DATA TRANSFORMATIONS

Data Science Nights, Session-2

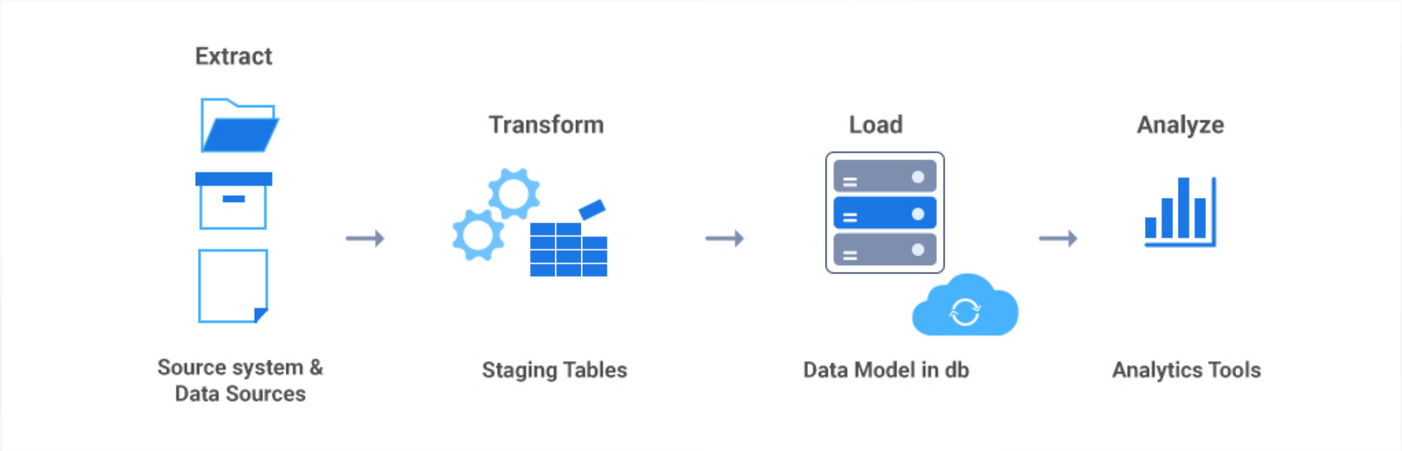
VIT BHOPAL

WHAT (AND WHY) IS
DATA TRANSFORMATION?

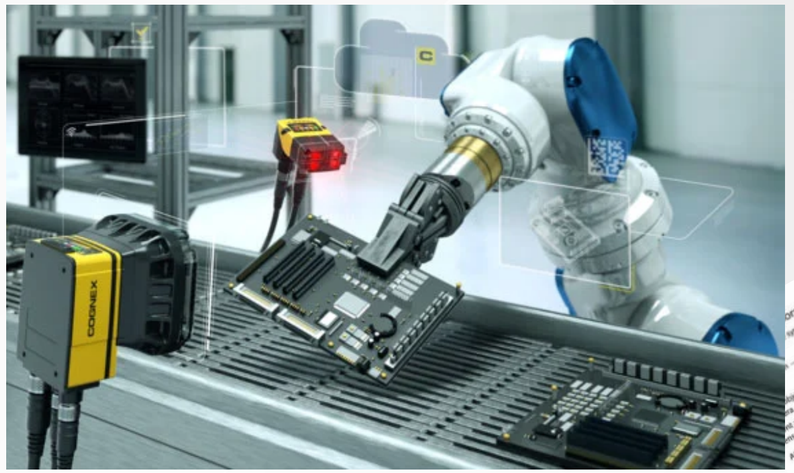
Manipulating and Transferring Data into a form deemed usable for data analysis and modelling.

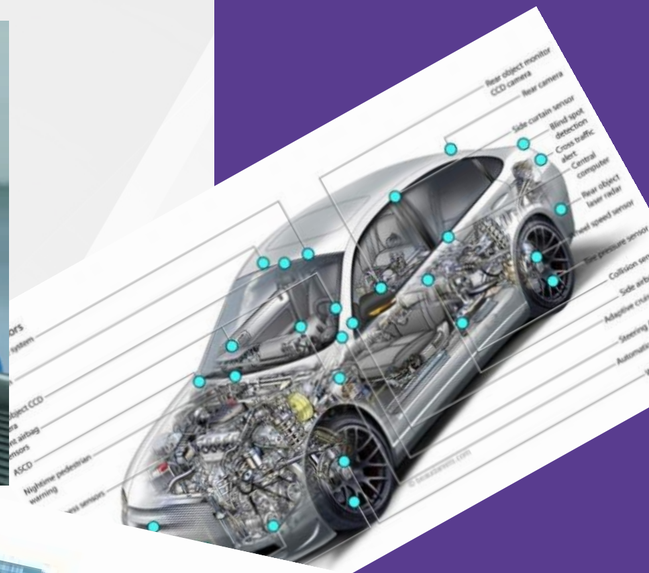
TRANSFORMATION IN THE DATA PIPELINE

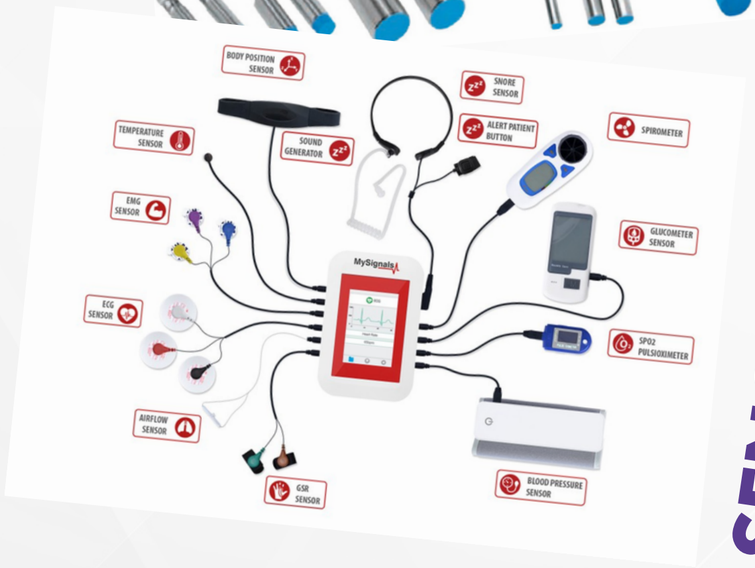
A typical organisation with traditional data storage methods use an extract, transform, load, with the data transformation taking place during the middle ‘transform’ step.



OUR
PROBLEM STATEMENT







SENSORS!!

PREDICTING
SENSOR FAILURES

The dataset consists of

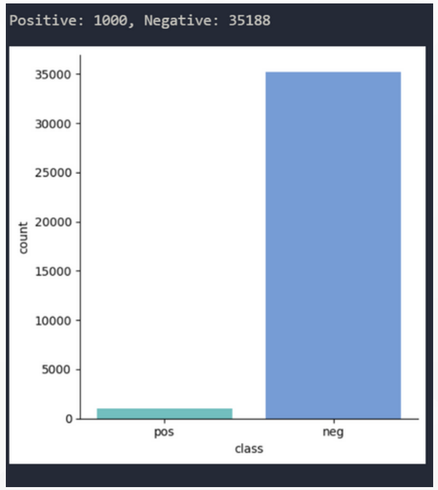
- Sensor Data
- Corresponding label regarding the data (whether the sensor data is correct or not)

```
We have 170 numerical features : ['aa_000', 'ab_000', 'ac_000',

We have 1 categorical features : ['class']
```

We also have

- A Grossly imbalanced dataset
- An immense amount of null values
- A bunch of seemingly nonsensical number data

```
df.isnull().sum()

class         0
aa_000        0
ab_000    27896
ac_000     2141
ad_000     9200
          ...
ee_007      379
ee_008      379
ee_009      379
ef_000     1730
eg_000     1729
Length: 171, dtype: int64
```




```
df= pd.read_csv("sensor_failure.csv", na_values="na")
df
```

	class	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001
0	pos	153204	0.0	1.820000e+02	NaN	0.0	0.0	0.0	0.0
1	pos	453236	NaN	2.926000e+03	NaN	0.0	0.0	0.0	0.0
2	pos	72504	NaN	1.594000e+03	1052.0	0.0	0.0	0.0	244.0
3	pos	762958	NaN	NaN	NaN	NaN	NaN	776.0	281128.0
4	pos	695994	NaN	NaN	NaN	NaN	NaN	0.0	0.0
...	...	...	...	...	...	...	...	...	...
36183	neg	153002	NaN	6.640000e+02	186.0	0.0	0.0	0.0	0.0
36184	neg	2286	NaN	2.130707e+09	224.0	0.0	0.0	0.0	0.0
36185	neg	112	0.0	2.130706e+09	18.0	0.0	0.0	0.0	0.0
36186	neg	80292	NaN	2.130706e+09	494.0	0.0	0.0	0.0	0.0
36187	neg	40222	NaN	6.980000e+02	628.0	0.0	0.0	0.0	0.0

36188 rows × 171 columns

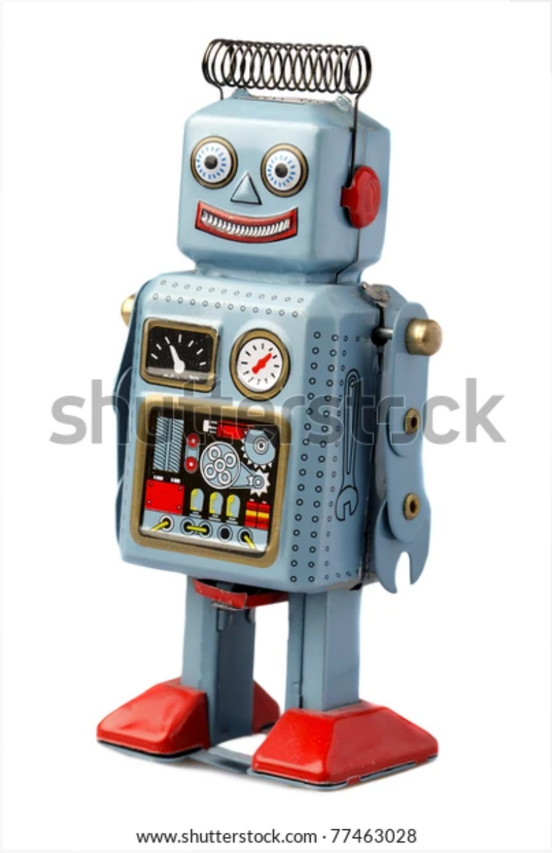
TRANSFORMATION TO THE RESCUE!

