Inone of my previous posts, I talked about Data Preprocessing in Data Mining & Machine Learning conceptually. This will continue on that, if you haven’t read it, read it here in order to have a proper grasp of the topics and concepts I am going to talk about in the article.

Data Preprocessing refers to the steps applied to make data more suitable for data mining. The steps used for Data Preprocessing usually fall into two categories:

selecting data objects and attributes for the analysis.
creating/changing the attributes.
Inthis post I am going to walk through the implementation of Data Preprocessing methods using Python. I will cover the following, one at a time:

Importing the libraries
Importing the Dataset
Handling of Missing Data
Handling of Categorical Data
Splitting the dataset into training and testing datasets
Feature Scaling
For this Data Preprocessing script, I am going to use Anaconda Navigator and specifically Spyder to write the following code. If Spyder is not already installed when you open up Anaconda Navigator for the first time, then you can easily install it using the user interface.

If you have not code in Python beforehand, I would recommend you to learn some basics of Python and then start here. But, if you have any idea of how to read Python code, then you are good to go. Getting on with our script, we will start with the first step.

Importing the libraries

```
# libraries
import numpy as np # used for handling numbers
import pandas as pd # used for handling the dataset
from sklearn.impute import SimpleImputer # used for handling missing data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder # used for encoding
categorical data
from sklearn.model_selection import train_test_split # used for splitting training and…
```