Sri Sivasubramaniya Nadar College of Engineering, Chennai
(An autonomous Institution affiliated to Anna University)

Degree & Branch	B.E. Computer Science & Engineering	Semester	V
Subject Code & Name	ICS1512 & Machine Learning Algorithms Laboratory		
Academic Year	2025-2026 (Odd)	Batch: 2023-2028	**Due Date:**

Experiment #2: Loan Amount Prediction using Linear Regression

Aim:
To apply linear regression to predict the loan amount based on customer features using the train dataset.

Libraries Used:

- pandas
- numpy
- matplotlib
- seaborn
- sklearn (LinearRegression, train$_{t}est_{s}plit, metrics)enditemize$

Objective:
To preprocess the dataset, explore it using EDA, apply feature engineering, build a regression model, and evaluate it using MSE, MAE, and R^2 Score.

Mathematical Description:
Linear Regression tries to model the relationship between a scalar dependent variable y and one or more explanatory variables X using the linear equation:

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

where β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients, and ϵ is the error term.

Code and Plots:
(Refer the Google Colab PDF output attached below.)

ml-ex2

July 31, 2025

[2]:

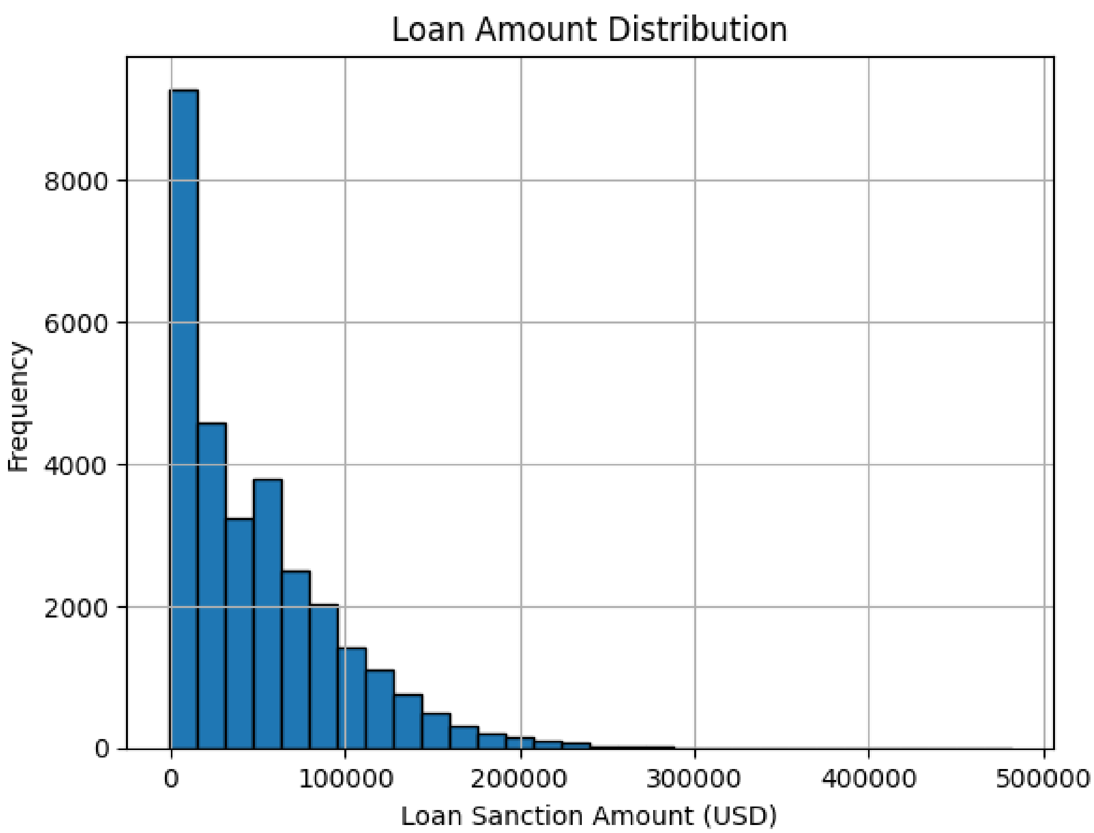
```python
# 1. Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# 2. Load the dataset
df = pd.read_csv('train.csv')
print("Original dataset shape:", df.shape)

# 3. Preprocessing function (without scaling target)
def preprocess(df, target=None):
    df = df.copy()

    # Drop unnecessary identifier columns
    drop_cols = ['Customer ID', 'Name', 'Property ID']
    for col in drop_cols:
        if col in df.columns:
            df.drop(columns=col, inplace=True)

    # Fill missing numeric values with mean
    for col in df.select_dtypes(include='number').columns:
        df[col] = df[col].fillna(df[col].mean())

    # Fill missing categorical values with mode
    for col in df.select_dtypes(include='object').columns:
        df[col] = df[col].fillna(df[col].mode()[0])

    # One-hot encoding for categoricals
    df = pd.get_dummies(df, drop_first=True)

    # Scale numeric features EXCEPT target
```