Things to do with the dataset

- Clean the data
- Identify the Outliers
- Remove the bias in data
- Scale the data in a computationally friendly manner
- Somehow make sense of this nonsense 🙃

Technical Steps of Data Transformation

- Data Cleaning
- Data Normalization
- Data Aggregation
- Data Reduction



DATA IMPUTATION

Process of extrapolating values of missing data from existing data.

DATA IMPUTATION

Why should we refrain from dropping these samples from the dataset?

SIMPLE IMPUTER

Univariate Imputation Strategy i.e estimate missing values by only utilising data present in a single feature.

Commonly used method for handling MCAR data.

SIMPLE IMPUTER

1) Mean $$\overline{X} = \frac{\sum X}{N}$$

2) Median $$\text{Med}(X) = \begin{cases} X[\frac{n+1}{2}] & \text{if n is odd} \\ \frac{X[\frac{n}{2}] + X[\frac{n}{2}+1]}{2} & \text{if n is even} \end{cases}$$

3) Mode $$M_o = l + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2}\right) h$$

MISSING DATA
ASSUMPTIONS

Imagine a dataset containing data from *student surveys* about their **study habits** and **stress levels**.

MISSING DATA
ASSUMPTIONS

1) MCAR: Data is missing due to a completely random reason. (**M**issing **C**ompletely **A**t **R**andom)

- *Misplaced questionnaires*
- *Accidental mixup*
- *Software glitch*

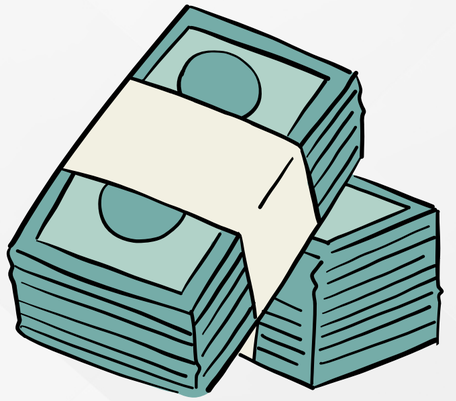
2) MAR: Probability of missing data is tied to another feature or variable. (**M**issing **A**t **R**andom)

- *Students with **more** extracurriculars, tend not to fill up the entire form*
- *Scheduling of survey **during school hours** may impact proper survey conduction*

MISSING DATA
ASSUMPTIONS

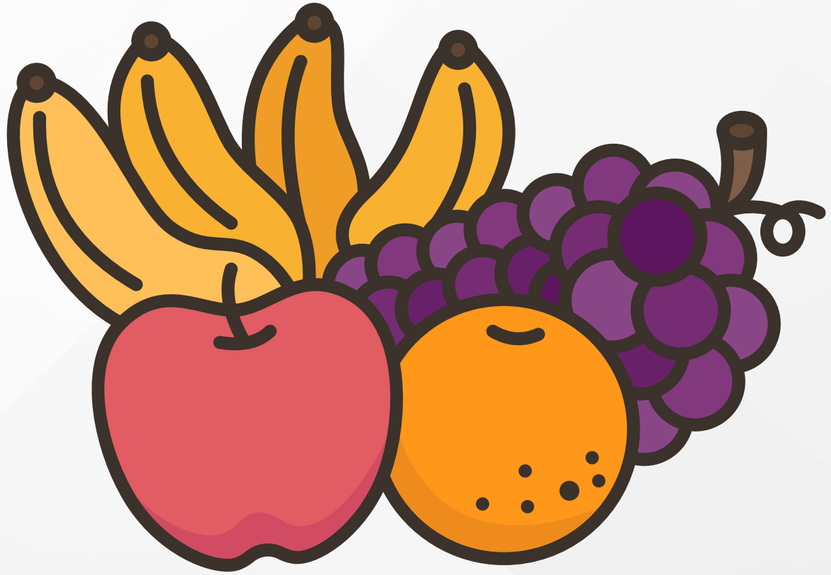
3) MNAR: Missing data related to true value of the field (**M**issing **N**ot **A**t **R**andom)

- *Students struggling with **mental health issues** or **stress** likely to skip certain questions*
- ***Financial incentives** may alter the participants' answers*



MISSING DATA
ASSUMPTIONS

Think about how MAR may occur in this scenario.



KNN IMPUTER

Multivariate imputer (as opposed to simple imputer). Uses data from multiple features to estimate missing data values

We only choose ***n*** neighbors (which are the most similar to) the missing data sample to find the missing value.

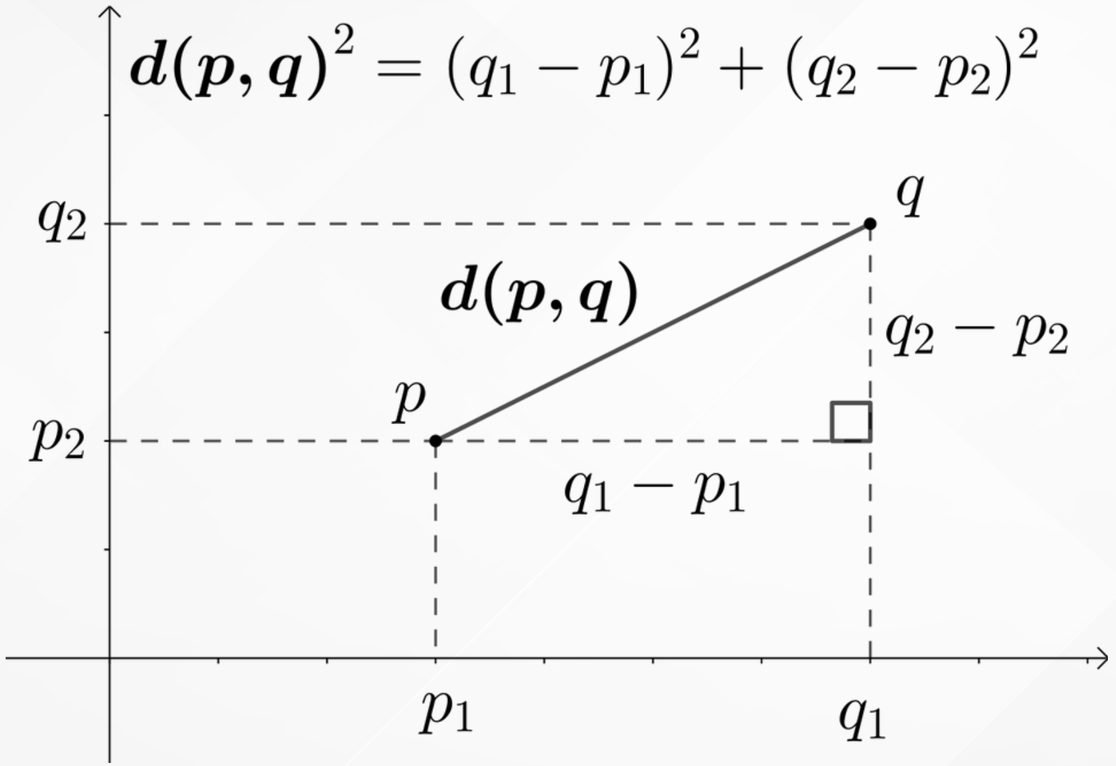
KNN
IMPUTER

Example

	SibSp	Fare	Age	
0	1	7.2500	22.0	
1	1	71.2833	38.0	
2	0	7.9250	26.0	← Neighbor 1
3	1	53.1000	35.0	
4	0	8.0500	35.0	← Neighbor 2
5	0	8.4583	NaN	

KNN
IMPUTER

$$\boldsymbol{d(p,q)^2} = (q_1 - p_1)^2 + (q_2 - p_2)^2$$



UNDERSTANDING
KNN IMPUTER METRIC

Euclidean Distance

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

NaN Euclidean Distance

$$\text{weight} \times \text{Euclidean distance}$$

$$\text{weight} = \sqrt{\frac{\text{total number of features}}{\text{number of non-null value features}}}$$

UNDERSTANDING
KNN IMPUTER METRIC

	SibSp	Fare	Age
0	1	7.2500	22.0
1	1	71.2833	38.0
2	0	7.9250	26.0
3	1	53.1000	35.0
4	0	8.0500	35.0
5	0	8.4583	NaN

p1 = (0, 8.4583, NaN)
p2 = (0, 7.925, 26)

weight = sqrt(3 / 2) = 1.2247
NaN Euclidean = 0.6531

---

p1 = (0, 8.4583, NaN)
p2 = (1, 71.2833, 38)

