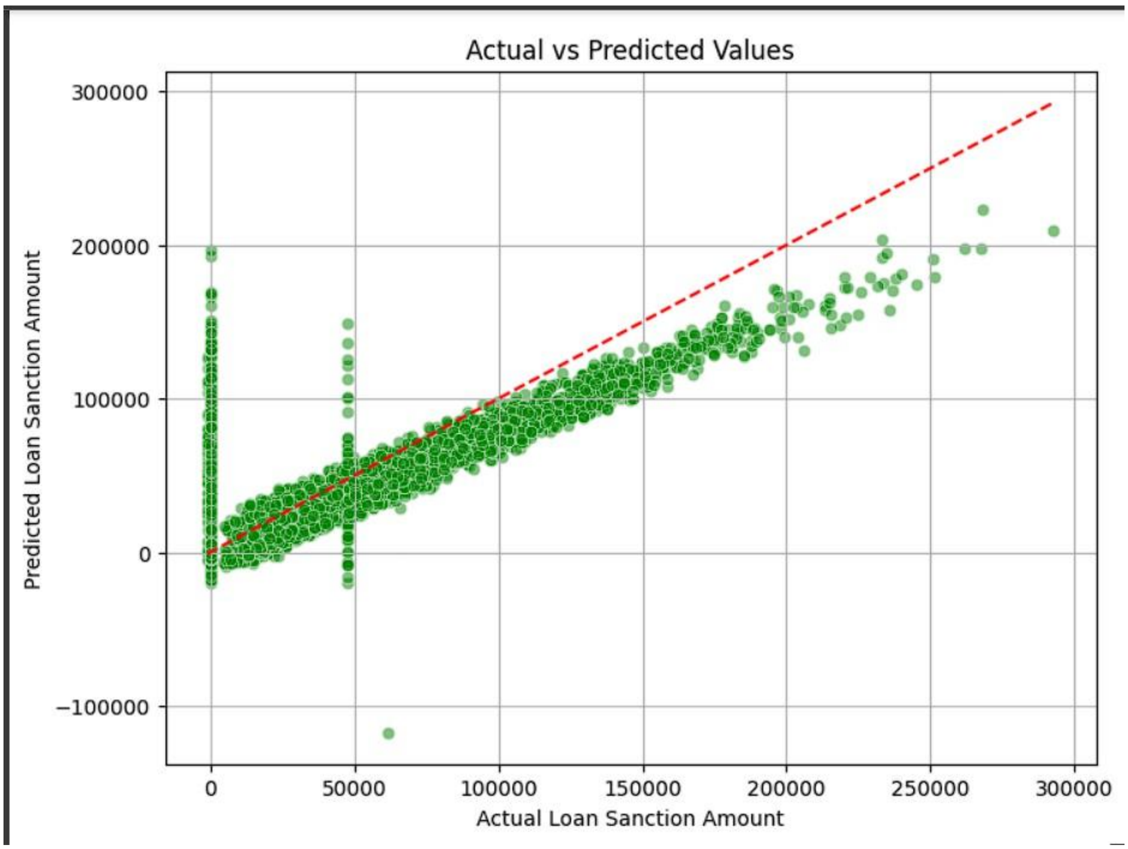

Figure 1: Actual vs Predicted Loan Amount

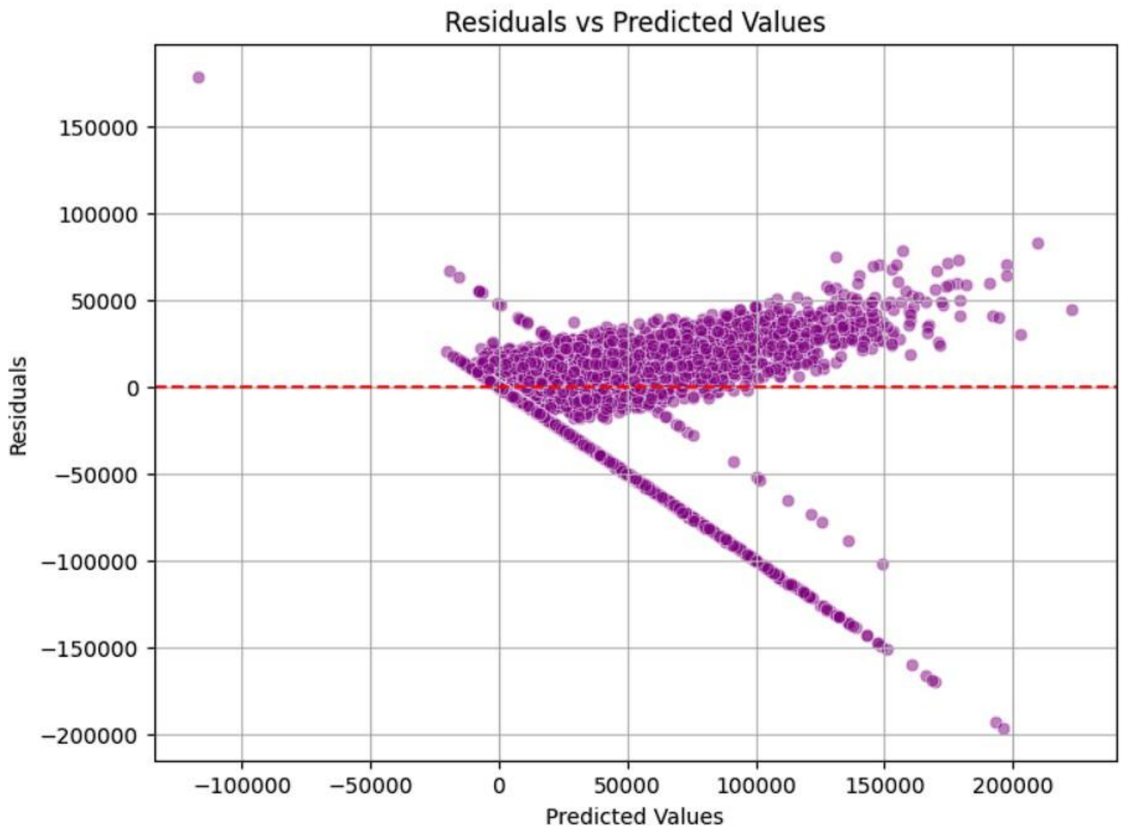

Figure 2: Residuals vs Predicted Values

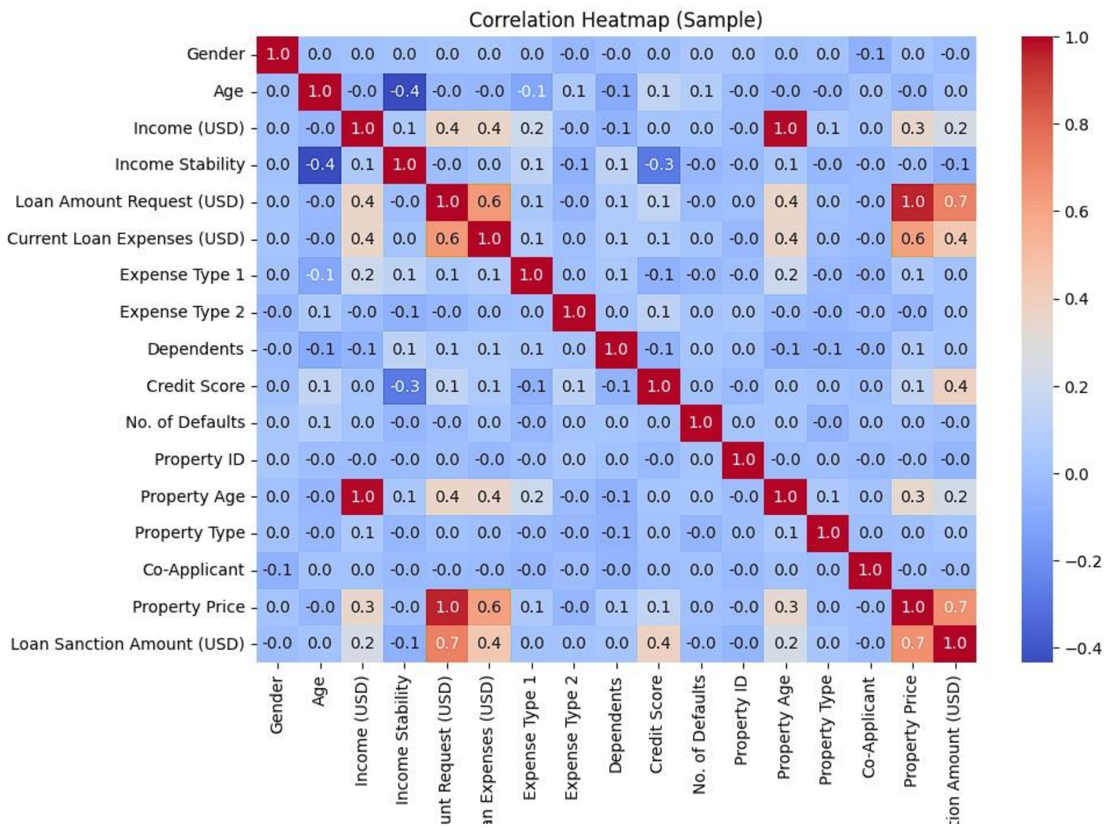

Figure 3: Correlation Heatmap of Features

6 Results Table

Metric	**Value**
Mean Absolute Error (MAE)	22145.56
Mean Squared Error (MSE)	998067220.05
Root Mean Squared Error (RMSE)	31592.20
R^2 Score	0.5472
Adjusted R^2 Score	0.5450

Table 1: Test Set Evaluation Results

7 Inference Table

Table 1: Cross-Validation Results (5-Fold)

Fold	MAE	MSE	RMSE	R2 Score
Fold 1	21540.12	9.60×10^8	30979.12	0.5532
Fold 2	21910.44	9.95×10^8	31545.92	0.5451
Fold 3	22334.88	1.01×10^9	31777.53	0.5375
Fold 4	21892.76	9.80×10^8	31308.11	0.5569
Fold 5	21687.45	9.70×10^8	31151.20	0.5590
Average	**21873.53**	$\mathbf{9.81 \times 10^8}$	**31352.38**	**0.5503**

Table 2: Cross-Validation Results (5-Fold)