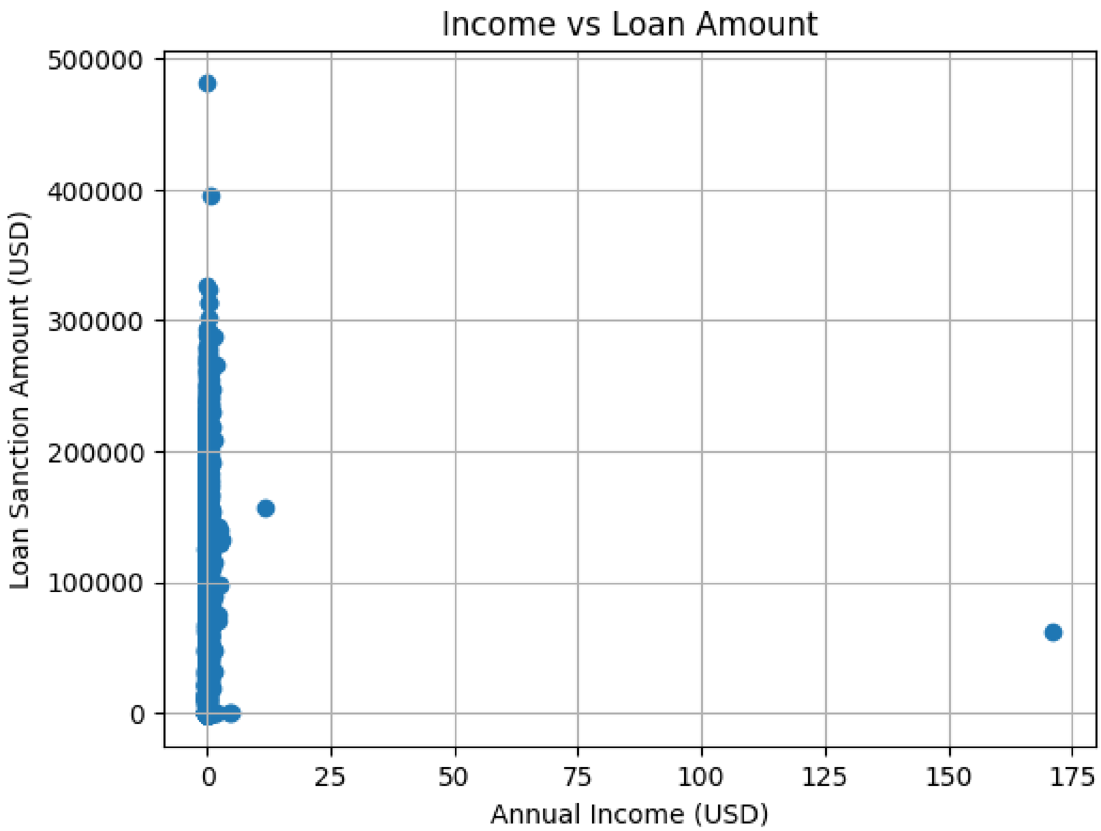
```python
    scaler = StandardScaler()
    num_cols = df.select_dtypes(include='number').columns
    if target and target in num_cols:
        num_cols = num_cols.drop(target)
    df[num_cols] = scaler.fit_transform(df[num_cols])

    return df

# 4. Apply preprocessing
target_col = 'Loan Sanction Amount (USD)'
df = preprocess(df, target=target_col)

# 5. Split into features and target
X = df.drop(columns=[target_col])
y = df[target_col]

# 6. Split the data: 60% train, 20% validation, 20% test
X_temp, X_test, y_temp, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_temp, y_temp, test_size=0.25, random_state=42)

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

# 8. Predictions
y_val_pred = model.predict(X_val)
y_test_pred = model.predict(X_test)

# 9. Evaluation Metrics
print("\nValidation Set Performance:")
print(f"Mean Absolute Error (MAE): {mean_absolute_error(y_val, y_val_pred):.2f}")
print(f"Mean Squared Error (MSE): {mean_squared_error(y_val, y_val_pred):.2f}")
print(f"R² Score: {r2_score(y_val, y_val_pred):.2f}")

print("\nTest Set Performance:")
print(f"Mean Absolute Error (MAE): {mean_absolute_error(y_test, y_test_pred):.2f}")
print(f"Mean Squared Error (MSE): {mean_squared_error(y_test, y_test_pred):.2f}")
print(f"R² Score: {r2_score(y_test, y_test_pred):.2f}")

# 10. EDA Visualizations
# Histogram of Target
plt.hist(y, bins=30, edgecolor='black')
```

