Experiment 1: Working with Python Packages

Jayavarshini K S

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

Aim

To explore Python libraries such as NumPy, Pandas, Scikit-Learn, Matplotlib and Seaborn using a real-world dataset and perform exploratory data analysis and preprocessing.

Libraries Used

- NumPy
- Pandas
- Matplotlib
- Seaborn
- Scikit-Learn

Dataset Used

- Loan Amount Prediction
- Handwritten Character Recognition
- Email Spam Classification and MNIST
- Predicting Diabetes
- Iris Dataset

Type of ML Task

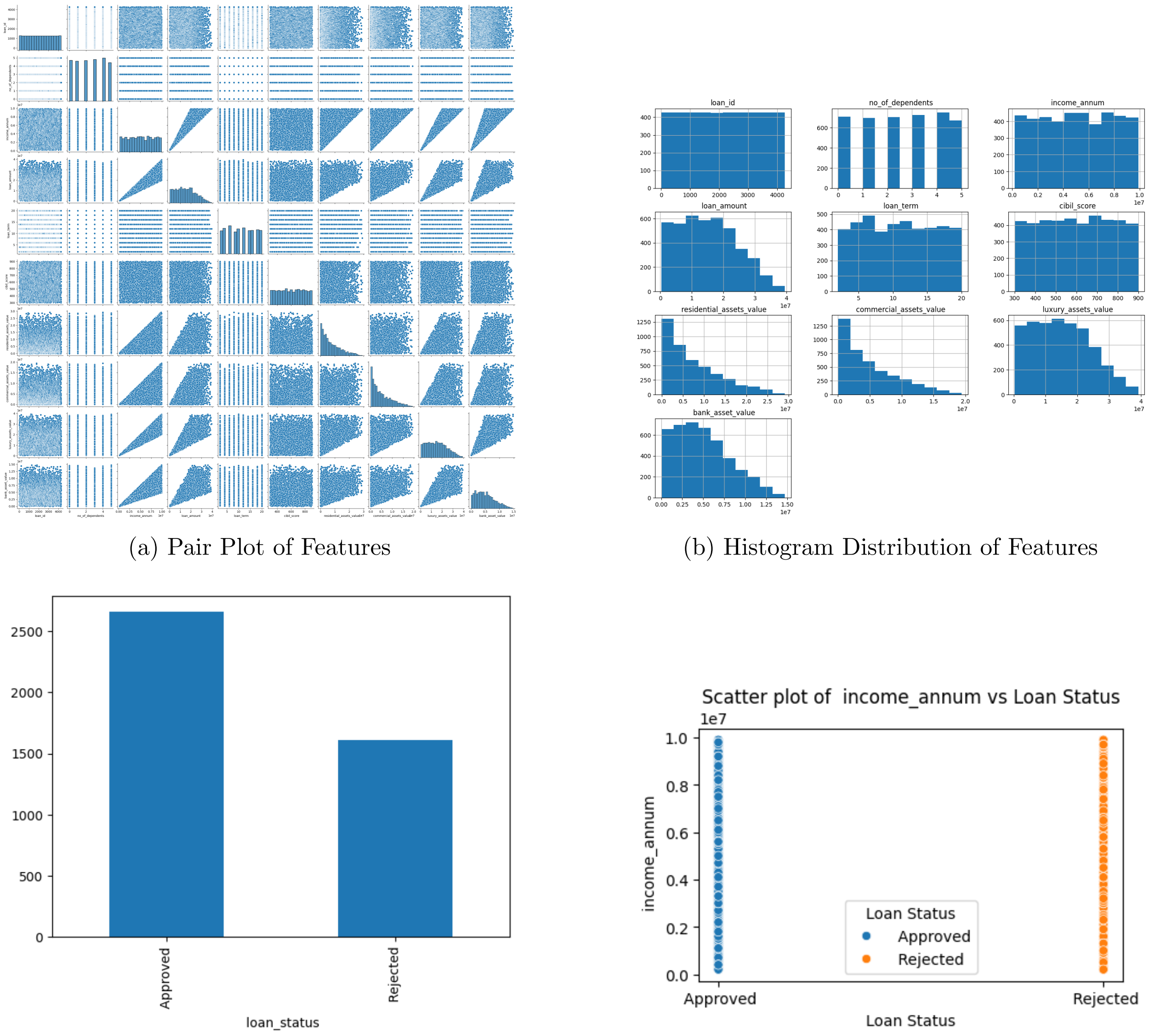
Loan Amount Prediction – Supervised Regression Problem.

Python Code

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

df = pd.read_csv("Dataset\loan_approval_dataset.csv")
df.head()
df.describe()
df.info()
df.columns

sns.pairplot(df)
df.hist(bins=10, figsize=(15,12))

df['␣loan_status'].value_counts().plot(kind='bar')

cols = ['␣income_annum', '␣loan_amount', '␣loan_term', '␣cibil_score',
       '␣residential_assets_value', '␣commercial_assets_value',
       '␣luxury_assets_value', '␣bank_asset_value']

for i in cols:
    plt.figure(figsize=(5,3))
    sns.scatterplot(data=df, x='␣loan_status', y=i, hue='␣loan_status')
    plt.title(f'Scatter␣plot␣of␣{i}␣vs␣Loan␣Status')
    plt.xlabel('Loan␣Status')
    plt.ylabel(i)
    plt.legend(title='Loan␣Status')
    plt.show()

df_copy = df.select_dtypes(include='number')
corr = df_copy.corr()
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Feature␣Correlation␣Heatmap")
plt.show()

data = [df['␣income_annum'],df[ '␣loan_amount'],df[ '␣loan_term'],df[ '␣
   cibil_score'],
       df['␣residential_assets_value'],df[ '␣commercial_assets_value'],
       df['␣luxury_assets_value'],df[ '␣bank_asset_value']]

plt.figure(figsize=(18,8))
plt.boxplot(data,tick_labels=['income_annum', 'loan_amount', 'loan_term',
   'cibil_score',
       'residential_assets_value', 'commercial_assets_value',
       'luxury_assets_value', 'bank_asset_value'])
plt.title("Feature␣Distribution␣Comparison")
plt.ylabel("Value")
plt.show()
```

