Understanding Data
Student Training Program on AI/ML at IIIT Hyderabad – May 2025

Dr. Monalisa Patra

June 2, 2025

Contents

The data analysis pipeline refers to the process and tools used to gather, analyze the raw data, and present the results in an understandable format.

1. Raw data collected and transformed into a usable format (XML, JSON, .csv, Excel, etc.).
2. Data Cleaning: Handling duplicates, missing data.
3. Data Exploration: Statistical summary, Identifying patterns and potential issues in data.
4. Data Transformation: Normalizing or scaling data.
5. Data Visualization: Using plots for: Descriptive Statistics. Correlations with Bivariate Analysis.

1 Missing Data

Missing data is a common problem in data analysis. It can occur for a variety of reasons, such as data entry errors, incomplete surveys, or sensor malfunctions. Missing data can cause problems in data analysis because it can bias the results. For example, if a large number of data points are missing, the results of a statistical analysis may be inaccurate. Several techniques can be used to handle missing data. Some of the most common techniques include:

Deleting missing data: This is the simplest technique, but it can also be the most destructive. It involves removing rows or columns that contain missing values. This can be a good option if the number of missing values is small or if the missing values are not concentrated in a particular area of the dataset. Deleting missing data can, however, reduce the size of the dataset and can bias the results of the analysis.

Imputing missing data: This technique involves replacing missing data with estimated values. The missing values can be replaced with the mean or median of the non-missing values in the same column. In the case of a categorical column, we can replace it with the mode value of the column. This technique can be a good option, if the number of missing values is large or if the missing values are concentrated in a particular area of the dataset. However, there is a chance of introducing bias into the data with this method.

The goals of the analysis, along with the dataset in question, will let us decide the best technique required for handling the missing data. It is important to carefully consider the impact of missing data on the analysis before deciding how to handle it.

2 Outliers

Outliers in machine learning refer to data points that significantly deviate from the majority of the dataset, as shown in Fig. 1 by the red points. These data points can have a substantial impact on the statistical analysis or model training process, potentially leading to inaccurate results or biased models. The outliers in a dataset can occur due to various reasons, including:

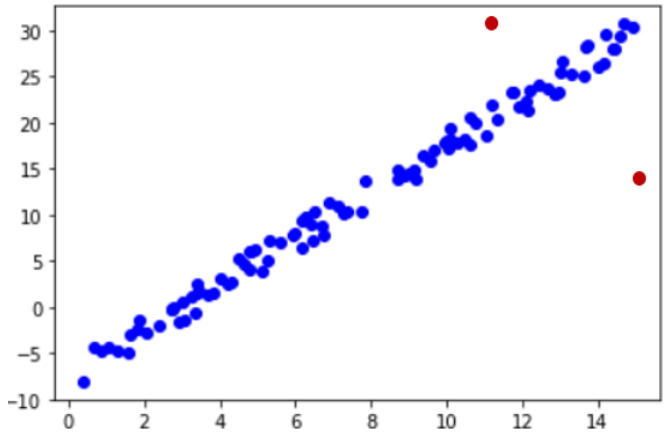


Figure 1: The red dots in the figure are outliers as they do not follow the regular trend of the data.

1. Measurement errors: Outliers may result from errors during data collection or recording. These errors could be caused by human mistakes, malfunctioning instruments, or data entry errors.

2. Natural variation: In some cases, outliers may arise naturally due to the inherent variability of the phenomenon being observed. These outliers represent extreme or rare events or observations that do not follow the typical pattern of the majority of the data.

3. Data preprocessing errors: Outliers can be introduced during data preprocessing steps such as data transformation or imputation. These errors may occur due to inappropriate handling of missing values or data manipulation techniques.

4. Data entry mistakes: Outliers can be caused by typographical errors or incorrect data entry. For example, a decimal point may be misplaced, resulting in a value that is significantly different from the rest of the dataset.

5. Sampling issues: Outliers can arise when the sampling process is biased or not representative of the overall population. If the sampling process does not capture the full range of variability, outliers may occur.