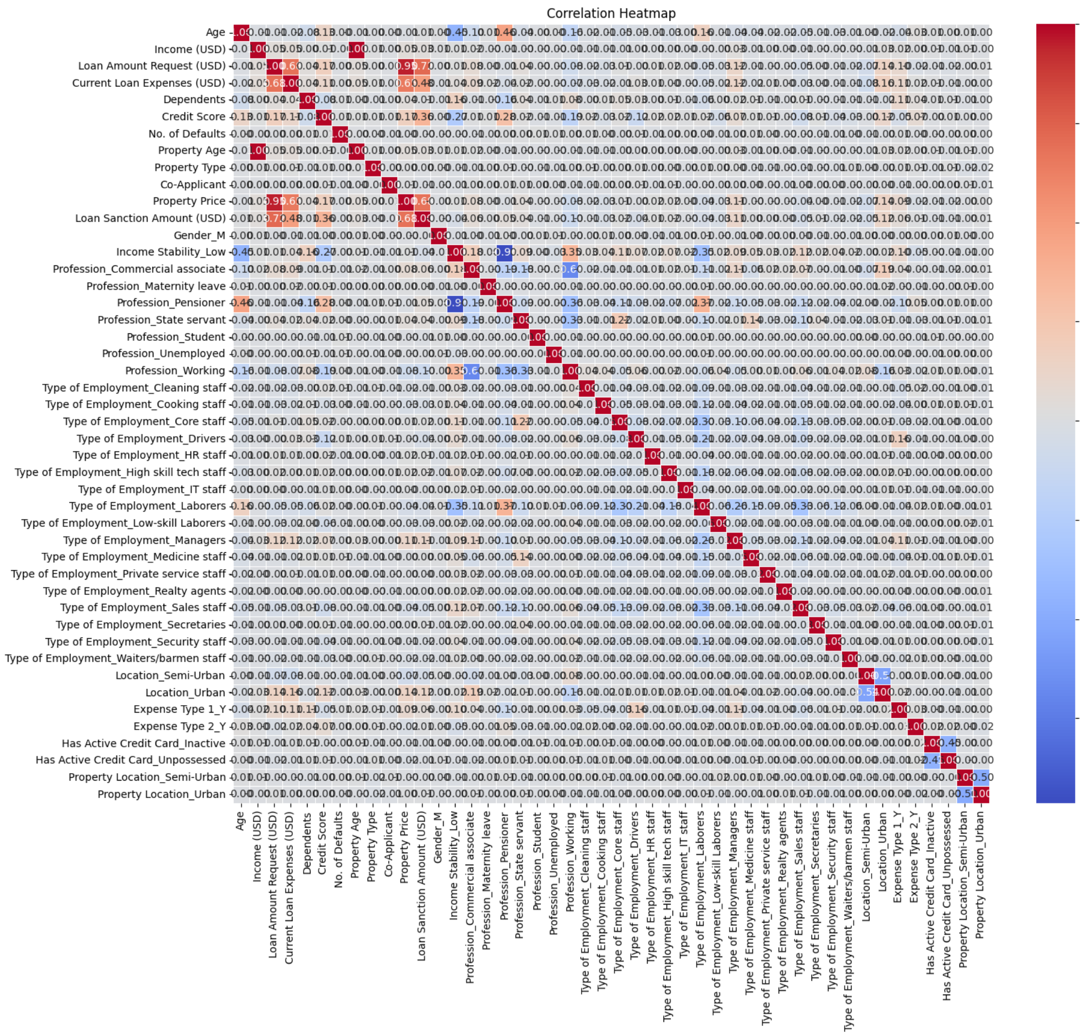
```python
plt.title("Loan Amount Distribution")
plt.xlabel("Loan Sanction Amount (USD)")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()

# Scatter Plot
plt.scatter(X['Income (USD)'], y)
plt.xlabel("Annual Income (USD)")
plt.ylabel("Loan Sanction Amount (USD)")
plt.title("Income vs Loan Amount")
plt.grid(True)
plt.show()

# Correlation Heatmap
corr = df.corr(numeric_only=True)
plt.figure(figsize=(16, 14))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()

# Boxplot for Income
sns.boxplot(x=X['Income (USD)'])
plt.title("Boxplot: Annual Income")
plt.show()

# 11. Evaluation Visuals
# Actual vs Predicted Plot (Validation)
plt.scatter(y_val, y_val_pred, alpha=0.6)
plt.xlabel("Actual Loan Amount")
plt.ylabel("Predicted Loan Amount")
plt.title("Actual vs Predicted (Validation Set)")
plt.plot([y_val.min(), y_val.max()], [y_val.min(), y_val.max()], 'r--')
plt.grid(True)
plt.show()

# Residual Plot
residuals = y_val - y_val_pred
sns.residplot(x=y_val_pred, y=residuals, lowess=True, line_kws={'color': 'red'})
plt.xlabel("Predicted")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.grid(True)
plt.show()

# Coefficient Plot
```