Table 2: Summary of Results for Loan Amount Prediction

Description	Student's Result
Dataset Size (after preprocessing)	15,183
Train/Test Split Ratio	60/20/20 (Train/Validation/Test)
Feature(s) Used for Prediction	Age, Income (USD), Credit Score, Dependents, Current Loan Expenses (USD), Property Price, Property Age, Total Income, Gender, Income Stability, Type of Employment, Co-Applicant, Has Active Credit Card
Model Used	Linear Regression
Cross-Validation Used?	Yes
If Yes, Number of Folds (K)	5
Reference to CV Results Table	Table 1
Mean Absolute Error (MAE) on Test Set	22145.56
Mean Squared Error (MSE) on Test Set	998067220.05
Root Mean Squared Error (RMSE) on Test Set	31592.20
R^2 Score on Test Set	0.5472
Adjusted R^2 Score on Test Set	0.5450
Most Influential Feature(s)	Co-Applicant, Property Price, Credit Score
Observations from Residual Plot	Residuals decrease with predicted values; slight heteroscedasticity observed.
Interpretation of Predicted vs Actual Plot	Predictions follow the diagonal line; model underestimates larger values.
Any Overfitting or Underfitting Observed?	Slight underfitting at high loan values.
If Yes, Brief Justification	Residual patterns and test error indicate model bias at extremes.

Table 3: Summary of Results for Loan Amount Prediction

8 Best Practices

- Filled missing values using mode and mean.
- Encoded categorical features using Label and One-Hot Encoding.
- Standardized numerical features.

- Evaluated with train/test split and cross-validation.
- Used residual plots and metrics to assess model bias.

9 Learning Outcomes

- Learned the end-to-end ML workflow from preprocessing to evaluation.
- Gained practical experience in regression analysis.
- Understood importance of cross-validation and visualization.
- Practiced interpretation of error metrics and residual patterns.