6. Intentional outliers: In some cases, outliers may be deliberately introduced into the dataset to test the robustness of a model or to examine the behavior of the system under extreme conditions.

It is important to understand the underlying causes of outliers in a dataset as it can inform the appropriate handling and treatment of these data points during data analysis and modeling. Detecting and handling outliers is an important step in data preprocessing. To detect outliers in a dataset, various statistical and machine learning techniques can be used. Here are some common approaches:

Statistical Methods:

Z-Score: A Z-score is a statistical measure that indicates how many standard deviations a specific data point is away from the mean of a data set.

$$Z = \frac{x - \mu}{\sigma}, \tag{1}$$

where x is the data point, μ is the mean of the data set and σ is the standard deviation of the data set. The Z-score for each data point is calculated, and the points with a Z-score beyond a threshold (typically a Z-score greater than or equal to $|3|$) are considered as outliers.

Interquartile Range (IQR): The IQR is a measure of the variability of a data set, based on dividing the data into quartiles. Quartiles divide a rank-ordered data set into four equal parts so that each part contains 25% of the data. The first quartile (Q_1) is the middle value between the smallest value and the median of the data set. The third quartile (Q_3) is the middle value between the median and the largest value of the data set. The IQR is calculated by subtracting Q_1 from Q_3. The IQR can also be used to identify outliers. The IQR is calculated, and the lower bound is found by subtracting 1.5 times the IQR from Q_1. The upper bound is obtained by adding 1.5 times the IQR to Q_3. Any data points that fall outside the lower bound or upper bound are considered outliers.

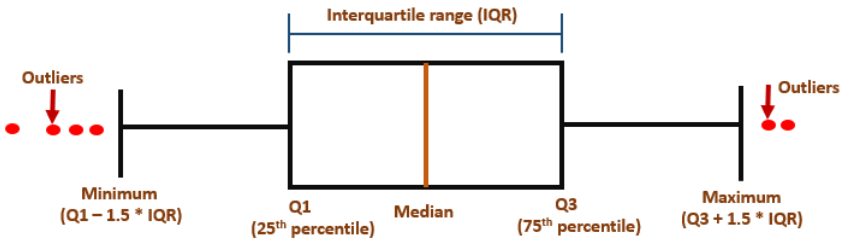


Figure 2: Diagrammatic representation of the box-plot

A box plot is a graphical representation of a data set that shows the median, quartiles, and outliers, as shown in Fig. 2. The box limits indicate the range of the central 50% of the data, with a central line marking the median value. Lines extend from each box to capture the range of the remaining data, with dots placed past the line edges to indicate outliers. Outliers are represented by the points that fall outside the whiskers in the box plot. It is important to note that the choice of handling outliers depends on the specific context and the characteristics of the dataset. A careful analysis of the data and domain knowledge is crucial for making informed decisions regarding outlier detection and handling.

3 Normalization

Normalization of data is required in machine learning to ensure that all features are on a comparable scale. This is important because it helps to prevent features with larger magnitudes from dominating the model and skewing the results. Normalization can also help to improve the speed and accuracy of machine learning algorithms. There are two main types of normalization: Min-max normalization and Z-score normalization. The best type of normalization to use depends on the specific machine learning algorithm that is being used. Some algorithms, such as support vector machines, are more sensitive to normalization than others. It is important to experiment with different normalization techniques to find the one that works best for a particular algorithm and dataset.

1. A Z-score normalization or standardization is a statistical measure that describes how far a specific data point is from the mean of a group of data points. It is mainly used to identify outliers as described earlier. It is calculated by subtracting the mean from the data point and

then dividing it by the standard deviation.

$$\text{z score} = \frac{x - \mu}{\sigma}$$

Z-scores are used to compare data points that are measured on different scales. It converts data to a common scale (mean 0, std 1), thereby making it easier to compare data from different units (e.g., heights in feet vs. weights in kg). For example, you could use z-scores to compare the heights of students in different classes, even if the classes use different measuring systems. Let's say we have a dataset of the heights of students. The mean height is 6 feet and the standard deviation is 0.5 feet. If we have a student who is 6.5 feet tall, their Z-score would be: $Z = (6.5 - 6) / 0.5 = 1$. This means that the student who is 6.5 feet tall is 1 standard deviation above the mean height. The Z-scores are sensitive to outliers and can lose information about the original scale of the data. If there are extreme values in your dataset, they will inflate the standard deviation, which can distort Z-scores. This means some "normal" points may appear unusual, and vice versa. An example is demonstrated in Fig. 3.