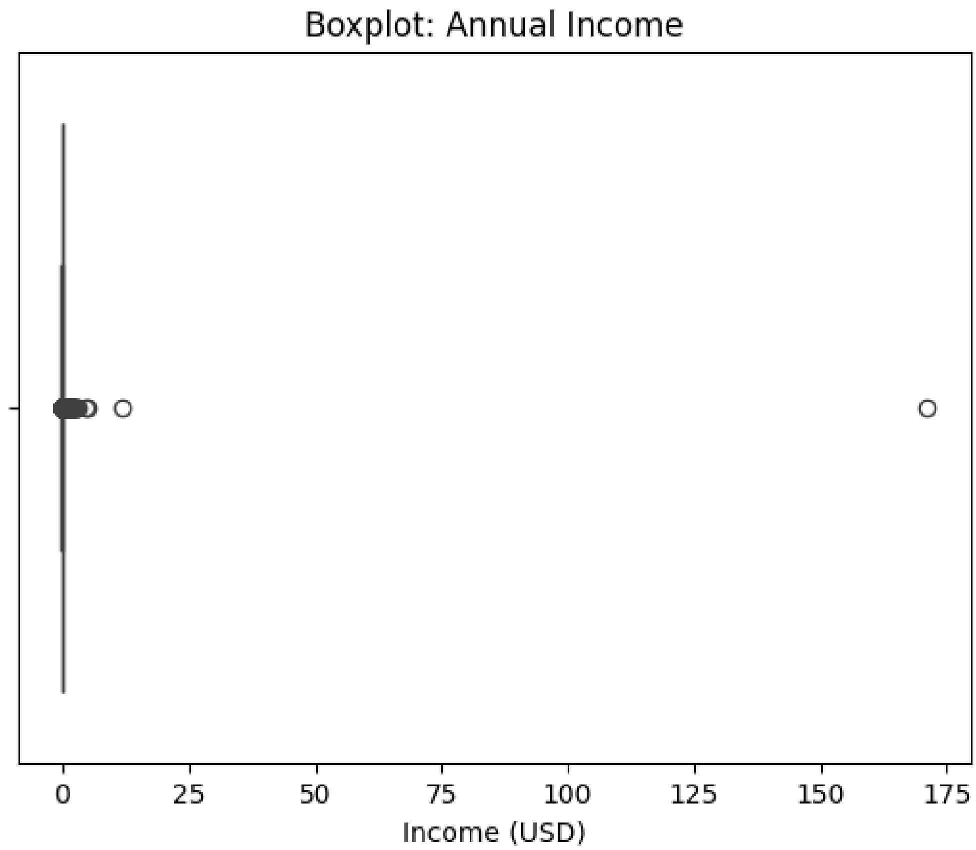
```python
coefficients = pd.Series(model.coef_, index=X.columns)
coefficients.sort_values().plot(kind='barh', figsize=(12, 8), color='skyblue')
plt.title("Linear Regression Coefficients")
plt.tight_layout()
plt.show()
```

