| Experiment # | <TO BE FILLED BY STUDENT> | Student ID | <TO BE FILLED BY STUDENT> |
|---|---|---|---|
| Date | <TO BE FILLED BY STUDENT> | Student Name | [@KLWKS_BOT] THANOS |

# Experiment # 8: Implement various Data preprocessing techniques on a given data set

Aim/Objective:

This experiment aims to implement data pre-processing techniques to clean, transform, and prepare raw data for further analysis or machine learning tasks

**Description:**

In this experiment, students will learn the importance of data pre-processing in the data science workflow. They will understand the various steps involved in cleaning and transforming raw data to make it suitable for analysis or model building. Students will implement a data pre-processing pipeline using Python and relevant libraries, gaining hands-on experience in handling missing values, outliers, categorical variables, feature scaling, and more.

**Pre-Requisites:**

Basic understanding of data types, including numerical and categorical variables.

Familiarity with Python programming and data manipulation libraries such as pandas

**Pre-Lab:**

1. Why data are dirty?

Data are dirty due to missing values, duplicates, errors, inconsistencies, and noise.

2. What is data preprocessing? Why is it important in machine learning?

It is the process of cleaning and preparing data for machine learning to improve accuracy and reliability.

3. What are some common problems that occur during data processing? How can they be fixed?

- **Missing values** → Impute (mean, median) or remove.
- **Duplicates** → Identify & delete.
- **Inconsistent formats** → Standardize.
- **Outliers** → Detect & handle using statistical methods.
- **Noisy data** → Smooth using binning or regression.

4. How do you handle the missing data?

- Delete incomplete data.
- Impute values (mean, median, mode).
- Use models that handle missing data.

5. What is the difference between missing value treatment and outliers treatment?

- **Missing values** → Imputed or removed.
- **Outliers** → Detected and adjusted to prevent model distortion.

**In-Lab:**

You are a data analyst for an online store that sells electronics, clothing, and home goods. The company wants to analyze customer behavior to improve sales. You have been provided with a dataset containing customer purchase information, but it is messy and requires preprocessing before analysis.

---

**Dataset Overview**

The dataset is stored in a CSV file called customer_purchases.csv. It contains the following columns:

| Customer_ID | Age | Gender | Product_Category | Purchase_Amount | Purchase_Date | Country | Feedback | Discount_Code |
|---|---|---|---|---|---|---|---|---|
| 101 | 25 | Male | Electronics | 250 | 2023-06-15 | USA | Positive | DISC10 |
| 102 | NaN | Female | Clothing | NaN | 2023-06-16 | USA | Neutral | NaN |
| 103 | 32 | Female | Electronics | 300 | Missing | Missing | Positive | DISC20 |
| 104 | -5 | Male | Home Goods | 450 | 2023-06-20 | Canada | Negative | DISC10 |
| 103 | 32 | Female | Electronics | 300 | Missing | Missing | Positive | DISC20 |
| 105 | 40 | Female | Clothing | 120 | 2023-06-22 | India | NaN | None |
| 102 | NaN | Female | Clothing | NaN | 2023-06-16 | USA | Neutral | NaN |

---

**Tasks**

Your goal is to clean and preprocess the dataset so it can be used for analysis and modeling. Follow these steps:

1. **Handle Missing Values**:
    - Replace missing values in numerical columns (e.g., Age, Purchase_Amount) with the column's median.
    - Replace missing or invalid values in categorical columns (Feedback, Purchase_Date, Country) with appropriate placeholders or most frequent values.
2. **Deduplication :**
    - Remove duplicate rows.
3. **Encode Categorical Data**:
    - Convert Gender, Product_Category, and Feedback into numerical labels.
4. **Normalize Numerical Columns**:
    - Normalize Purchase_Amount and Age to a scale between 0 and 1.
5. **Data Splitting:**
    - **Divide the data into X, y (input and output columns)**
    - **Divide the data into X_train, X_test, y_train and y_tes**
    - 
    - **t**

Procedure/Program:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split


data = {
   "Customer_ID": [101, 102, 103, 104, 103, 105, 102],

   "Age": [25, None, 32, -5, 32, 40, None],

   "Gender": ["Male", "Female", "Female", "Male", "Female", "Female", "Female"],

   "Product_Category": ["Electronics", "Clothing", "Electronics", "Home Goods",
"Electronics", "Clothing", "Clothing"],

   "Purchase_Amount": [250, None, 300, 450, 300, 120, None],

   "Purchase_Date": ["2023-06-15", "2023-06-16", "Missing", "2023-06-20",
"Missing", "2023-06-22", "2023-06-16"],

   "Country": ["USA", "USA", "Missing", "Canada", "Missing", "India", "USA"],

   "Feedback": ["Positive", "Neutral", "Positive", "Negative", "Positive", None,
"Neutral"],