```
print("Number␣of␣missing␣values␣per␣column", df.isna().sum())
print("Total␣missing␣values:␣",df.isna().sum().sum())

df['␣income_annum'] = df['␣income_annum'].fillna(df['␣income_annum'].
   median())

num_col = ['␣income_annum', '␣loan_amount']
for col in num_col:
    lower = df[col].quantile(0.1)
    upper = df[col].quantile(0.95)
    df[col] = np.clip(df[col], lower, upper)

print("Duplicates␣(ignoring␣loan_id):",
      df.drop(columns=['loan_id']).duplicated().sum())

df = df.drop_duplicates(subset=df.columns.difference(['loan_id']))

print("Duplicates␣after␣drop:",
      df.drop(columns=['loan_id']).duplicated().sum())

loan_X = df.drop(columns=['␣loan_amount'])
loan_y = df['␣loan_amount']
lX_train, lX_test, ly_train, ly_test = train_test_split(
    loan_X, loan_y, test_size=0.2, random_state=42)

from sklearn.preprocessing import LabelEncoder, RobustScaler

lX_train['␣education'] = LabelEncoder().fit_transform(lX_train['␣education
   '])
lX_train['␣loan_status'] = LabelEncoder().fit_transform(lX_train['␣
   loan_status'])
lX_train['␣self_employed'] = LabelEncoder().fit_transform(lX_train['␣
   self_employed'])

robust_list = ['␣residential_assets_value','␣commercial_assets_value','␣
   bank_asset_value']
X_scaled = RobustScaler().fit_transform(lX_train[robust_list])
```

1 Output Screenshots



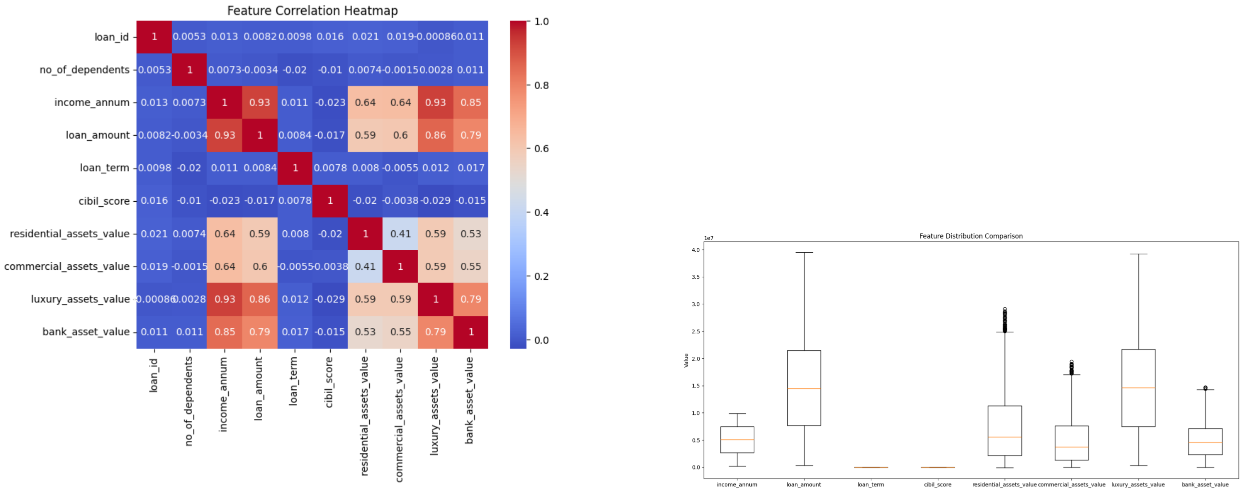
(a) Pair Plot of Features

(b) Histogram Distribution of Features

(c) Loan Status Bar Chart

(d) Scatter Plot of income vs Loan Status

Figure 1: Exploratory Data Analysis Plots



(a) Feature Correlation Heatmap

(b) Feature Distribution Comparison (Box Plot)

Figure 2: Correlation and Distribution Analysis

ML Task Identification Table

Dataset	Type of ML Task	Feature Selection	Algorithm
Iris Dataset	Classification	ANOVA	KNN, SVM
Loan Amount Prediction	Regression	SelectKBest	Linear Regression
Predicting Diabetes	Classification	Chi-Square	Logistic Regression
Email Spam Classification	Classification	Chi-Square	Naive Bayes
Handwritten Recognition (MNIST)	Classification	PCA	CNN

Inference

The experiment shows how Python libraries support data loading, visualization, preprocessing, encoding, scaling, and splitting before applying machine learning models.

Learning Outcomes

- Understood Pandas operations.
- Visualized data using Matplotlib and Seaborn.
- Applied preprocessing methods.
- Identified ML task types.