

Machine Learning

Video 21:

EDA using Bivariate and Multivariate Analysis:

Bivariate analysis examines the relationship between two variables, using techniques like scatter plots, correlation, and cross-tabulation. Multivariate analysis extends this to more than two variables, employing methods like multiple regression, PCA, or clustering to identify patterns, interactions, and dependencies across complex datasets. Both aid in data-driven insights.

Code link:

<https://colab.research.google.com/drive/1g-KvVcSVfrkLhZn4XpTpW8BfXJm59nL6?usp=sharing>

Dataset link:

<https://github.com/campusx-official/100-days-of-machine-learning/tree/main/day21-bivariate-analysis>

Here are key takeaways from the video on EDA using Bivariate and Multivariate Analysis:

1. **Import Libraries:** Use pandas for data manipulation and seaborn for visualization.
2. **Load Datasets:** `sns.load_dataset()` for built-in datasets like 'tips', 'flights', 'iris'. `pd.read_csv()` for custom datasets.
3. **Scatterplot:** For numerical vs numerical data (e.g., `sns.scatterplot(x='total_bill', y='tip', data=tips)`).
4. **Bar Plot:** For categorical vs numerical data (e.g., `sns.barplot(x='Pclass', y='Age', data=titanic)`).
5. **Multivariate Bar Plot:** Use hue to differentiate categories (e.g., `sns.barplot(x='Pclass', y='Fare', hue='Sex', data=titanic)`).
6. **Box Plot:** Visualizes numerical vs categorical data (e.g., `sns.boxplot(x='day', y='total_bill', data=tips)`).
7. **Multivariate Box Plot:** Compare numerical vs categorical data with additional category differentiation (e.g., `sns.boxplot(x='day', y='total_bill', hue='sex', data=tips)`).
8. **Displot:** Use for distribution of numerical data (e.g., `sns.displot(x='total_bill', data=tips)`).
9. **Heatmap:** For categorical vs categorical relationships (e.g., `pd.crosstab(titanic['Pclass'], titanic['Survived'])` and `sns.heatmap()`).
10. **Clustermap:** Visualizes hierarchical clustering between categories (e.g., `sns.clustermap(pd.crosstab(titanic['SibSp'], titanic['Survived']))`).
11. **PairPlot:** For visualizing relationships between multiple numerical variables (e.g., `sns.pairplot(iris)`), with hue for categorical differentiation.
12. **Lineplot:** Use for numerical vs numerical trends over time (e.g., `sns.lineplot(x='year', y='passengers', data=flights)`).
13. **Pivot Heatmap:** Use `pivot_table` for visualizing data trends over two categorical axes (e.g., `sns.heatmap(flights.pivot_table(values='passengers', index='month', columns='year'))`).

Video 22:

