VAUTECH SOLUTIONS — AI INTERNSHIP REPORT

Task 2: Data Cleaning & Industry-Grade Preprocessing

Project Title

Industry-Grade Data Cleaning and Preprocessing for Machine Learning

Intern Name

Lokesh Girish Bharambe

Intern ID

VT26AI001

Domain

Artificial Intelligence & Data Science

Mentor

Vishal Ramkumar Rajbhar

Company

Vautech Solutions IT Solutions

Abstract

In real-world Artificial Intelligence and Machine Learning applications, raw datasets collected from industry sources are often incomplete, inconsistent, and noisy. Such data cannot be directly used for model training. Therefore, data cleaning and preprocessing become essential steps in the machine learning pipeline.

This task focuses on performing industry-grade data preprocessing using Python libraries such as Pandas, NumPy, and Scikit-learn. The preprocessing workflow includes handling missing values, removing duplicates, treating inconsistent data, detecting and removing outliers, encoding categorical variables, and scaling numerical features. The final outcome is a clean, structured, and production-ready dataset suitable for machine learning model development.

1. Introduction

Data preprocessing is one of the most critical stages in the AI/ML lifecycle. Real-world datasets often contain missing values, duplicate records, inconsistent formats, and extreme values. If such issues are ignored, they can significantly reduce model accuracy and reliability.

This task demonstrates a complete preprocessing pipeline applied to an industry-style dataset. The implemented workflow follows best practices commonly used in real-world AI and data science projects to transform raw data into a machine-readable and model-ready format.

2. Objectives

The main objectives of this task are:

- To handle missing and inconsistent data
- To remove duplicate records
- To detect and treat outliers
- To encode categorical variables

• To scale numerical features

• To generate a clean, production-ready dataset

3. Tools and Technologies Used

The following tools and technologies were used:

• Python – Programming language for data preprocessing

• Pandas – Data manipulation and cleaning

• NumPy – Numerical computations and missing value handling

• Scikit-learn – Feature encoding and scaling

4. Dataset Description

4.1 Data Source

The dataset used for this task is a Loan Prediction dataset obtained from Kaggle, a popular platform for real-world data science datasets. The dataset simulates data used by financial institutions to evaluate loan eligibility.

4.2 Data Structure

Attribute	Description
Total Records	418 (before preprocessing)
Total Features	12
Data Types	Numerical & Categorical
Missing Values	Present (? and null values)
Duplicates	Present
Outliers	Present

4.3 Feature Description

Numerical Features

- Age
- ApplicantIncome
- CoapplicantIncome
- LoanAmount
- Loan_Amount_Term

Categorical Features

- Gender
- Married
- Education
- Self_Employed

- Property_Area

Target Feature

- Loan_Status – Indicates loan approval or rejection

The dataset required extensive preprocessing before it could be used for machine learning.

5. Data Cleaning Process

5.1 Loading the Dataset

The dataset was loaded using Pandas, and its structure was inspected by viewing shape and sample records.

5.2 Handling Missing Values

- Missing values represented by ? were replaced with NaN
- Numerical columns were filled using mean imputation
- Categorical columns were filled using mode imputation

5.3 Removing Duplicate Records

Duplicate rows were removed to avoid redundancy and bias.

5.4 Handling Inconsistent Data

- Converted text data to lowercase
- Removed extra whitespaces
- Standardized categorical labels

6. Feature Encoding

Categorical features were converted into numerical format using Label Encoding, enabling machine learning algorithms to process categorical information effectively.

7. Feature Scaling

Numerical features were standardized using StandardScaler, ensuring:

- Mean = 0
- Standard Deviation = 1

This improves training stability and model performance.

8. Outlier Detection and Treatment

Outliers were detected using the Interquartile Range (IQR) method.
Rows containing extreme values beyond acceptable limits were removed to prevent model bias.