weight = sqrt(3 / 2) = 1.2247
NaN Euclidean = 76.95

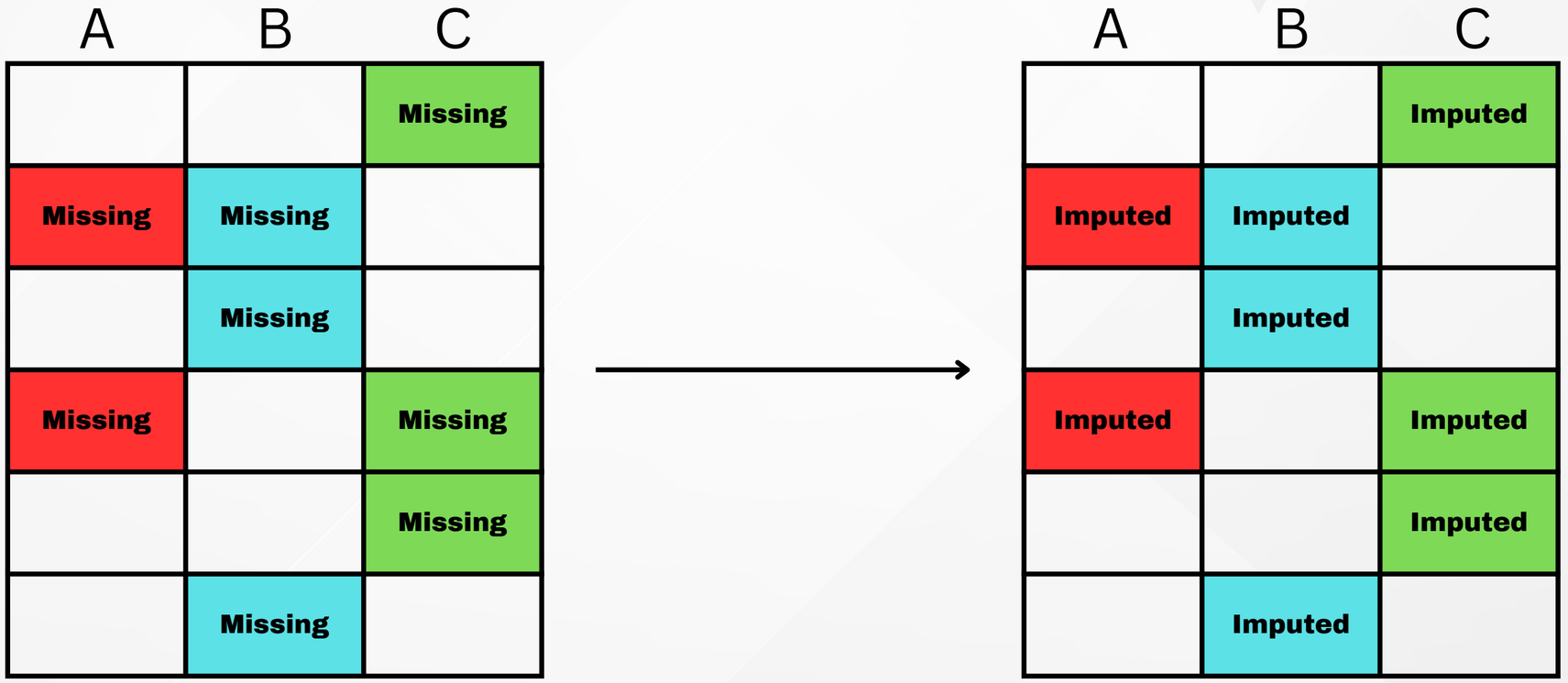
MICE
ALGORITHM

Multiple **I**mputation by **C**hained **E**quation
An Iterative approach to imputing data.

MICE

ALGORITHM STEPS

1) Impute missing data as with column mean / median / mode while marking them as placeholders

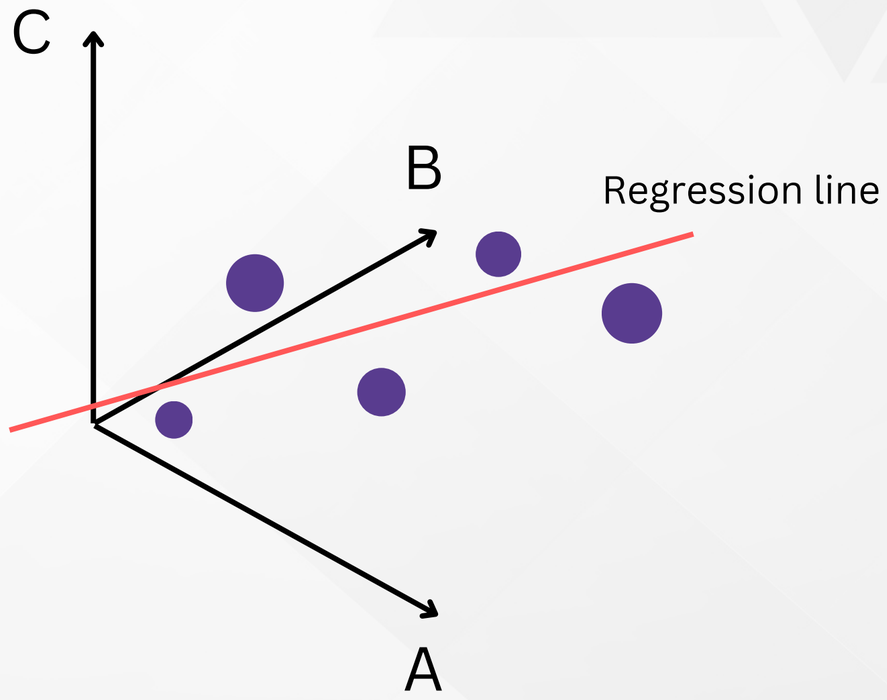


MICE

ALGORITHM STEPS

2) Drop placeholders in one of the columns and regress that column with the other ones

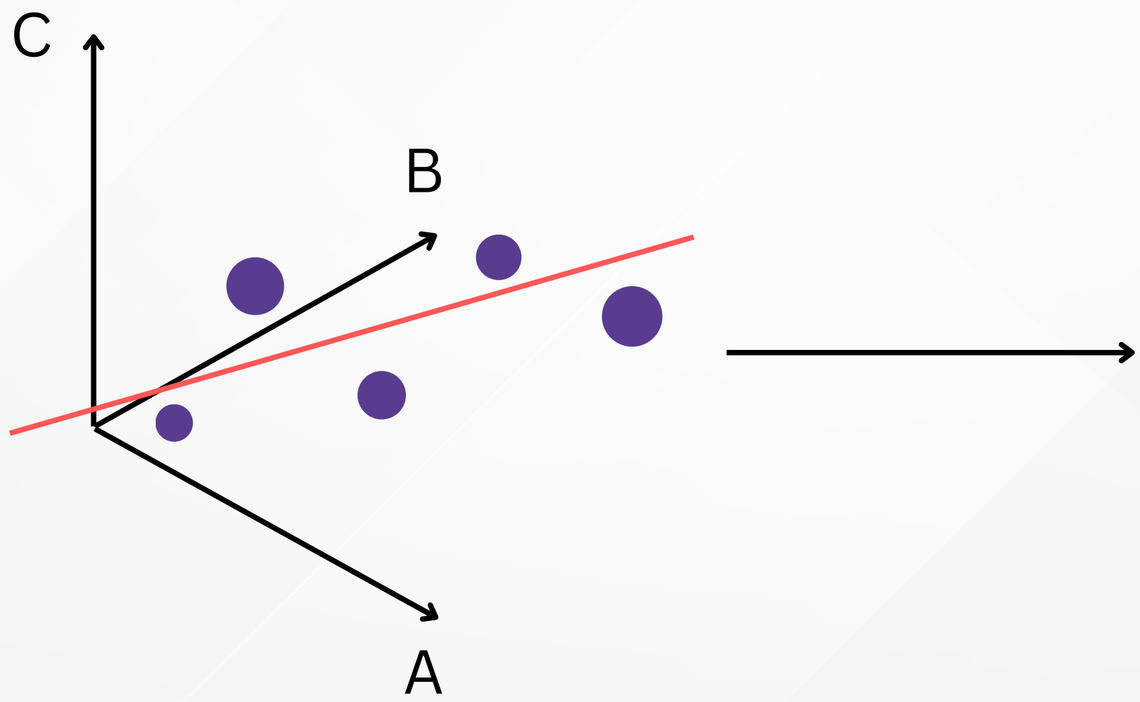
A	B	C
		Imputed
	Imputed	
	Imputed	
		Imputed
		Imputed
	Imputed	



MICE

ALGORITHM STEPS

3) Use regression model created to “predict” values for place holder values in the current column

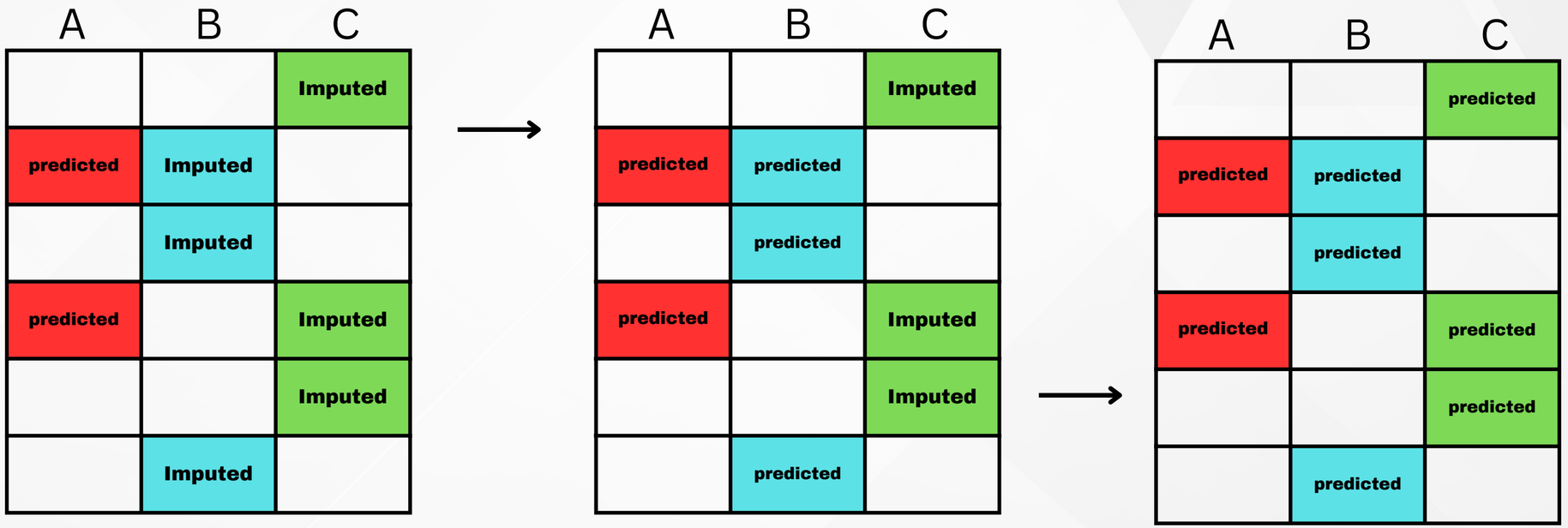


A	B	C
		Imputed
predicted	Imputed	
	Imputed	
predicted		Imputed
		Imputed
	Imputed	