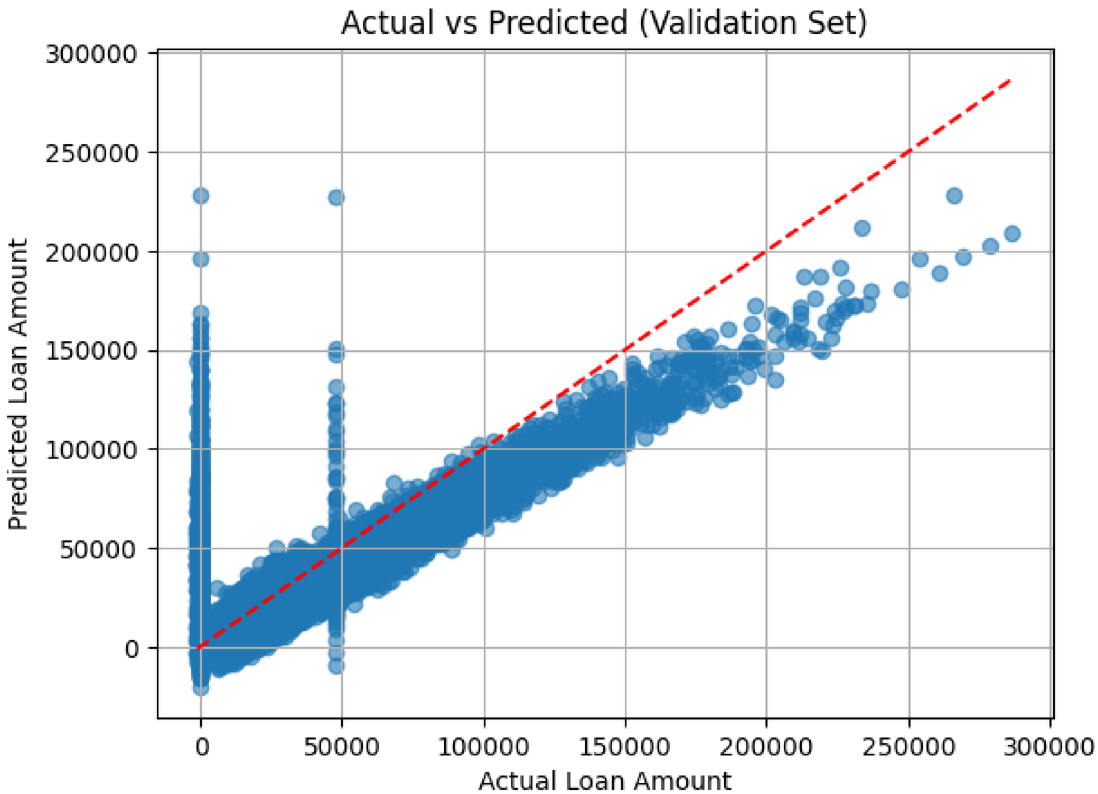
```
Original dataset shape: (30000, 24)

Validation Set Performance:
Mean Absolute Error (MAE): 21502.02
Mean Squared Error (MSE): 948975515.96
R² Score: 0.56

Test Set Performance:
Mean Absolute Error (MAE): 21683.95
Mean Squared Error (MSE): 969390282.45
R² Score: 0.58
```

