* What is Data Cleaning?
* Methods For Data Cleaning
* What is Data Normalization?
* Methods for Data Normalization
* Cleaning and Normalization In Python
* Conclusion

**What is Data Cleaning?**

Data Cleaning is a critical aspect of the domain of data management. The data cleansing process involves reviewing all the data present within a database to either remove or update information that is incomplete, incorrect or duplicated and irrelevant. Data cleansing is just not simply about erasing the old information to make space for new data, but the process is about rather finding a way to maximize the dataset’s accuracy without necessarily tampering with the data available. Data Cleaning is the process of determining and correcting the wrong data. Organizations rely on data for most things but only a few properly address the data quality.

Utilizing the effectiveness and use of data can tremendously increase the reliability and value of the brand.  Hence, Business enterprises have started giving more importance to data quality. Data Cleaning includes many more actions than just removing the data, the process also requires fixing wrong spellings and syntactical errors, correcting and filling of empty fields, and identifying duplicate records to name a few. Data cleaning is considered a foundational step for data science basics, as it plays an important role in an analytical process that helps uncover reliable answers. Improving the data quality through data cleaning can eliminate problems like expensive processing errors and incorrect invoices.

Data quality is also very important as several pieces of data like customer information are always changing and evolving. Although there is no one such absolute way to describe the precise steps in the data cleaning process as the processes vary from dataset to dataset it plays an important part in deriving reliable answers. The crucial motive of data cleaning is to construct uniform and standardized data sets that enable the analytical tools and business intelligence easy access and help to perceive accurate data for each problem.

**Methods For Data Cleaning**

There are several techniques for producing reliable and hygienic data through data cleaning. Some of the data cleaning methods are as follows :

* The first and basic need in data cleaning is to remove the unwanted observations. This process includes removing duplicate or irrelevant observations. Irrelevant observations are ones that do not fit with the problem statement one is trying to solve. Making sure that the data is irrelevant and that you won’t need to clean it again is a great way to begin.
* Getting rid of unwanted outliers is another method because outliers can cause problems with certain models. Removing outliers will not only help with the model’s performance but also improve its accuracy. Although one should make sure that there is a legitimate reason to remove them.
* Small mistakes are often made when numbers are entered. If there are any mistakes present with the numbers being entered, it needs to be changed to actual readable data. All of the data presents will have to be converted so that the numbers are readable by the system. Data types should be uniform across all of the datasets. A string can’t be termed as numeric nor can a numeric be a boolean value.
* Correcting missing values is another important method that can’t be ignored. Knowing how to handle missing values will help keep the data clean. At times, there might be too many missing values present in a single column. For such occurrences, there might not be enough data to work with, so deleting the column may stand as the best option in such cases. There are also other different ways to impute missing data values in the dataset. This can be done by estimating what the missing data might just be and performing linear regression or median can help calculate this.
* Fixing the typos as a result of human error is important and one can fix typos through multiple algorithms and techniques. One of the methods can be to map the values and convert them into their correct spelling. Typos are essential to fix because models treat different values differently. Strings present in data rely a lot on their spellings and cases.

**What is Data Normalization?**

Normalization is the process of organizing data from a database. This includes processes like creating tables and establishing relationships between those tables according to the rules designed to protect the data as well as to make the database more flexible by eliminating the redundancy and inconsistency present. Data normalization is the method of organizing data to appear similar across all records and fields. Performing so always results in getting higher quality data. This process basically includes eliminating unstructured data and duplicates in order to ensure logical data storage.

When data normalization is performed correctly a higher value of insights are generated. In machine learning, some feature values at times differ from others multiple times. The features with higher values will always dominate the learning process. However, it does not mean that those variables are more important to predict the outcome of the model. Data normalization transforms the multiscaled data all to the same scale. After normalization, all variables have a similar weightage on the model, hence improving the stability and performance of the learning algorithm. Normalization gives equal importance to each variable so that no single variable drives the model performance. It also prevents any issues created from database modifications such as insertions, deletions, and updates.

For businesses to achieve further heights needs to regularly perform data normalization. It is one of the most important things that can be done to get rid of errors that make an analysis of data a complicated and difficult task. With normalization, an organization can make the most of its data as well as invest in data gathering at a greater, more efficient level.

**Methods for Data Normalization**

There are several methods for performing Data Normalization, few of the most popular techniques are as follows :

* The min-max normalization is the simplest of all methods as it converts floating-point feature values from their natural range into a standard range, usually between 0 and 1. It is a good choice when one knows the approximate upper and lower bounds on the data with few or no outliers and the data are approximately uniformly distributed across that range.
* Decimal place normalization can be done with data tables having numerical data types. By default, the algorithm places two digits after the decimal for normal comma-separated numbers. One can decide on how many decimals are required to scale this throughout the table.
* Z-score normalization is a methodology of normalizing the data and hence helps avoid the issue of outliers in the data. Here, μ is considered the mean value of the feature and σ is the standard deviation from the data points. If a value comes to be equal to the mean of all the values present, only then will it be normalized to 0. If it is below the mean value, it will be considered to be a negative number, and if it is above the mean value it will be termed as a positive number. The number of the negative and positives are determined by the standard deviation of the original feature.

**Cleaning and Normalization of Data In Python**

Using Python we can easily perform Data Cleaning and Normalization. Here I have tried to demonstrate an example using Python and Numpy, to demonstrate the difference between the general data and normalized data.

import pandas as pd

import numpy as np

from scipy import stats

from mlxtend.preprocessing import minmax\_scaling

import seaborn as sns

import matplotlib.pyplot as plt

You might highly consider normalizing your data if you are going to implement a machine learning or statistical technique that comes with the assumption of your data being normally distributed. Some examples of these include linear regression, T-tests and Gaussian naive Bayes. Now let’s see what normalizing data looks like:

normalized\_data = stats.boxcox(original\_data)

fig, ax=plt.subplots(1,2)

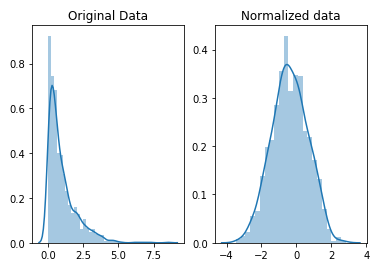
sns.distplot(original\_data, ax=ax[0])

ax[0].set\_title("Original Data")

sns.distplot(normalized\_data[0], ax=ax[1])

ax[1].set\_title("Normalized data")

Output :



We can notice that the shape of our data has changed. Before normalizing it was almost L-shaped, but after normalizing, it looks more like a bell-shaped curve. This helps us ensure that the data is now properly skewed and scaled!