MICE

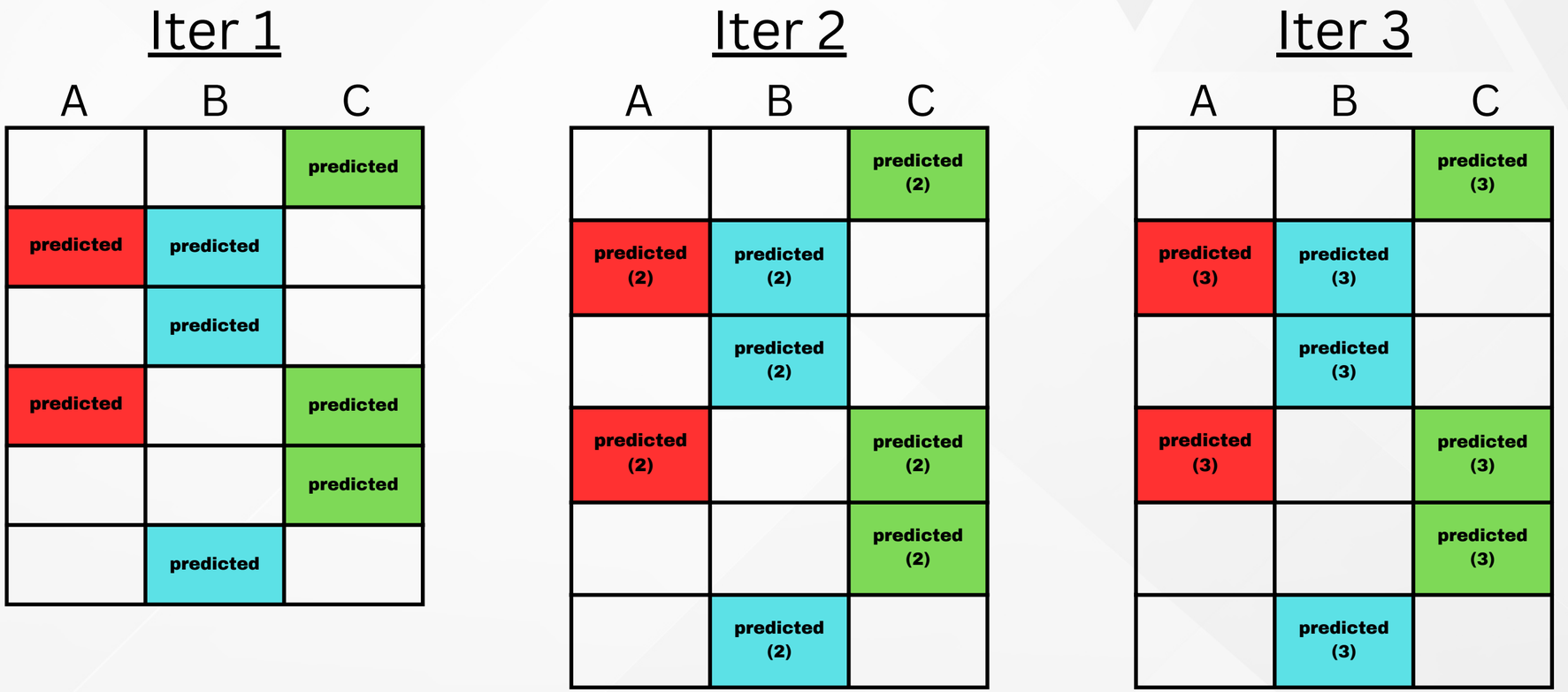
ALGORITHM STEPS

4) Repeat this process for the other columns



MICE
ALGORITHM STEPS

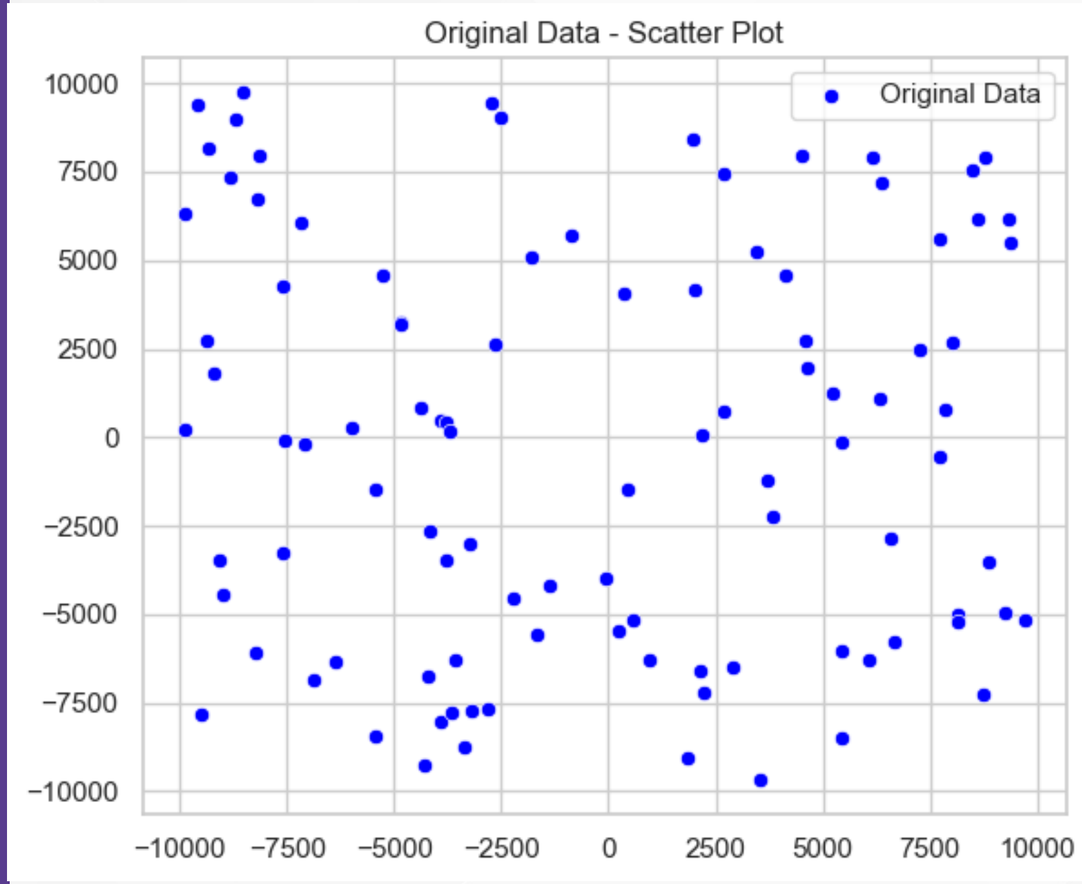
5) Do the previous steps as specified by the hyperparameter.

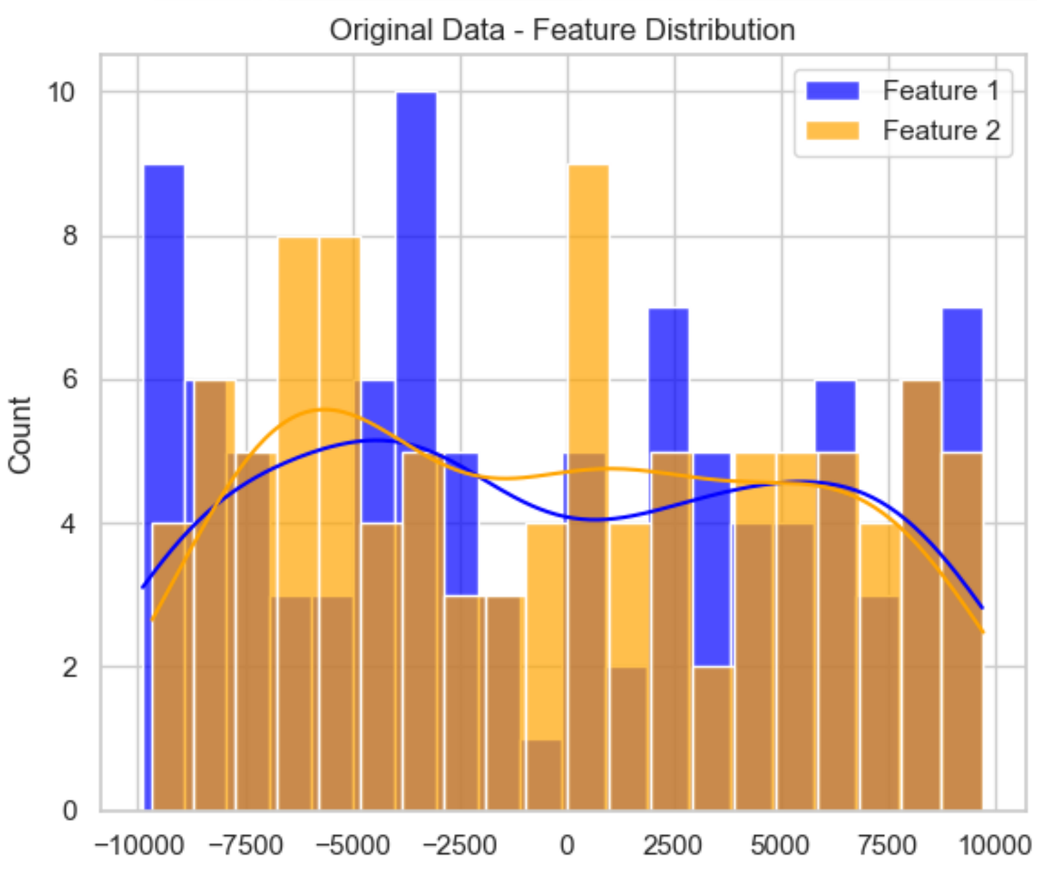


FEATURE SCALING

Feature scaling is a technique used in machine learning to transform input data into a range that is suitable for training and testing models.

WHY ?