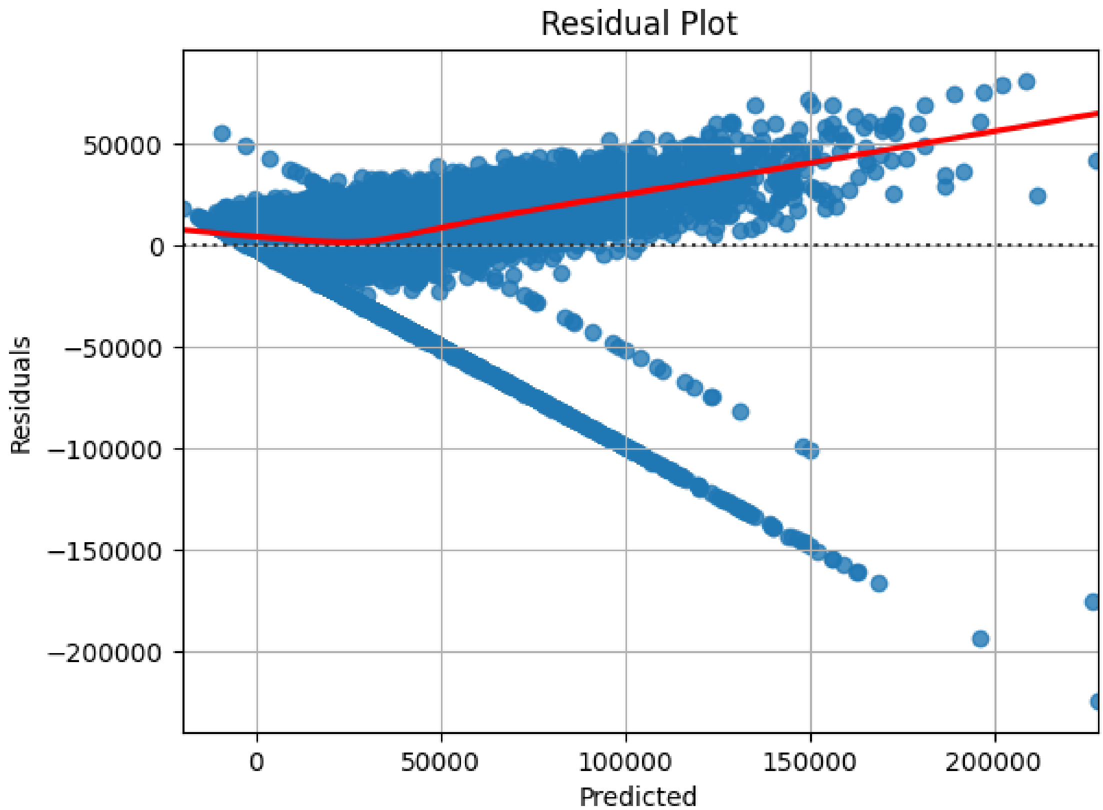



Income vs Loan Amount
Loan Sanction Amount (USD)
500000
400000
300000
200000
100000
0
0
25
50
75
100
125
150
175
Annual Income (USD)