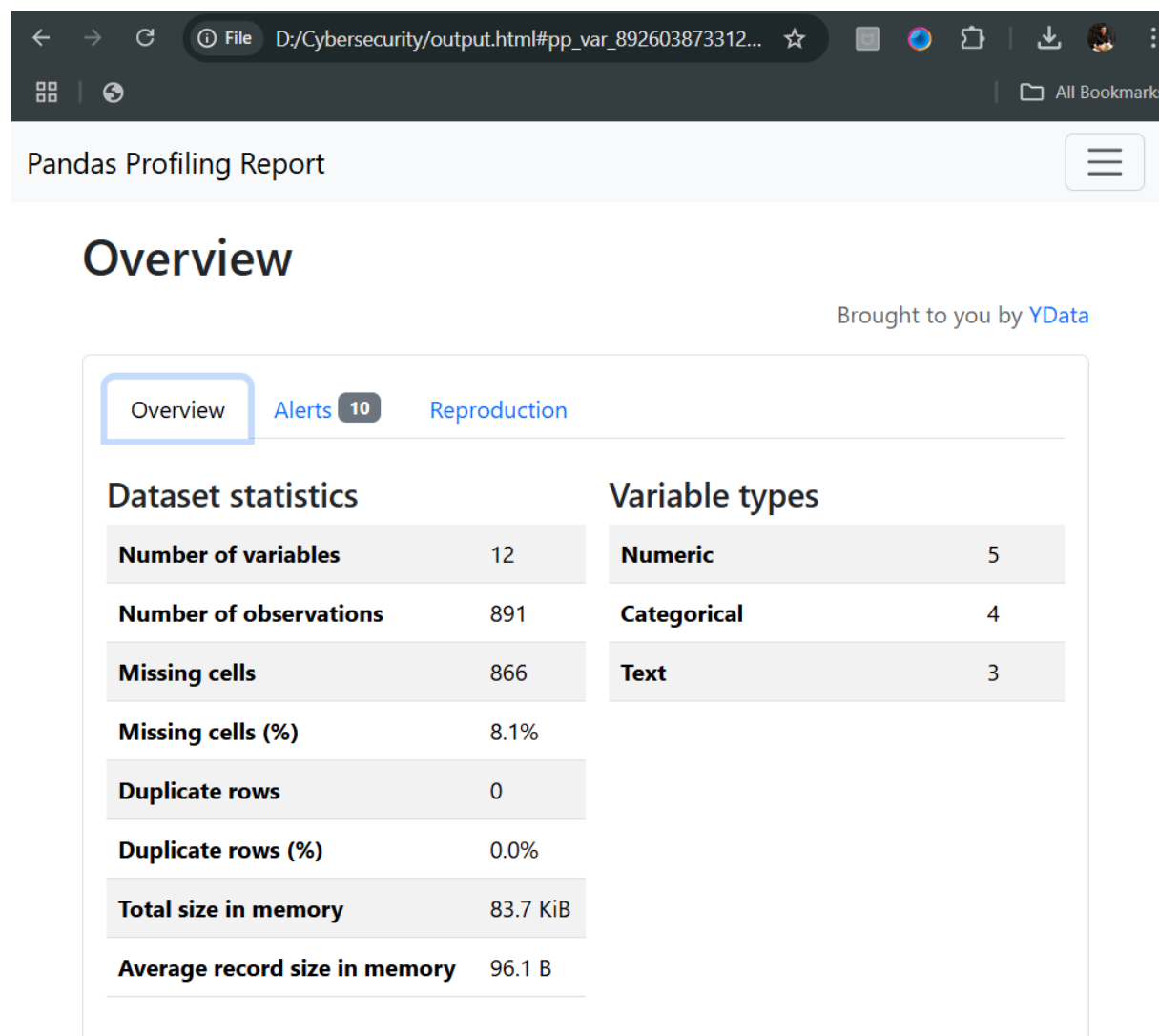
Pandas Profiling:

Pandas Profiling is a Python library that generates detailed reports for exploratory data analysis (EDA). It automatically computes various statistics, visualizations, and correlations of the dataset, helping users understand the data distribution, missing values, outliers, and relationships between features. It simplifies the process of initial data analysis.

Code link:

<https://colab.research.google.com/drive/1g-KvVcSVfrkLhZn4XpTpW8BfXJm59nL6?usp=sharing>

Output:



Pandas Profiling Report

Overview

Brought to you by YData

Overview Alerts 10 Reproduction

Dataset statistics		Variable types	
Number of variables	12	Numeric	5
Number of observations	891	Categorical	4
Missing cells	866	Text	3
Missing cells (%)	8.1%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	83.7 KiB		
Average record size in memory	96.1 B		