   "Discount_Code": ["DISC10", None, "DISC20", "DISC10", "DISC20", "None", None]
}

df = pd.DataFrame(data)

df.columns = df.columns.str.replace(" ", "_")

median_age = df[df["Age"] > 0]["Age"].median()

df["Age"] = df["Age"].apply(lambda x: median_age if pd.isna(x) or x < 0 else x)

df["Purchase_Amount"].fillna(df["Purchase_Amount"].median(), inplace=True)

most_frequent_country = df[df["Country"] != "Missing"]["Country"].mode()[0]

df["Country"].replace("Missing", most_frequent_country, inplace=True)

df["Purchase_Date"].replace("Missing", "Unknown", inplace=True)

df["Feedback"].fillna("No Feedback", inplace=True)

df["Discount_Code"].fillna("No Code", inplace=True)

df.drop_duplicates(inplace=True)
```

```
label_encoders = {}

for column in ["Gender", "Product_Category", "Feedback"]:

    le = LabelEncoder()

    df[column] = le.fit_transform(df[column])

    label_encoders[column] = le

scaler = MinMaxScaler()

df[["Age", "Purchase_Amount"]] = scaler.fit_transform(df[["Age",
"Purchase_Amount"]])

X = df.drop(columns=["Feedback"])
y = df["Feedback"]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

print(df.head())
```

- **Data and Results:**

## Data

The dataset contains customer purchase records with missing and duplicate values.

## Result

After preprocessing, data is cleaned, normalized, and encoded properly.

- **Analysis and Inferences:**

## Analysis

Key trends in age, purchase amount, and customer feedback observed.

## Inferences

Data cleaning improves accuracy for customer behavior analysis and predictions.

**VIVA-VOCE Questions (In-Lab):**

1. What is the difference between normalization and standardization?

- **Normalization:** Scales data to [0,1] or [-1,1]. (Min-max scaling)
- **Standardization:** Transforms data to have mean = 0 and std = 1. (Z-score)

2. What are the different encoding techniques for categorical data?

- **One-Hot Encoding, Label Encoding, Ordinal Encoding, Frequency Encoding, Target Encoding, Binary Encoding.**

3. What are some common techniques for data reduction?

- **Dimensionality Reduction** (PCA, LDA)
- **Feature Selection** (RFE, mutual info)
- **Sampling** (Random, Stratified)
- **Aggregation** (Summarization)
- **Binning** (Grouping values)

4. How do you preprocess time-series data?

- Handle **missing values** (fill, interpolate)
- **Smooth** (moving average)
- Remove **trend/seasonality** (differencing)
- **Feature engineering** (lags, rolling stats)
- **Scale & handle outliers**

5. What is data integration and what challenges are associated with it?

- **Combining multiple data sources**
- Challenges: **Schema mismatch, duplicates, inconsistencies, scalability, ETL complexity**

**Post-Lab:**

Implement a Python program to apply various data preprocessing techniques on the following dataset.
Dataset Link:
https://catalog.data.gov/dataset/electric-vehicle-population-data/resource/fa51be35-691f-45d2-9f3e-535877965e69

**Procedure/Program:**

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
import numpy as np
import re
from google.colab import files

uploaded = files.upload()
file_name = list(uploaded.keys())[0]
df = pd.read_csv(file_name)

df.fillna({
   "County": "Unknown",
   "City": "Unknown",
   "Postal Code": df["Postal Code"].mode()[0],
   "Electric Range": df["Electric Range"].median(),
   "Base MSRP": df["Base MSRP"].median(),
   "Legislative District": df["Legislative District"].mode()[0],
   "Vehicle Location": "Unknown",
   "Electric Utility": "Unknown",
   "2020 Census Tract": df["2020 Census Tract"].mode()[0]
}, inplace=True)

label_encoders = {}
categorical_columns = ["Make", "Model", "Electric Vehicle Type", "Clean
Alternative Fuel Vehicle (CAFV) Eligibility"]

for col in categorical_columns:
   le = LabelEncoder()
   df[col] = le.fit_transform(df[col])
   label_encoders[col] = le
```

```
scaler = MinMaxScaler()
numerical_columns = ["Electric Range", "Base MSRP", "Model Year"]
df[numerical_columns] = scaler.fit_transform(df[numerical_columns])

def extract_coordinates(location):
    match = re.search(r'\(([-\d.]+) ([-\d.]+)\)', str(location))
    if match:
        return float(match.group(1)), float(match.group(2))
    return np.nan, np.nan

df['Latitude'], df['Longitude'] = zip(*df['Vehicle
Location'].apply(extract_coordinates))
df.drop(columns=['Vehicle Location'], inplace=True)

processed_file = "preprocessed_electric_vehicle_data.csv"
df.to_csv(processed_file, index=False)

files.download(processed_file)

print("Data preprocessing completed. Downloading preprocessed file...")
```

**Data and Results:**

## Data

This dataset contains information on electric vehicle registrations.

## Result

The preprocessed dataset is cleaned, encoded, and normalized successfully.

**Analysis and Inferences:**

## Analysis

Categorical data was encoded, numerical data was normalized properly.

## Inferences

The dataset is now structured for further machine learning applications.

| | |
|---|---|
| **Evaluator Remark (if Any):** | **Marks Secured _____ out of 50** |
| | **Signature of the Evaluator with Date** |