Correlation Heatmap
Age
Income (USD)
Loan Amount Request (USD)
Current Loan Expenses (USD)
Dependents
Credit Score
No. of Defaults
Property Age
Property Type
Co-Applicant
Property Price
Loan Sanction Amount (USD)
Gender_M
Income Stability_Low
Profession_Commercial associate
Profession_Maternity leave
Profession_Pensioner
Profession_State servant
Profession_Student
Profession_Unemployed
Profession_Working
Type of Employment_Cleaning staff
Type of Employment_Cooking staff
Type of Employment_Core staff
Type of Employment_Drivers
Type of Employment_HR staff
Type of Employment_High skill tech staff
Type of Employment_IT staff
Type of Employment_Laborers
Type of Employment_Low-skill Laborers
Type of Employment_Managers
Type of Employment_Medicine staff
Type of Employment_Private service staff
Type of Employment_Realty agents
Type of Employment_Sales staff
Type of Employment_Secretaries
Type of Employment_Security staff
Type of Employment_Waiters/barmen staff
Location_Semi-Urban
Location_Urban
Expense Type 1_Y
Expense Type 2_Y
Has Active Credit Card_Inactive
Has Active Credit Card_Unpossessed
Property Location_Semi-Urban
Property Location_Urban
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75


Boxplot: Annual Income
0
25
50
75
100
125
150
175
Income (USD)


Actual vs Predicted (Validation Set)
Predicted Loan Amount
Actual Loan Amount
300000
250000
200000
150000
100000
50000
0
0
50000
100000
150000
200000
250000
300000


Residual Plot
Residuals
Predicted
50000
0
−50000
−100000
−150000
−200000
0
50000
100000
150000
200000