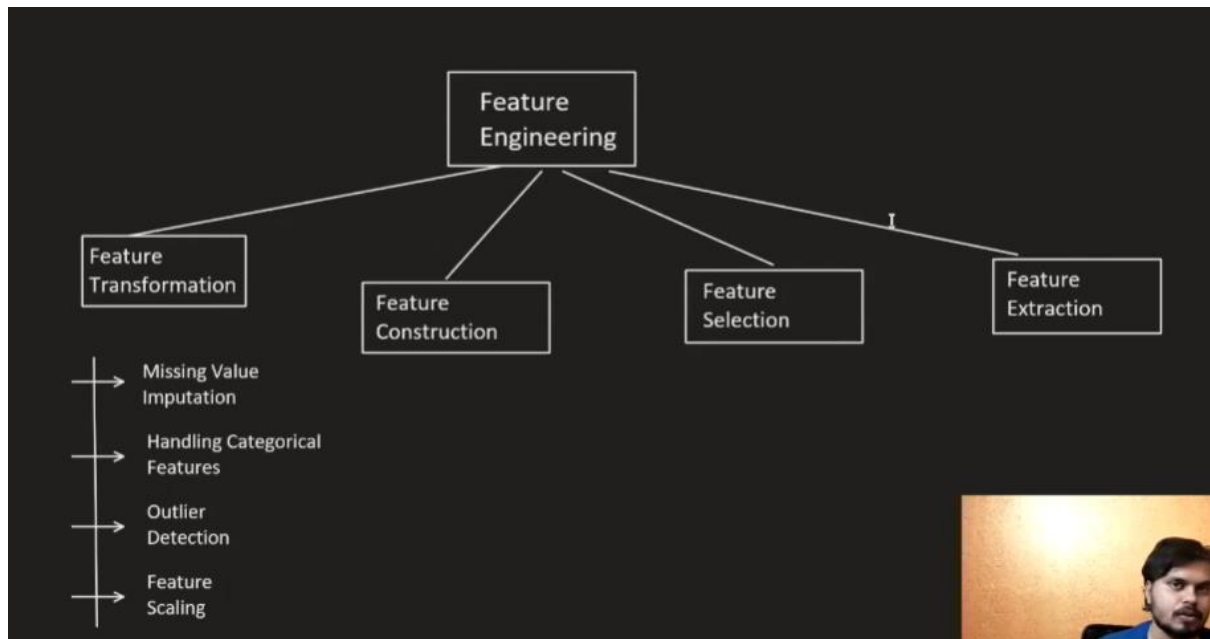
Variables

Video 23:

What is Feature Engineering:

Feature engineering is the process of using domain knowledge to extract features from raw data. These features can be used to improve the performance of ML algorithms.

Types:



Now, let's see how to do it?

1 **Feature Transformation:** The process of modifying or scaling input features to improve model performance, such as normalization or logarithmic transformations.

1.1 **Missing Value Implementation:** Techniques used to handle missing data, such as imputing with mean, median, mode, or using algorithms that handle missing values directly.

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	NaN	0
7	Berlin	30	1
8	Berlin	NaN	0
9	Berlin	25	1
10	Madrid	25	1

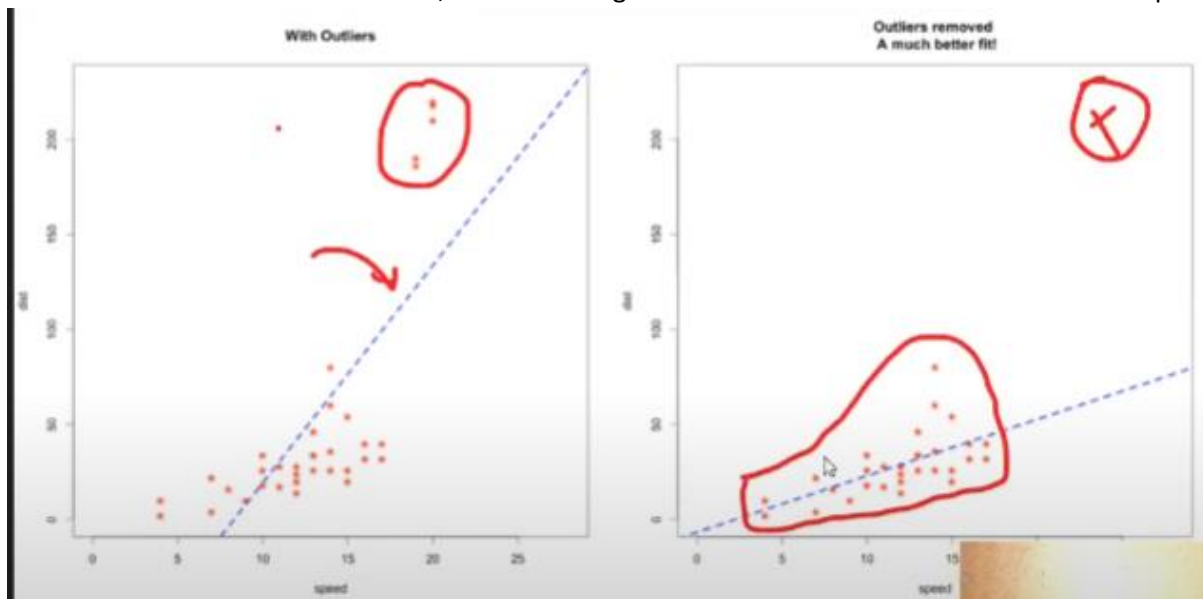


ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	26	0
7	Berlin	30	1
8	Berlin	26	0
9	Berlin	25	1
10	Madrid	25	1