9. Implementation & Code

```
import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder, StandardScaler


# Step 1: Load dataset
```

```
data = pd.read_csv("C:\\Users\\lokes\\OneDrive\\Desktop\\Vautech Internship\\Task 2\\loan_prediction.csv")

print("Initial Dataset Shape:", data.shape)
print(data.head())

# Replace '?' with NaN values
data.replace("?", np.nan, inplace=True)

# Step 2: Handle Missing Values
num_cols = data.select_dtypes(include=np.number).columns
data[num_cols] = data[num_cols].fillna(data[num_cols].mean())

cat_cols = data.select_dtypes(include='object').columns
for col in cat_cols:
    data[col] = data[col].fillna(data[col].mode()[0])

# Step 3: Remove Duplicates
data = data.drop_duplicates()

# Step 4: Handle Inconsistent Data
for col in cat_cols:
```

```
    data[col] = data[col].str.lower().str.strip()


# Step 5: Outlier Removal using IQR
Q1 = data[num_cols].quantile(0.25)
Q3 = data[num_cols].quantile(0.75)
IQR = Q3 - Q1

data = data[~((data[num_cols] < (Q1 - 1.5 * IQR)) |
              (data[num_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]

# Step 6: Feature Encoding
le = LabelEncoder()
for col in cat_cols:
    data[col] = le.fit_transform(data[col])

# Step 7: Feature Scaling
scaler = StandardScaler()
data[num_cols] = scaler.fit_transform(data[num_cols])

# Step 8: Save Clean Dataset
data.to_csv("loan_prediction_clean.csv", index=False)
```

print("Final Clean Dataset Shape:", data.shape)

print("Clean dataset saved as loan_prediction_clean.csv")

10. Output & Results

```
PS C:\Users\lokes\OneDrive\Desktop\Vautech Internship> python -u "c:\Users\lokes\OneDrive\Desktop\Vautech Internship\Task 2\tassk 2.py"
Initial Dataset Shape: (367, 12)
    Loan_ID Gender Married Dependents     Education  ... CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  Property_Area
0  LP001015   Male     Yes          0      Graduate  ...                 0       110.0             360.0             1.0          Urban
1  LP001022   Male     Yes          1      Graduate  ...              1500       126.0             360.0             1.0          Urban
2  LP001031   Male     Yes          2      Graduate  ...              1800       208.0             360.0             1.0          Urban
3  LP001035   Male     Yes          2      Graduate  ...              2546       100.0             360.0             NaN          Urban
4  LP001051   Male      No          0  Not Graduate  ...                 0        78.0             360.0             1.0          Urban

[5 rows x 12 columns]
Final Clean Dataset Shape: (210, 12)
Clean dataset saved as loan_prediction_clean.csv
PS C:\Users\lokes\OneDrive\Desktop\Vautech Internship>
```

Initial Dataset

- Shape: (367, 12)
- Contained missing values, duplicates, and outliers

Final Clean Dataset

- Shape: (210, 12)
- No missing values
- Encoded categorical features
- Scaled numerical features
- Outliers removed

Final file saved as: loan_prediction_clean.csv

11. Results and Observations

• Significant improvement in data quality
• Dataset consistency achieved
• Reduced noise and redundancy
• Improved reliability for ML model training
• Industry-standard preprocessing pipeline implemented

12. Conclusion

Data cleaning and preprocessing are fundamental steps in building reliable AI systems. In this task, an industry-style loan prediction dataset was successfully transformed into a clean and structured format using professional preprocessing techniques. Handling missing values, encoding categorical variables, scaling numerical features, and treating outliers ensures high-quality input data for machine learning models. This task provided practical exposure to real-world data preparation techniques used in industry environments.

13. Future Scope

• Automating preprocessing using pipelines
• Applying advanced outlier detection methods
• Feature selection and dimensionality reduction
• Integration with real-time data streams
• Deployment-ready preprocessing workflows

End of Report