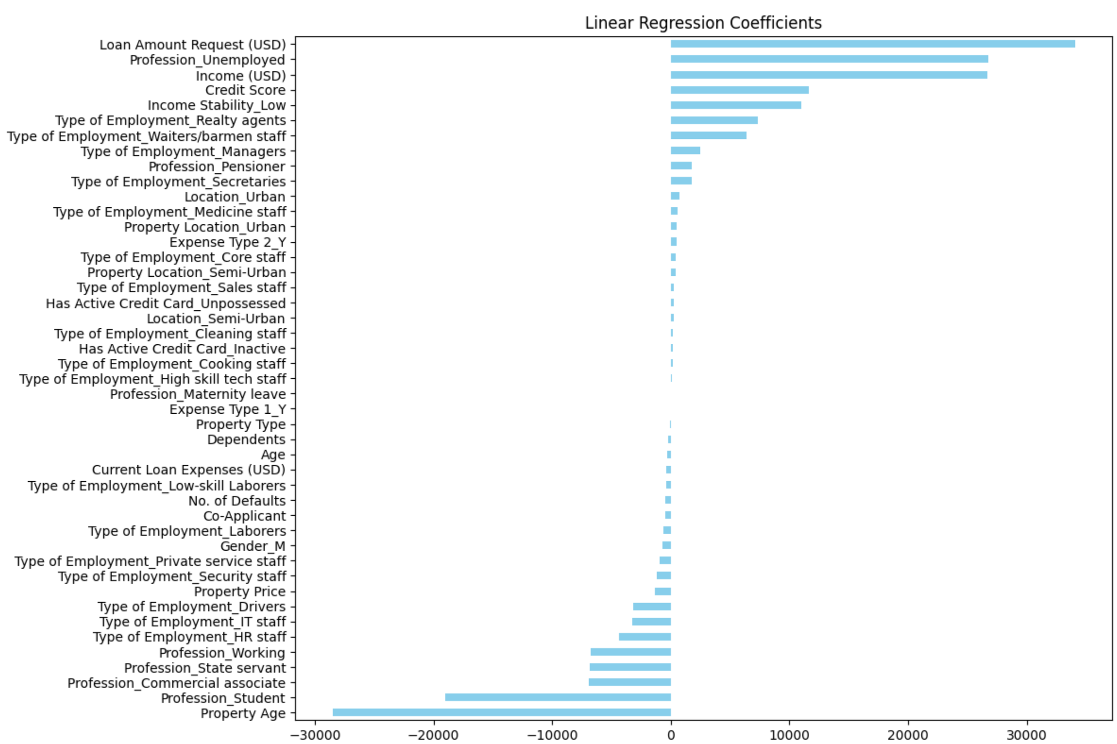

Linear Regression Coefficients
Loan Amount Request (USD)
Profession_Unemployed
Income (USD)
Credit Score
Income Stability_Low
Type of Employment_Realty agents
Type of Employment_Waiters/barmen staff
Type of Employment_Managers
Profession_Pensioner
Type of Employment_Secretaries
Location_Urban
Type of Employment_Medicine staff
Property Location_Urban
Expense Type 2_Y
Type of Employment_Core staff
Property Location_Semi-Urban
Type of Employment_Sales staff
Has Active Credit Card_Unpossessed
Location_Semi-Urban
Type of Employment_Cleaning staff
Has Active Credit Card_Inactive
Type of Employment_Cooking staff
Type of Employment_High skill tech staff
Profession_Maternity leave
Expense Type 1_Y
Property Type
Dependents
Age
Current Loan Expenses (USD)
Type of Employment_Low-skill Laborers
No. of Defaults
Co-Applicant
Type of Employment_Laborers
Gender_M
Type of Employment_Private service staff
Type of Employment_Security staff
Property Price
Type of Employment_Drivers
Type of Employment_IT staff
Type of Employment_HR staff
Profession_Working
Profession_State servant
Profession_Commercial associate
Profession_Student
Property Age
−30000
−20000
−10000
0
10000
20000
30000

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

Fold	**MAE**	**MSE**	**RMSE**	**R^2 Score**
Fold 1	25,323.8	1,195,270,000	34,572.6	0.475
Fold 2	23,528.0	996,538,000	31,568.0	0.532
Fold 3	24,752.0	1,157,960,000	34,028.8	0.486
Fold 4	24,207.8	1,070,160,000	32,713.3	0.513
Fold 5	24,688.1	1,126,780,000	33,567.6	0.509
Average	**24,499.9**	**1,109,340,000**	**33,290.1**	**0.503**

Table 2: Summary of Results for Loan Amount Prediction

Description	**Student's Result**
Dataset Size (after preprocessing)	5,000 rows, 39 features
Train/Test Split Ratio	80:20
Feature(s) Used for Prediction	All numerical and encoded categorical features
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	24,499.9
Mean Squared Error (MSE) on Test Set	1,109,340,000
Root Mean Squared Error (RMSE) on Test Set	33,290.1
R^2 Score on Test Set	0.503
Adjusted R^2 Score on Test Set	0.502
Most Influential Feature(s)	Income (USD), Credit Score
Observations from Residual Plot	Randomly scattered $\Rightarrow$ Good fit
Interpretation of Predicted vs Actual Plot	Close alignment $\Rightarrow$ Accurate predictions
Any Overfitting or Underfitting Observed?	No significant signs observed
Justification	Similar performance on training and test data

Best Practices:

- Handled missing data carefully using imputation.
- Performed EDA to understand data distribution and correlations.
- Used 'SelectKBest' for feature selection.
- Used training-validation-test split for reliable performance estimation.

Learning Outcomes:

- Understood how to implement Linear Regression for a real-world problem.
- Practiced data preprocessing, feature selection, model training and evaluation.
- Gained insights on error metrics (MAE, MSE, R^2) to evaluate regression models.