1.2 **Handling Categorical Values:** Converting non-numeric categorical data into numerical values using methods like one-hot encoding, label encoding, or target encoding.

Index	Animal	One-Hot code →	Index	Dog	Cat	Sheep	Lion	Horse
0	Dog		0	1	0	0	0	0
1	Cat		1	0	1	0	0	0
2	Sheep		2	0	0	1	0	0
3	Horse		3	0	0	0	0	1
4	Lion		4	0	0	0	1	0

1.3 Outlier Detection: The process of identifying and handling data points that deviate significantly from the rest of the dataset, often using statistical methods or visual techniques.



1.4 Feature Scaling: The technique of standardizing or normalizing features to ensure they have similar ranges, helping algorithms converge faster and perform better, especially in models sensitive to feature magnitudes.

	A	B	C	D
1	Country	Age	Salary	Purchased
2	France	44	72000	No
3	Spain	27	48000	Yes
4	Germany	30	54000	No
5	Spain	38	61000	No
6	Germany	40		Yes
7	France	35	58000	Yes
8	Spain		52000	No
9	France	48	79000	Yes
10	Germany	50	83000	No
11	France	37	67000	Yes

2. Feature Construction: The process of creating new features from existing ones to enhance model performance, often by combining, transforming, or aggregating data in ways that provide more predictive power.

3. Feature Selection: The process of selecting a subset of relevant features from the original set to improve model performance, reduce overfitting, and decrease computational cost by eliminating redundant or irrelevant features.

4. Feature Extraction: The process of transforming raw data into a set of meaningful features that better represent the underlying patterns, often through techniques like principal component analysis (PCA) or domain-specific methods.

Video 24:

Feature Scaling – Standardization:

What is Feature scaling?

It is a technique to standardize the independent features present in the data in a fixed range.

Why do we need Feature Scaling?

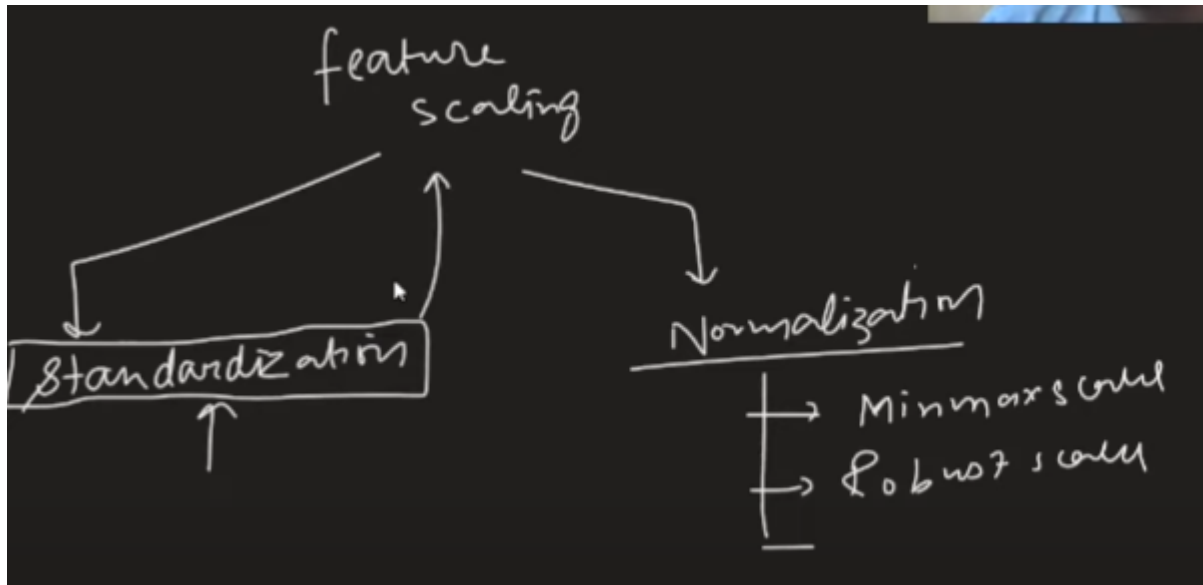
Feature scaling is important because many machine learning algorithms rely on the assumption that features are on similar scales. Without scaling, features with larger ranges can dominate the model, leading to biased or inefficient learning. Here's why it's needed:

- **Improved Model Performance:** Algorithms like gradient descent converge faster when features are scaled, as it helps avoid uneven updates during training.
- **Prevents Dominance of Certain Features:** Features with larger values can disproportionately affect the model, skewing the results.

- **Ensures Fair Treatment of All Features:** Some algorithms (like KNN, SVM, and logistic regression) calculate distances between data points. Scaling ensures that all features contribute equally to distance metrics.

It helps in achieving better accuracy and faster convergence, especially with algorithms sensitive to feature magnitudes!

Types of feature scaling?



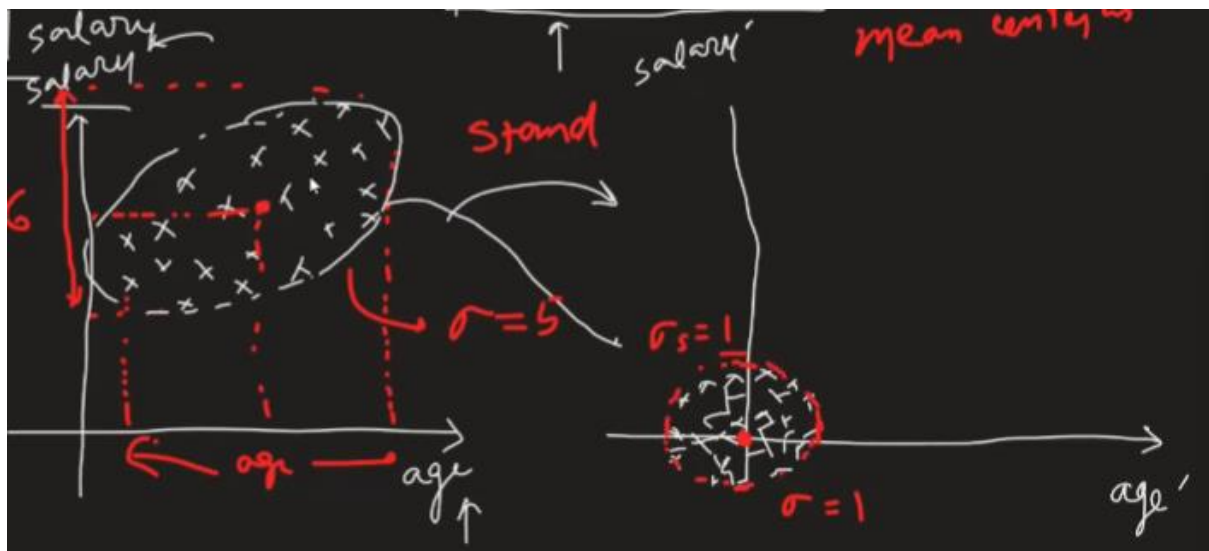
Standardization (Z-score Scaling):

- Centers the data around 0 with a standard deviation of 1, by subtracting the mean and dividing by the standard deviation.
- Formula:

$$Z = \frac{x - \mu}{\sigma}$$

Score
Mean
SD

- Useful when data has varying units and needs to be compared on the same scale.
- The obtained series will have mean = 0 and std = 1



Example:

Code link:

https://colab.research.google.com/drive/1Ni3fWzpjP_310K6mc4WpQGGSoXbT_sla?usp=sharing

Data link:

<https://github.com/campusx-official/100-days-of-machine-learning/tree/main/day24-standardization>

When to use Standardization?

