NAAN MUDHALAVAN : IBM

PHASE : 3

TECHNOLOGY : DATA SCIENCE

PROJECT TITLE : DEVELOPMENT PART 1

Data Preprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task

some stems is there in preprocessing :

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

Get the Dataset To create a machine learning model, the first thing required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the **dataset**.

```
# check for null values
dataset.isnull().shape[0]
print("Non-missing values: " + str(dataset.isnull().shape[0]))
print("Missing values: " + str(dataset.shape[0] - dataset.isnull().shape[0]))
Non-missing values: 284807
Missing values: 0
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler().fit(dataset[["Time", "Amount"]])
dataset[["Time", "Amount"]] = scaler.transform(dataset[["Time", "Amount"]])
dataset.head().append(dataset.tail())
```

OUTPUT:

```
╓─────────┬───────────┬────────────┬───────────┬───────────┬───────────┬──
──────────┬───────────┬───────────┬───────────┬───────────┬─────┬─────────
┬───────────┬───────────┬───────────┬───────────┬───────────┬───────────┬─
──────────┬───────────┬───────╖
║         │ Time      │ V1         │ V2        │ V3        │ V4        │ V5
│ V6        │ V7        │ V8        │ V9        │ ... │ V21       │ V22
│ V23       │ V24       │ V25       │ V26       │ V27       │ V28       │
Amount    │ Class ║
╟═════════╪═══════════╪════════════╪═══════════╪═══════════╪═══════════╪══
══════════╪═══════════╪═══════════╪═══════════╪═══════════╪═════╪═════════
╪═══════════╪═══════════╪═══════════╪═══════════╪═══════════╪═══════════╪═
══════════╪═══════════╪═══════╢
║ 0       │ -0.994983 │ -1.359807  │ -0.072781 │ 2.536347  │ 1.378155  │ -
0.338321 │ 0.462388  │ 0.239599  │ 0.098698  │ 0.363787  │ ... │ -0.018307
│ 0.277838  │ -0.110474 │ 0.066928  │ 0.128539  │ -0.189115 │ 0.133558  │
-0.021053 │ 1.783274  │ 0       ║
╟─────────┼───────────┼────────────┼───────────┼───────────┼───────────┼──
──────────┼───────────┼───────────┼───────────┼───────────┼─────┼─────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
║ 1       │ -0.994983 │ 1.191857   │ 0.266151  │ 0.166480  │ 0.448154  │
0.060018  │ -0.082361 │ -0.078803 │ 0.085102  │ -0.255425 │ ... │ -
0.225775 │ -0.638672 │ 0.101288  │ -0.339846 │ 0.167170  │ 0.125895  │ -
0.008983 │ 0.014724  │ -0.269825 │ 0         ║
╟─────────┼───────────┼────────────┼───────────┼───────────┼───────────┼──
──────────┼───────────┼───────────┼───────────┼───────────┼─────┼─────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
```

```
║ 2      │ -0.994972 │ -1.358354  │ -1.340163 │ 1.773209  │ 0.379780  │ -
0.503198 │ 1.800499  │ 0.791461  │ 0.247676  │ -1.514654 │ ... │ 0.247998
│ 0.771679  │ 0.909412  │ -0.689281 │ -0.327642 │ -0.139097 │ -0.055353 │
-0.059752 │ 4.983721  │ 0       ║
╟────────┼───────────┼────────────┼───────────┼───────────┼───────────┼───
─────────┼───────────┼───────────┼───────────┼───────────┼─────┼──────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
║ 3      │ -0.994972 │ -0.966272  │ -0.185226 │ 1.792993  │ -0.863291 │ -
0.010309 │ 1.247203  │ 0.237609  │ 0.377436  │ -1.387024 │ ... │ -0.108300
│ 0.005274  │ -0.190321 │ -1.175575 │ 0.647376  │ -0.221929 │ 0.062723  │
0.061458  │ 1.418291  │ 0       ║
╟────────┼───────────┼────────────┼───────────┼───────────┼───────────┼───
─────────┼───────────┼───────────┼───────────┼───────────┼─────┼──────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
║ 4      │ -0.994960 │ -1.158233  │ 0.877737  │ 1.548718  │ 0.403034  │ -
0.407193 │ 0.095921  │ 0.592941  │ -0.270533 │ 0.817739  │ ... │ -0.009431
│ 0.798278  │ -0.137458 │ 0.141267  │ -0.206010 │ 0.502292  │ 0.219422  │
0.215153  │ 0.670579  │ 0       ║
╟────────┼───────────┼────────────┼───────────┼───────────┼───────────┼───
─────────┼───────────┼───────────┼───────────┼───────────┼─────┼──────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
║ 284802 │ 1.034951  │ -11.881118 │ 10.071785 │ -9.834783 │ -2.066656 │ -
5.364473 │ -2.606837 │ -4.918215 │ 7.305334  │ 1.914428  │ ... │ 0.213454
│ 0.111864  │ 1.014480  │ -0.509348 │ 1.436807  │ 0.250034  │ 0.943651  │
0.823731  │ -0.296653 │ 0       ║
╟────────┼───────────┼────────────┼───────────┼───────────┼───────────┼───
─────────┼───────────┼───────────┼───────────┼───────────┼─────┼──────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
║ 284803 │ 1.034963  │ -0.732789  │ -0.055080 │ 2.035030  │ -0.738589 │
0.868229  │ 1.058415  │ 0.024330  │ 0.294869  │ 0.584800  │ ... │ 0.214205
│ 0.924384  │ 0.012463  │ -1.016226 │ -0.606624 │ -0.395255 │ 0.068472  │
-0.053527 │ 0.038986  │ 0       ║
╟────────┼───────────┼────────────┼───────────┼───────────┼───────────┼───
─────────┼───────────┼───────────┼───────────┼───────────┼─────┼──────────
┼───────────┼───────────┼───────────┼───────────┼───────────┼───────────┼─
──────────┼───────────┼───────╢
║ 284804 │ 1.034975  │ 1.919565   │ -0.301254 │ -3.249640 │ -0.557828 │
2.630515  │ 3.031260  │ -0.296827 │ 0.708417  │ 0.432454  │ ... │ 0.232045
```

```
│ 0.578229  │ -0.037501 │ 0.640134  │ 0.265745  │ -0.087371 │ 0.004455  │
-0.026561 │ 0.641096  │ 0       ║
╟─────────┼────────────┼─────────────┼────────────┼────────────┼────────────┼───
────────┼────────────┼────────────┼────────────┼────────────┼──────┼────────────
┼────────────┼────────────┼────────────┼────────────┼────────────┼────────────┼─
──────────┼────────────┼────────║
║ 284805 │ 1.034975  │ -0.240440  │ 0.530483  │ 0.702510  │ 0.689799  │ -
0.377961 │ 0.623708  │ -0.686180 │ 0.679145  │ 0.392087  │ ...  │ 0.265245
│ 0.800049  │ -0.163298 │ 0.123205  │ -0.569159 │ 0.546668  │ 0.108821  │
0.104533  │ -0.167680 │ 0       ║
╟─────────┼────────────┼─────────────┼────────────┼────────────┼────────────┼───
────────┼────────────┼────────────┼────────────┼────────────┼──────┼────────────
┼────────────┼────────────┼────────────┼────────────┼────────────┼────────────┼─
──────────┼────────────┼────────║
║ 284806 │ 1.035022  │ -0.533413  │ -0.189733 │ 0.703337  │ -0.506271 │ -
0.012546 │ -0.649617 │ 1.577006  │ -0.414650 │ 0.486180  │ ...  │ 0.261057
│ 0.643078  │ 0.376777  │ 0.008797  │ -0.473649 │ -0.818267 │ -0.002415 │
0.013649  │ 2.724796  │ 0       ║
╚═════════╧════════════╧═════════════╧════════════╧════════════╧════════════╧═══
════════╧════════════╧════════════╧════════════╧════════════╧══════╧════════════
╧════════════╧════════════╧════════════╧════════════╧════════════╧════════════╧═
══════════╧════════════╧════════
```