Original Data - Scatter Plot
Original Data
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000


Original Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000

STANDARD SCALER

In Machine Learning, Standard Scaler is used to resize the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1.

$$Z = \frac{(X - U)}{S}$$

Z-is the scaled value

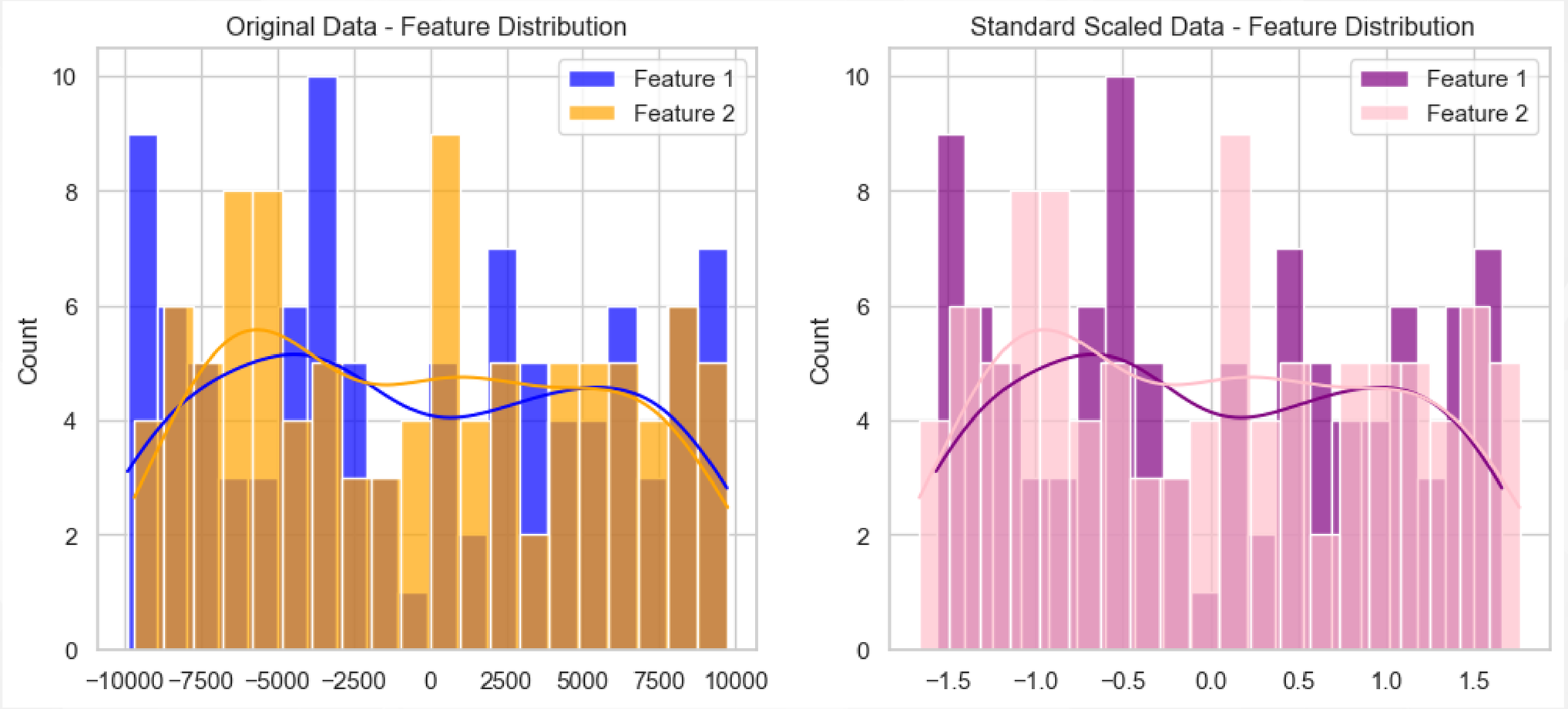
U-is the mean of the column

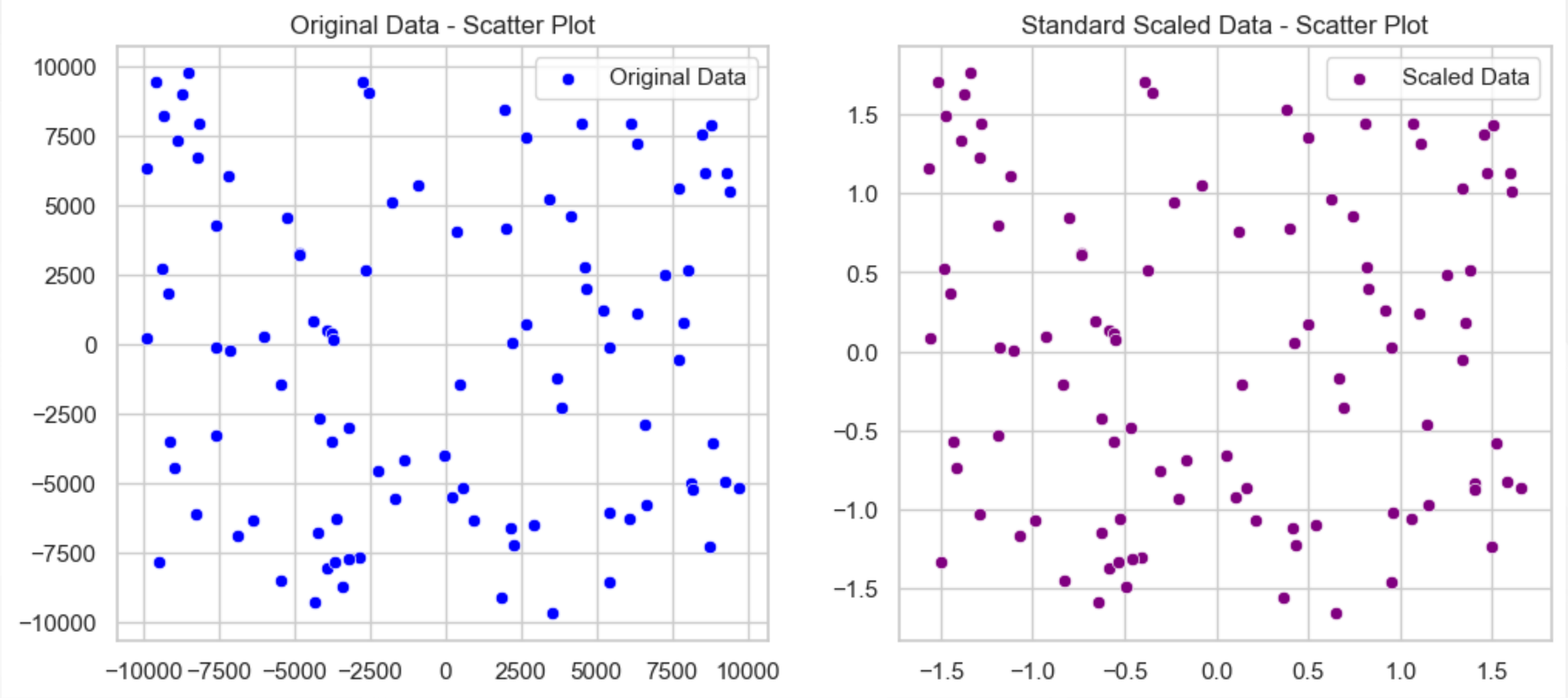
S -is the standard deviation of the column

Mean is nothing but the average of the given set of values.

Standard deviation is a statistic that measures the
distribution or scattering of a dataset relative to its mean.

(Standard Deviation) $$S = \sqrt{\sum_{i=1}^{n} \frac{(Xi - X)^2}{N}}$$


Original Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000
Standard Scaled Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Original Data - Scatter Plot
Original Data
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
−10000 −7500 −5000 −2500 0 2500 5000 7500 10000
Standard Scaled Data - Scatter Plot
Scaled Data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5

MINMAX SCALER

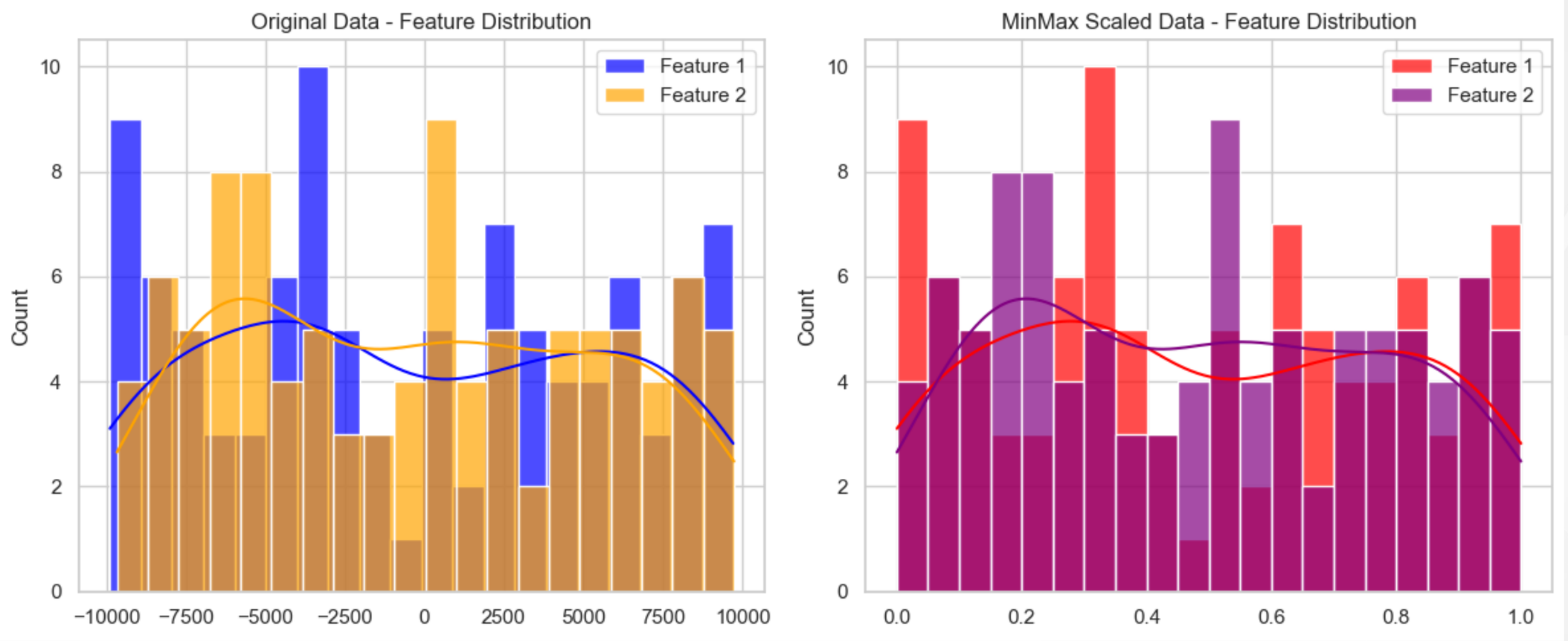
In machine learning, the Min-Max Scaler is used to scale and transform features in a way that the values fall within a specified range, typically [0, 1].

