Experiment 01 - Preprocessing

| Roll No. | 65 |
| --- | --- |
| Name | Shreyansh Singh |
| Class | D15 |
| Subject | Business Intelligence Lab |
| LO Mapped | LO2:  Organize and prepare the data needed for data mining algorithms in terms of attributes and class inputs, training, validating, and testing files. |
|  |  |

**Aim**: To perform data preprocessing on the dataset using python.

**Introduction**:

Preprocessing is a term used to describe the steps taken to prepare data for analysis or modeling. It involves cleaning, transforming, and organizing raw data into a structured, easily analyzed format.

The preprocessing step is often considered one of the most critical steps in the data analysis pipeline because it can significantly impact the quality and accuracy of the results. Some common preprocessing techniques include

Data cleaning: Removing or correcting invalid, duplicate, or missing data.

Data normalization: Scaling data to an expected range or standard deviation.

Data encoding: Converting categorical data to numeric form.

Feature selection: Identifying and selecting the most relevant features for analysis.

Dimensionality reduction: Reducing the number of elements in the data to simplify analysis. Data transformation: Applying mathematical or statistical transformations to the data to normalize distributions or reduce skewness.

By performing these preprocessing steps, analysts and data scientists can improve the quality of their analyses and derive more accurate and reliable insights from their data.

Dataset Schema:

We are conducting experiments we self-made dummy dataset which includes columns like Country, Salary, Age, and Purchased.

**Data Cleaning - removing missing values**

Data Cleaning means the process of identifying the incorrect, incomplete, inaccurate, irrelevant or missing part of the data and then modifying, replacing or deleting them according to the necessity. Data cleaning is considered a foundational element of basic data science. Machine Learning is a data- driven AI. In machine learning, if the data is irrelevant or error-prone then it leads to an incorrect model building. As much as you make your data clean, as much as you can make a better model. So, we need to process or clean the data before using it. Without the quality data,it would be foolish to expect anything good*.*

***Handling Missing Data***

Missing data is defined as the values or data that is not stored (or not present) for some variable/s in the given dataset. In the dataset, blank shows the missing values. In Pandas, usually, missing values are represented by NaN.

Handling missing values is an important step in data cleaning that can impact model validity and reliability.

There are 2 primary ways of handling missing values:

* Deleting the Missing Values
* Imputing the Missing Values
* Replacing with a arbitrary/default value
* Replacing With Mean Value
* Replacing With Mode
* Replacing with previous value –
* Forward fill Replacing with next value – Backward fill

Mostly valid will be using mean value

**Data Cleaning - removing noisy values**

Noisy values, also known as outliers, are values that are significantly different from the other values in a dataset. These values can be identified using various statistical techniques. Here are some common methods for identifying noisy values:

Z-score:

The z-score measures how many standard deviations an observation is from the mean of the dataset. Values with a z-score greater than 3 or less than -3 are typically considered outliers.

Box plot:

Box plots are a graphical representation of a dataset that can be used to identify outliers. Any points that fall outside the whiskers of the box plot are considered outliers.

Tukey method:

The Tukey method is a statistical technique for identifying outliers based on the interquartile range (IQR). Values outside the range of Q1-1.5(IQR) and Q3+1.5(IQR) are considered outliers.

Once noisy values have been identified, there are several ways to handle them, including:

Removing outliers:

This approach involves simply removing any observations that are identified as outliers. However, this can lead to a loss of data and potentially bias the analysis.

Winsorizing:

Winsorizing involves replacing the noisy values with the nearest non-noisy value. This approach can reduce the impact of outliers while still preserving the overall shape of the distribution.

Transformation:

Data transformation techniques such as logarithmic or square root transformation can help reduce the impact of outliers on the analysis.

Here is an example code snippet in Python using Pandas to identify and remove outliers using the Tukey method:

kotlin

Copy code

import pandas as pd

import numpy as np

# Load the data into a DataFrame

data = pd.read\_csv('data.csv')

# Calculate the interquartile range

Q1 = np.percentile(data, 25)

Q3 = np.percentile(data, 75)

IQR = Q3 - Q1

# Identify the outliers

outliers = data[(data < Q1-1.5\*IQR) | (data > Q3+1.5\*IQR)]

# Remove the outliers

data = data[(data >= Q1-1.5\*IQR) & (data <= Q3+1.5\*IQR)]

It's important to note that the approach to handling noisy values should be determined based on the nature of the data and the analysis requirements. Removing noisy values can be appropriate in some cases, but it can also lead to a loss of information and potential bias. Therefore, it's essential to carefully consider the trade-offs when deciding how to handle noisy values in a dataset.

**Data Transformation**

Data transformation is the process of converting raw data into a format or structure that would be more suitable for model building and also data discovery in general. It is an imperative step in feature engineering that facilitates discovering insights.

### Label Encoding This approach is very simple and it involves converting each value in a column to a number. Consider a dataset of bridges having a column named bridge-types having below values. Though there will be many more columns in the dataset, to understand label-encoding, we will focus on one categorical column only.

preprocessing.LabelEncoder() of sklearn is used for label encoding.

In our dataset we use this on Vehicle Style and Model

***One-Hot Encoder***

Though label encoding is straight, it has the disadvantage that the numeric values can be misinterpreted by algorithms as having some sort of hierarchy/order in them. This ordering issue is addressed in another common alternative approach called ‘One-Hot Encoding’. In this strategy, each category value is converted into a new column and assigned a 1 or 0 (notation for true/false) value to the column. Let’s consider the previous example of bridge type and safety levels with one-hot encoding.

In our dataset we use this on Transmission Type and Vehicle Size

**Data Normalization**

Feature scaling refers to putting the values in the same range or same scale so that no variable is dominated by the other.

Numerical data in the dataset can have a varied range i.e. one parameter may lie between 1 to 10 for all records whereas another parameter can lie between 1000 to 5000. Though data is logically correct but after passing to a particular algorithm, the features with higher magnitude become key parameters for that algorithm.

To avoid such situations feature scaling is performed using some statistical techniques like Min-Max scaling & Mean normalization. This creates a common range for all the parameters and thus removes Algorithmic bias.

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

**Data Reduction**

***Dimensionality Reduction***

*Any location* having less than 10 data points should be tagged as "other" location. This way number of categories can be reduced by huge amount. Later on when we do one hot encoding, it will help us with having fewer dummy columns

**Conclusion**:

Data preprocessing is a crucial step in any data analysis project as it helps to clean and transform raw data into a usable format for analysis. In this experiment, we discussed two common data preprocessing techniques: handling missing values and identifying and handling noisy values.

For handling missing values, we explored several techniques, including dropping missing values, replacing missing values with a default value or mean value, and replacing missing values with a grouped mean value. The most appropriate approach depends on the type of data and the analysis requirements. In general, dropping missing values should be used with caution as it can lead to a loss of data and potential bias.

For identifying and handling noisy values, we discussed various statistical techniques such as z-score, box plot, and Tukey method. The noisy values can be handled by removing outliers, winsorizing, or transforming the data.

Overall, data preprocessing is an iterative process that requires careful consideration of the trade-offs between different techniques and their impact on the final analysis. By effectively handling missing and noisy values, data preprocessing can help to ensure accurate and reliable results in data analysis projects.