Data pre-processing:

```
 train_test_split
from sklearn.preprocessing import StandardScaler

scalar = StandardScaler()

X = df.drop('Class', axis=1)
y = df.Class
X = scalar.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
```

Importing Libraries

In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing**.**

Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:

```
    import numpy as np
```

Here we have used **nm**, which is a short name for Numpy, and it will be used in the whole program.

Matplotlib: The second library is **matplotlib**, which is a Python 2D plotting library, and with this library, we need to import a sub-library **pyplot**. This library is used to plot any type of charts in Python for the code.

```
import matpoltlib pyplot as mtp
```

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. It will be imported as below:

```
import pandas as pd
```

Handling Missing data:

The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Ways to handle missing data:

There are mainly two ways to handle missing data, which are:

By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

To handle missing values, we will use Scikit-let

Encoding Categorical data:

Categorical data is data which has some categories such as, in our dataset; there are two categorical variable, **Country**, and **Purchased**.

Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

For Country variable:

Firstly, we will convert the country variables into categorical data. So to do this, we will use **LabelEncoder()** class from **preprocessing** librar

learn library in our code, which contains various libraries for building machine learning models. Here we will use **Imputer** class of **sklearn.preprocessing** library.

```
# categorical data
#for Country Variable
from sklearn.preprocessing import LabelEncoder
label_encoder_x= LabelEncoder()
x[:, 0]= label_encoder_x.fit_transform(x[:, 0])
```

Splitting the Dataset into the Training set and Test set

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code

```
from sklearn.model_selection import train_test_split
 x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, 
random_state=0)
```

Feature Scaling

Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no any variable dominate the other variable.