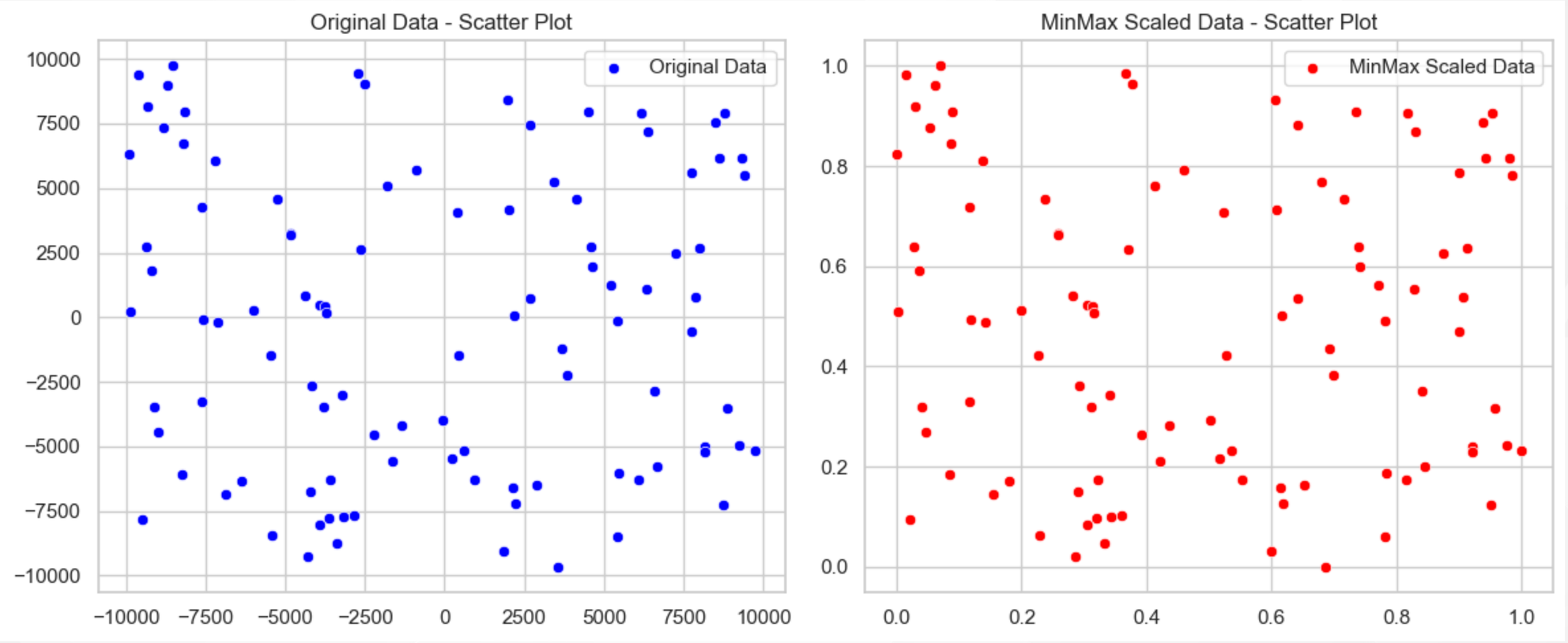
$$M = \frac{(X - \min)}{(\max - \min)}$$

M - The Scaled Value

min - The minimum Value

max - The maximum Value


Original Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000
MinMax Scaled Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
0.0
0.2
0.4
0.6
0.8
1.0


Original Data - Scatter Plot
Original Data
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000
MinMax Scaled Data - Scatter Plot
MinMax Scaled Data
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

ROBUST SCALER

In machine learning, Robust Scaler is particularly useful when dealing with datasets that contain outliers.
It scales features using interquartile range (IQR) and median which are robust to outliers, making it less sensitive to extreme values.

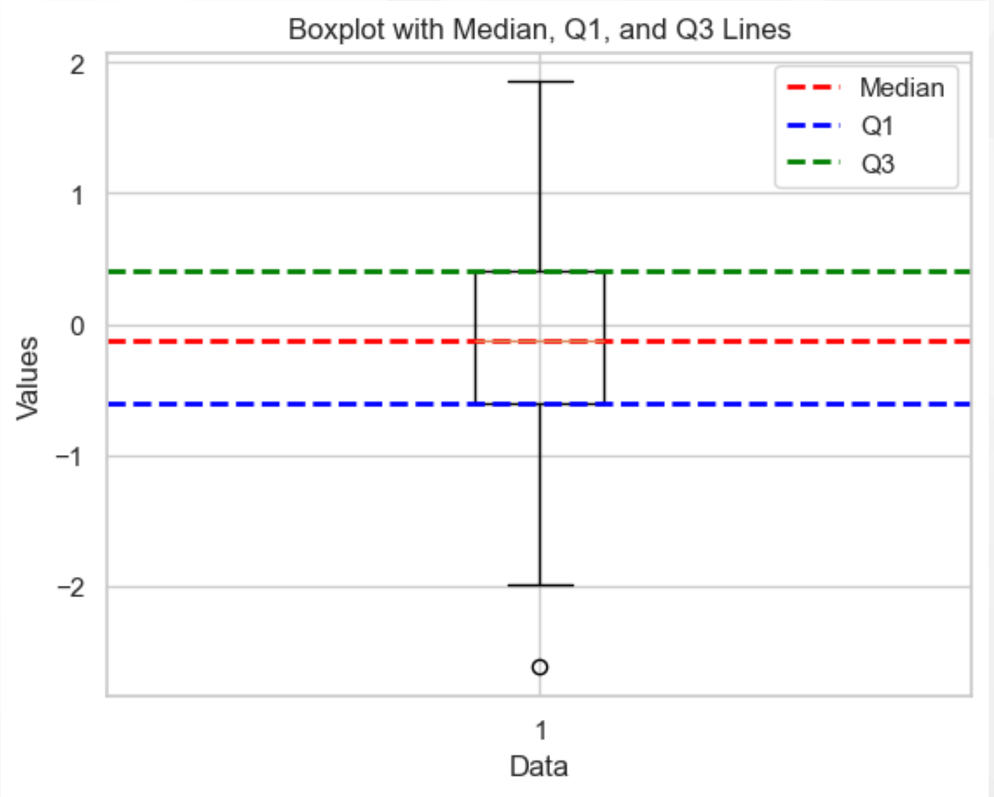
$$R = \frac{(X - \text{median})}{Q_3 - Q_1}$$