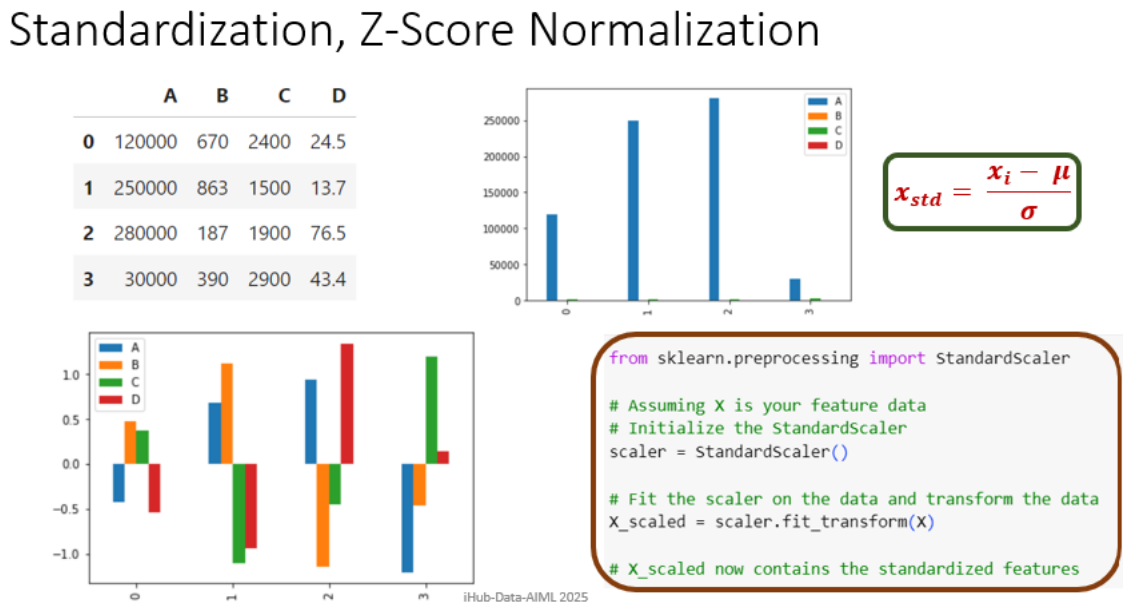


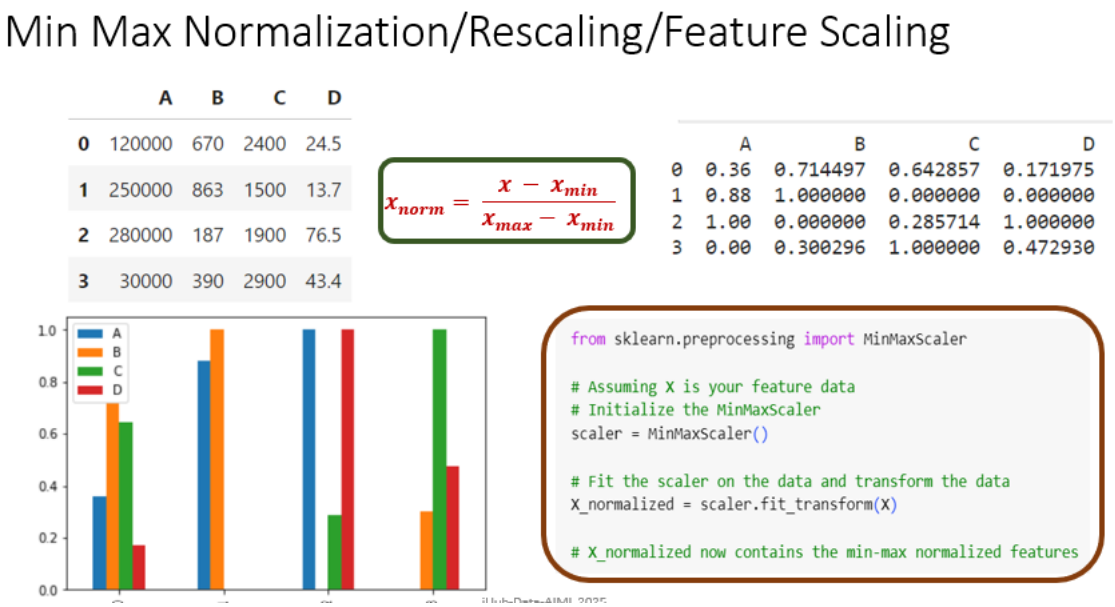
Figure 3: Standardization/Z-score Normalization