Algorithm(s)	Reason of applying feature scaling
1. <u>K-Means</u>	Use the Euclidean distance measure.
2. <u>K-Nearest-Neighbours</u>	Measure the distances between pairs of samples and these distances are influenced by the measurement units
3. <u>Principal Component Analysis (PCA)</u>	Try to get the feature with maximum variance
4. Artificial Neural Network	Apply Gradient Descent
5. Gradient Descent	Theta calculation becomes faster after feature scaling and the learning rate in the update equation of Stochastic Gradient Descent is the same for every parameter

Video 25:

Feature Scaling –Normalization:

What is Normalization?

It is a technique often applied as part of data preparation for ML. The goal of it is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the range of values or losing information.

Types of normalization:

1. Min max scaling
2. Mean Normalization
3. Max absolute
4. Robust scaling

Min max scaling:

Formulae:

$$X'_i = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$
$$= \frac{130 - 32}{130 - 32} = 1$$

Output will always be between 0 and 1.

Visualisation: compressing the data to a dimension where the unit of it be 1.



Example:

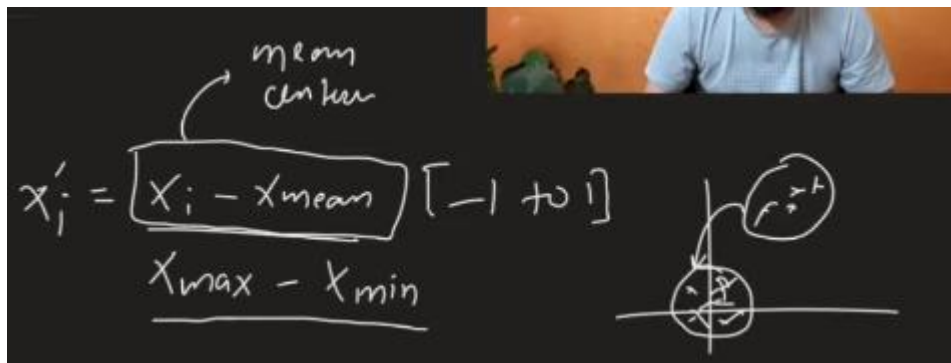
Code link:

<https://colab.research.google.com/drive/1DP3FxFdSx979EqEgrOrdVJhAUelccNE9?usp=sharing>

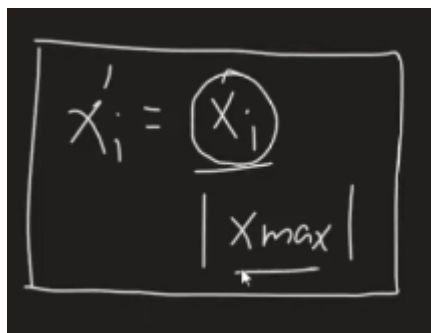
Data link:

<https://github.com/campusx-official/100-days-of-machine-learning/tree/main/day25-normalization>

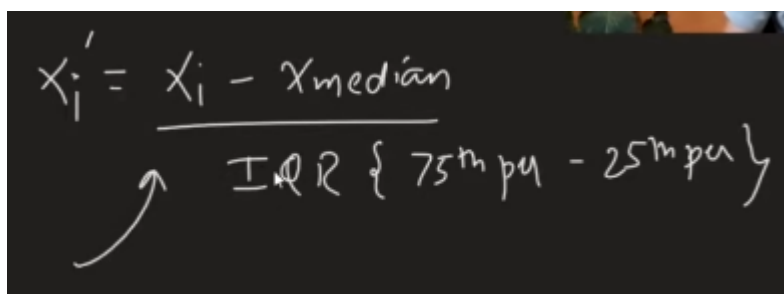
Mean Normalization: (rarely used- for centred data)


$$X'_i = \frac{X_i - x_{\text{mean}}}{x_{\text{max}} - x_{\text{min}}} [-1 \text{ to } 1]$$

Max Abs Scaling: use for sparse data (for data where 0 is frequent)


$$X'_i = \frac{X_i}{|x_{\text{max}}|}$$

Robust Scaling:


$$X'_i = \frac{X_i - x_{\text{median}}}{\text{IQR} \{ 75^{\text{th}} \text{ p} - 25^{\text{th}} \text{ p} \}}$$

Normalization vs Standardization:

1. Is feature scaling required?
2. Min max is used in normalization