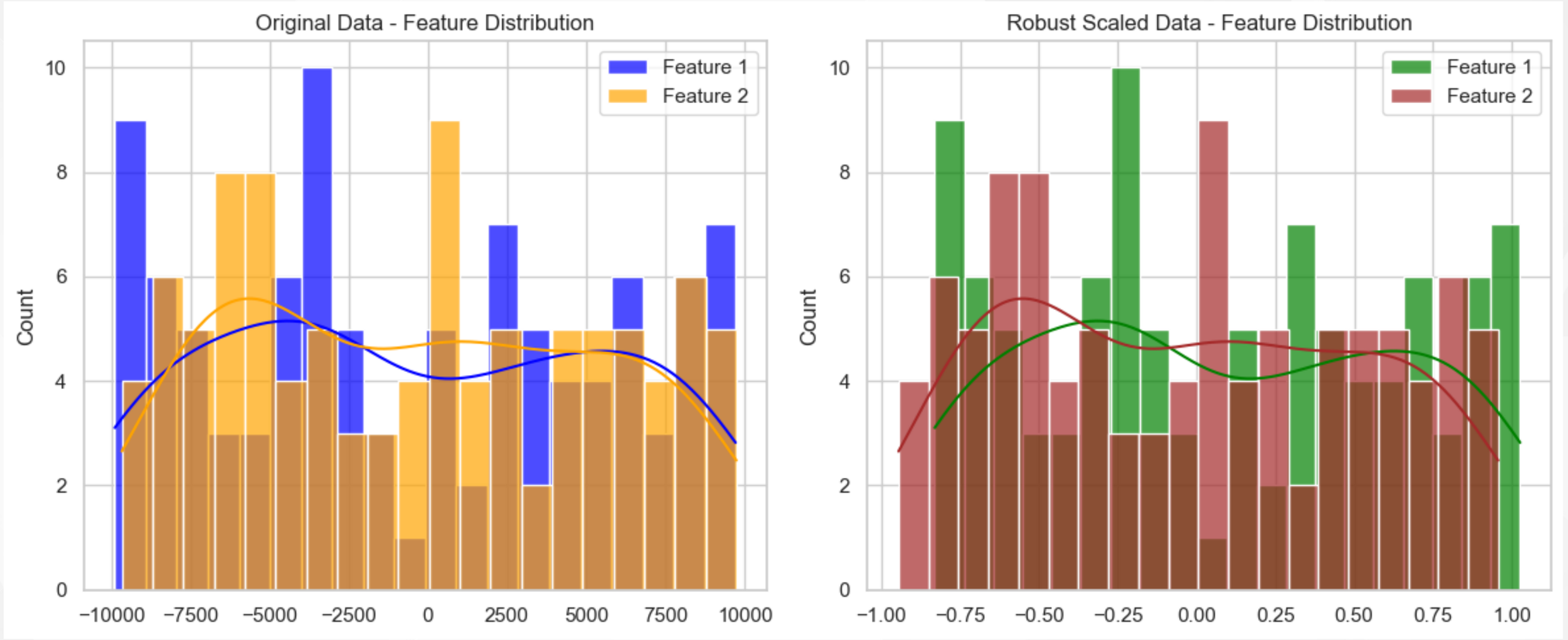
R - The Scaled Value

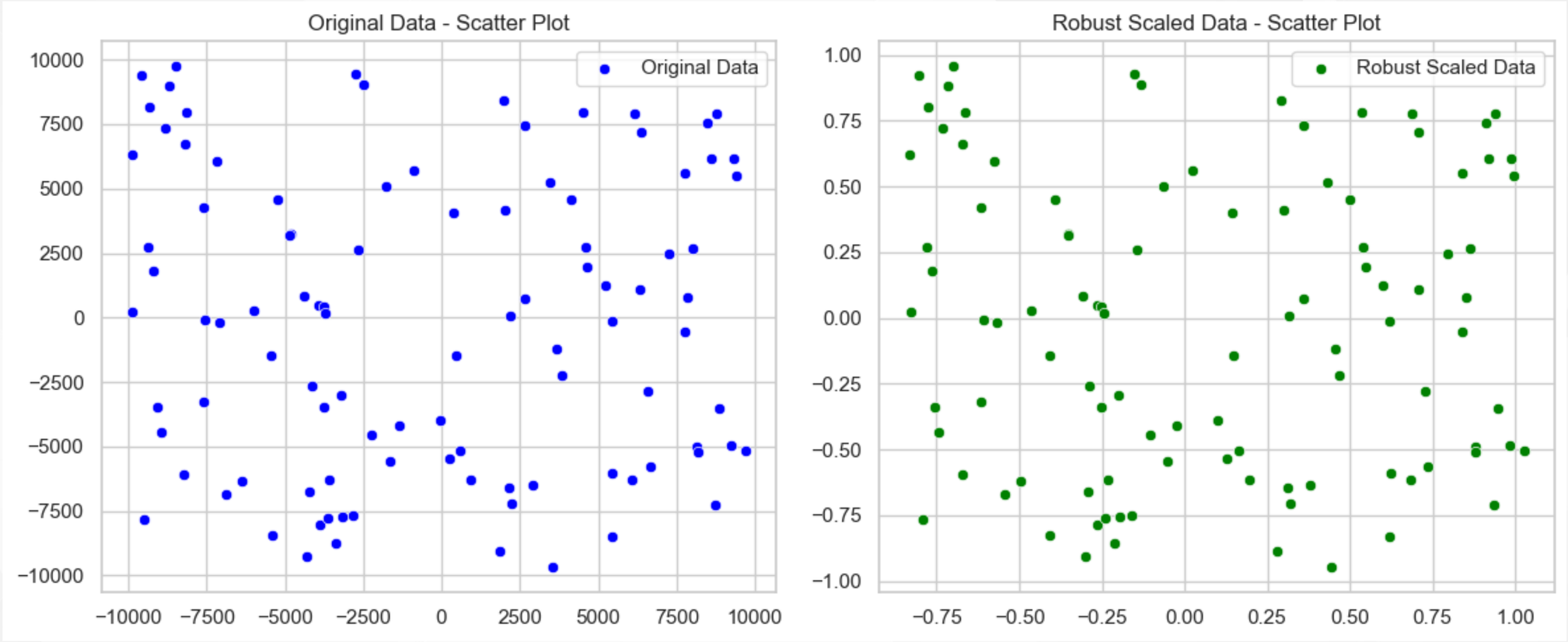
Q3 - 3rd Quartile Value(75%)

Q1 - 1st Quartile Value(25%)

The median is the middle value in a set of data. The interquartile range (IQR) is a measure of statistical dispersion, representing the range between the first quartile (Q1) and the third quartile (Q3) in a dataset.

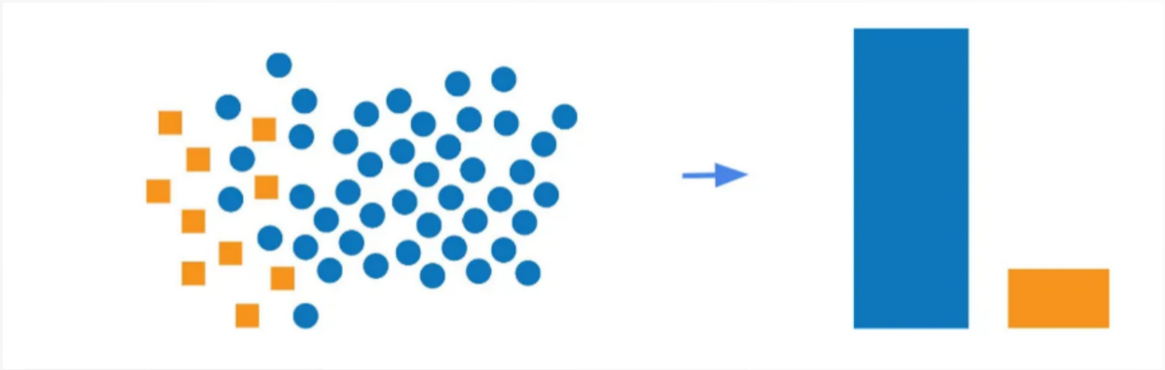



Original Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000
Robust Scaled Data - Feature Distribution
Feature 1
Feature 2
Count
10
8
6
4
2
0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


Original Data - Scatter Plot
Original Data
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
−10000 −7500 −5000 −2500 0 2500 5000 7500 10000
Robust Scaled Data - Scatter Plot
Robust Scaled Data
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

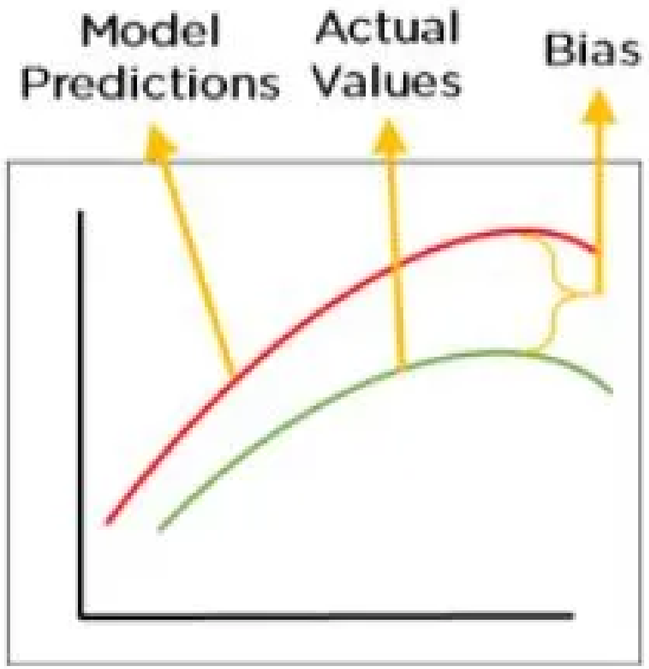
WHAT IS
IMBALANCED DATA?

Imbalaced Data refers to the conditions where the data is not uniform across the different classes.



HANDLING
IMBALANCED DATA

In an imbalanced datasets, one or more classes can be significantly underrepresented which might affect the trained model by increasing the bias for the majority class.



HANDLING
IMBALANCED DATA

There are various significant methods to handle the imbalanced data : -

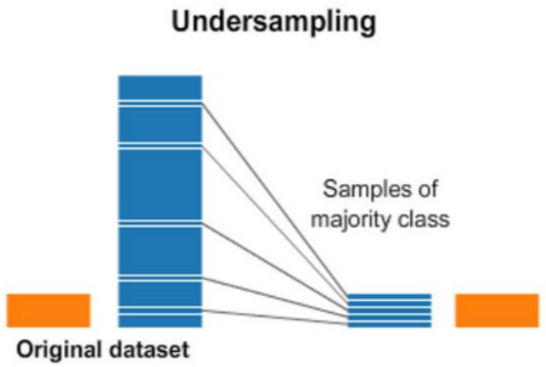
1. Resampling Techniques
2. Data Augmentation
3. Ensemble Techniques

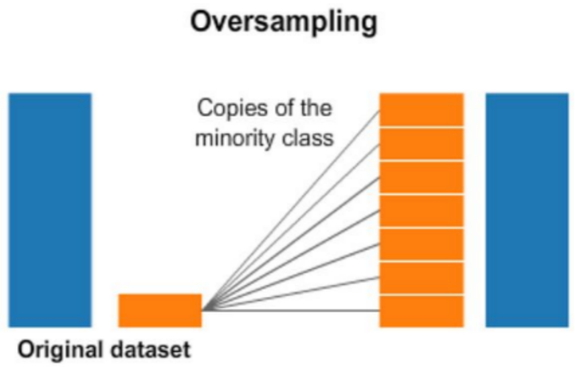
Resampling
Techniques

Resampling is a method for handling the imbalanced datasets, within Resampling , a sample from the pre-existing data or the population is to create the new samples to balance the existing data.