2. Min-max normalization is a data normalization technique that scales the values of each feature to a range of 0 to 1. This is done by subtracting the minimum value of the feature from each value and then dividing by the difference between the maximum and minimum values. The mathematical formula for min-max normalization is as follows:

$$x_{norm} = \frac{(x - min(\vec{x}))}{(max(\vec{x}) - min(\vec{x}))} \tag{2}$$

where, x_{norm} is the normalized value of x, x is the data point, $min(\vec{x})$ is the minimum value of the feature vector $\vec{x}$, and $max(\vec{x})$ is the maximum value of the feature vector $\vec{x}$. Data in this case has to be uniformly distributed across the range, e.g., age. Scaling on income will not be a good choice, as only a small set of people may have high income. This normalization technique is influenced by outliers. An example is demonstrated in Fig. 4.

3. Log normalization is a method of data normalization that transforms the data into a logarithmic scale. This can be useful for features that have a wide range of values or high variance, as it can help to reduce the impact of outliers. It makes it easier for us to identify patterns and trends and compare different data sets. The mathematical formulation for log normalization is $y = \log(x)$, where y is the normalized value, x is the original value, and log is the natural logarithm function.



	A	B	C	D
0	120000	670	2400	24.5
1	250000	863	1500	13.7
2	280000	187	1900	76.5
3	30000	390	2900	43.4

	A	B	C	D
0	0.36	0.714497	0.642857	0.171975
1	0.88	1.000000	0.000000	0.000000
2	1.00	0.000000	0.285714	1.000000
3	0.00	0.300296	1.000000	0.472930

Figure 4: Min-Max Normalization

Say, for example, we have the following data: $x = [1, 10, 100, 1000]$, $y = [0, 1, 2, 3]$. We can see that the log normalization has transformed the data into a scale where the values are all positive and have a smaller range. This can be useful for data analysis, as it can make it easier to identify patterns and trends. However, it is important to note that it is not a perfect solution. Log normalization can introduce bias into the data, and it can also make it more difficult to interpret the results. As with any data analysis technique, it is important to use log normalization with caution and to understand its limitations.

4. Feature Clipping/Capping takes care of outliers, by limiting or bounding the extreme values of a feature within a specified range. The upper and lower thresholds are typically determined based on a specific percentile or a fixed value. The feature is constrained within a specific range. An example is demonstrated in Fig. 5.

Link to Tutorials

- Seaborn Tutorials
- Matplotlib Tutorials
- Working with missing data in Pandas

```python
import numpy as np

def clip_feature(feature, lower_threshold, upper_threshold):
    """
    Clip the feature values between lower_threshold and upper_threshold.
    """
    clipped_feature = np.clip(feature, lower_threshold, upper_threshold)
    return clipped_feature

# Example usage
feature = np.array([10, 15, 200, 5, 25, 180])
lower_threshold = 0
upper_threshold = 100

clipped_feature = clip_feature(feature, lower_threshold, upper_threshold)
print("Original feature:", feature)
print("Clipped feature:", clipped_feature)
```

```
Original feature: [ 10  15 200   5  25 180]
Clipped feature: [ 10  15 100   5  25 100]
```

iHub-Data-AIML 2025

Figure 5: Feature Clipping