There are two well known techniques for the re-sampling of the data

1. Oversampling
2. Undersampling

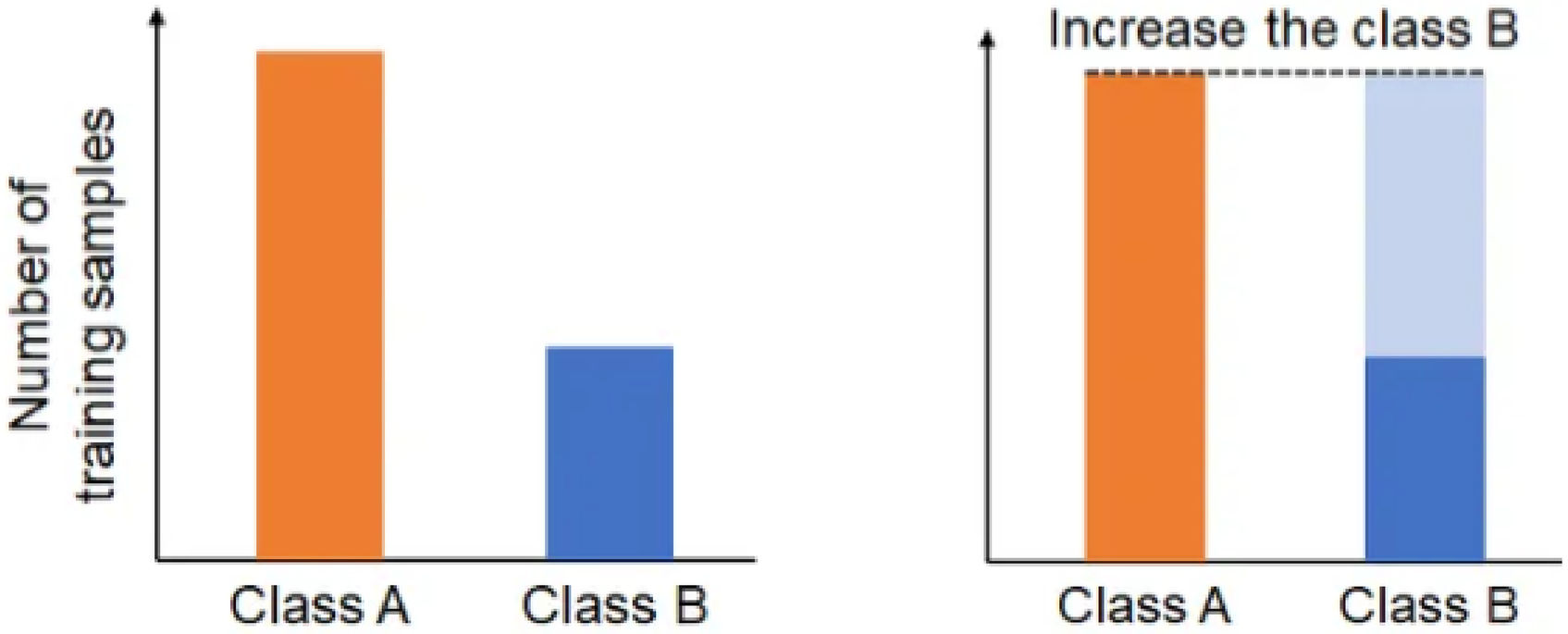




Oversampling

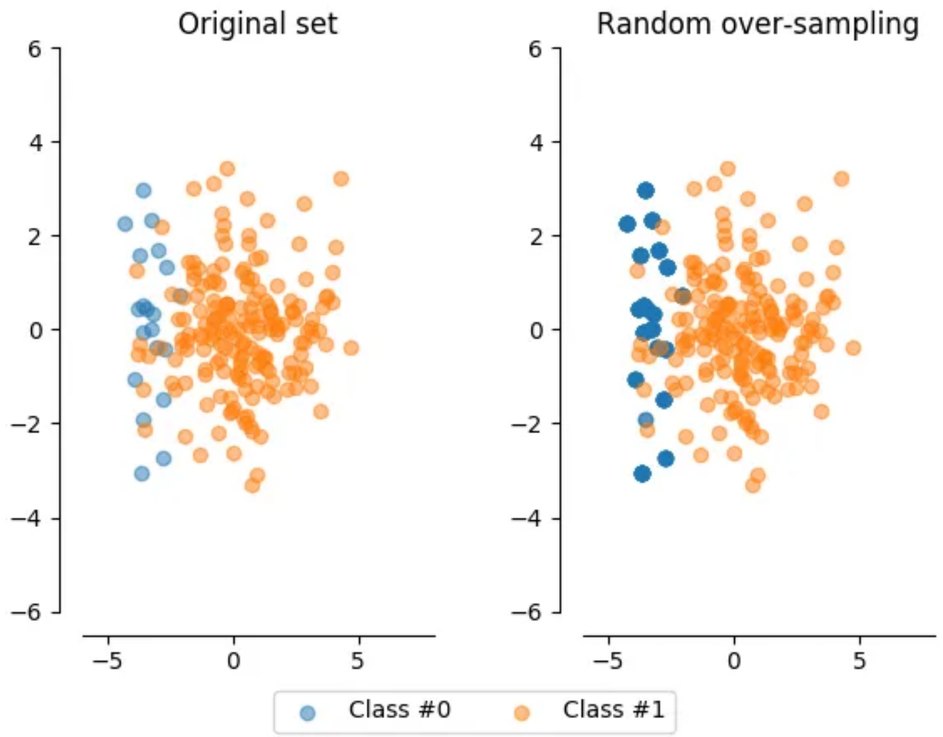
A Resampling Technique which involves duplicating the samples from the Minority class.
Some Popular Oversampling methods involves :-

- Random Oversampling
- SMOTE (Synthetic Minority Over-sampling Technique)
- SMOTEN (Synthetic Minority Over-sampling Technique for Nominal)



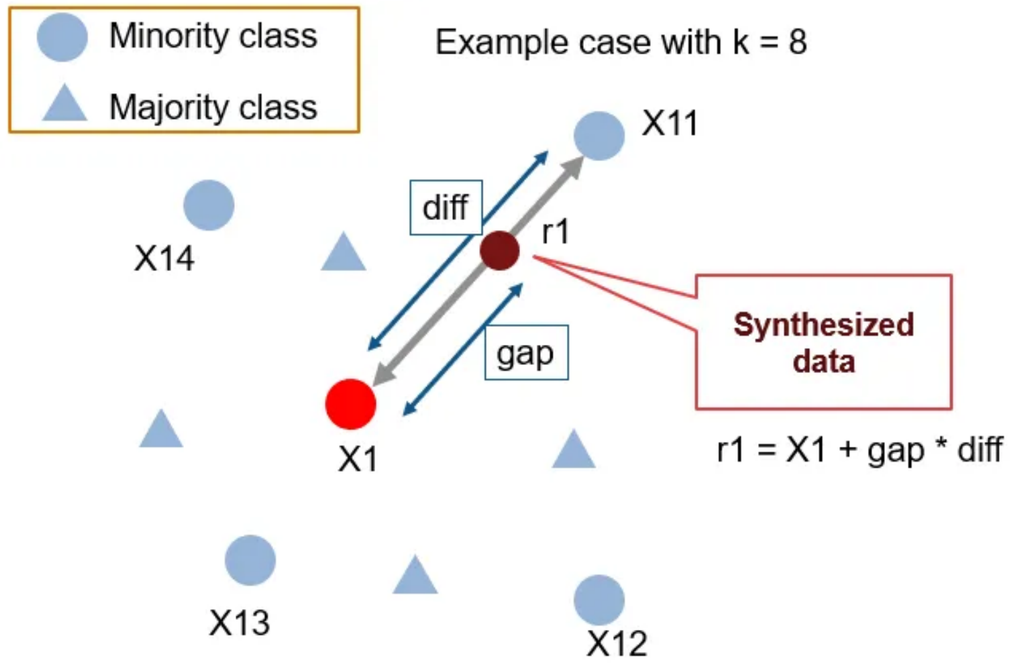
Random Oversampler

A common oversampling method that involes randomly selecting the samples from the minority class into the training set.



SMOTE

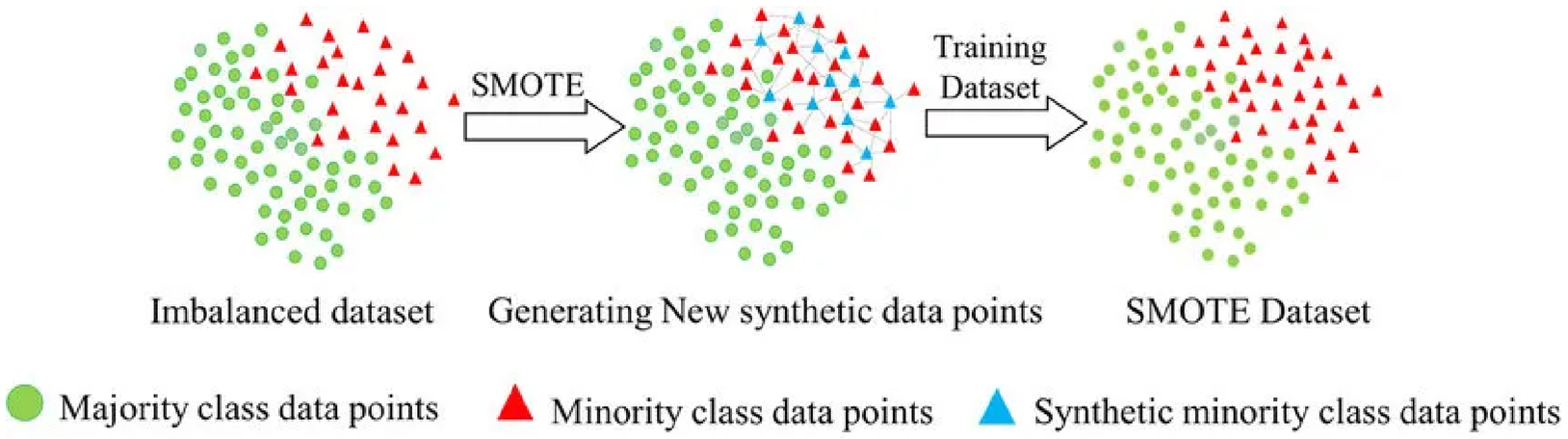
A popular method for the oversampling of the data which involves generating the synthetic data from the minority class.



Weight: w1 ∝ #(majority)/#(minority)
∝ # of synthesized data to generate

SMOTE-N

An extenstion of SMOTE which involves not only the numerical features, but also the categorical features of the dataset as well.



INTERACTIVE QUIZ GUIDELINES

In machine learning, Robust Scaler is particularly useful when dealing with datasets that contain outliers. It scales features usinQuiz Guidelines:
1) The quiz will be held on the Quizziz (https://quizizz.com/). You will join the quiz automatically by Clicking the link below.

2) Links of the quiz will be mailed to everyone by . There are many quizzes and each person will get link to their respective quiz. Quiz is based on first come first serve (Limit of 50 per quiz)

3) After clicking the link, you will be asked to write your name. Kindly set your name as your Registration Number Only.

5) The quiz will contain 10 MCQs based on our topic of discussion.

5) Winners will be officially announced by DSC in 24 hours and certificate will be given in a week.