FEATURE SCALING

Date [ ][ ][ ]

Note: usually we apply feature scaling at the end, means after all other feature transformations.

Agenda:
① What is feature Scaling?
② Why do we need feature Scaling?
③ Types of feature scaling.
④ Standardization
⑤ Normalization
⑥ Normalization Vs Standardization. (depends on algorithm)

What is feature Scaling?

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

Aim is to bring all the features (independent) on same scale.

ex: Features —→ Age, Height, Weight
[number   cm.   kg.] ← different units

So, these 3 features are now not comparable

To make them comparable we do scale them.

Why do we need feature scaling?

age	Salary	purchase (y/n)
10	10,000	0
15	25,000	1

say, this is classification problem having age & salary as independent features & based on that we need to find purchase (y/n).

Say, we are using KNN algorithm, which is distance based algorithm.

so, distance $= \sqrt{(15-10)^2 + (25000-10,000)^2}$

↑ there won't be much impact of Age feature

↑ This factor will dominate

so, as a result KNN won't perform well.

☀ Types of feature scaling :

Standardization   Normalization.
(Z-score normalization)

☀ Standardization :

Age	Standardized_age	formula :
10	$\left(\frac{10-30}{10}\right) = -2$	$\overset{i}{\underset{1}{\sum}}\left\{ x_i' = \dfrac{x_i - \bar{x}}{\sigma} \right\}$
20	$\left(\frac{20-30}{10}\right) = -1$	
30	$\left(\frac{30-30}{10}\right) = 0$	$\left\{ \begin{array}{l} \bar{x} = \text{mean} \\ \sigma = \text{std. dev.} \end{array} \right\}$
40	$\left(\frac{40-30}{10}\right) = 1$	

$(\text{say, } \bar{x} = 30, \ \sigma = 10)$.

→ Property of standardized value :
$\{ \text{mean} \approx 0, \ \text{std. dev.} \approx 1 \}$.

→ Geometric Intuition :
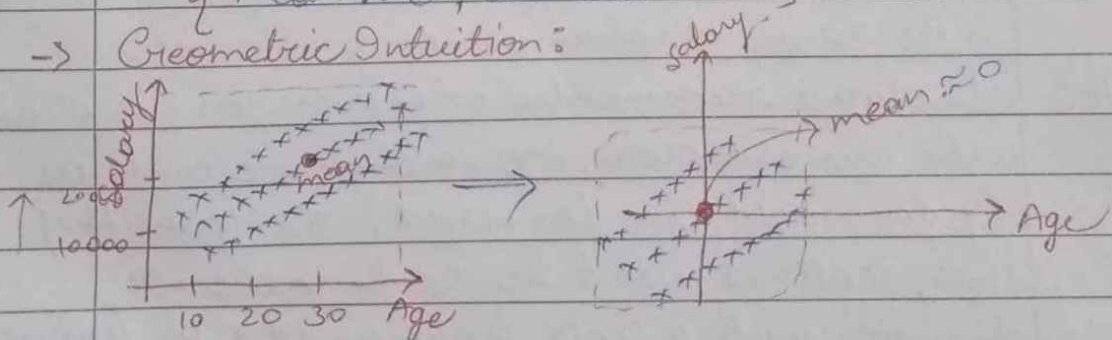
Before Standardization   After Standardization
                                of Age & salary both.

$\left\{ \begin{array}{l} \text{mean centering (mean} \approx 0\text{.)} \\ \text{squashed std. deviation } (\sigma \cong 1) \\ \text{scaling by the factor of std. dev.)} \end{array} \right.$

2

→ effect on outliers?

there is no impact of standardization on outliers. we have to explicitly deal with it.

→ when to use Standardization?

There is no loss on applying standardization. But, it will benefit us when applied on distance based alg

as, (i) K-means --> uses Euclidean Distances

(ii) KNN --> measures distances b/w pairs of data.

(iii) PCA --> try to get feature with max. variance

(iv) ANN --> Apply gradient descent

(v) Gradient Descent --> θ calculation becomes faster
   & learning rate is same.

(vi) Deep Learning --> Gradient descent.

⋋ Normalization:

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

Note: Its always recommended whenever you are dealing with numeric data, try to eliminate units as, (kg for weight, cm for height) bcz if we keep then, features won't be comparable. So, eliminate units & scale features. (bring all feature to common scale).

<div align="center">Types Normalization</div>

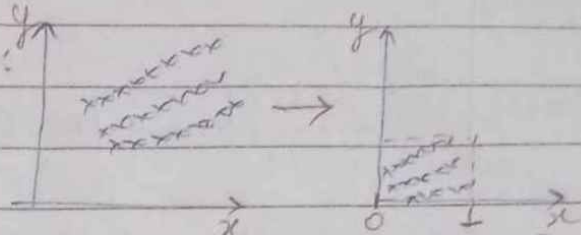| Min Max Scaling	Mean normalization	Max absolute	Robust scaling

↑
most used
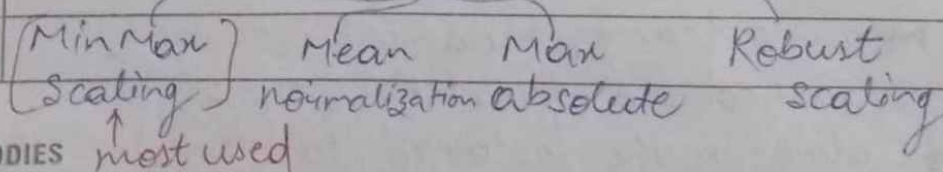
→ Min-Max Scaling :

formula: $$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$ ← when we know min & max.

Range → $[0, 1]$.

• Geometric Intuition :

after scaling x & y, data is squished b/w $[0, 1]$.
                                    unit sq., cube (3D), hyper (+3D)

• Note : here shape of distribution after scaling may change, unlike in Standardization.

• Effect on outliers : squishes.

→ Mean Normalization :
                                    → mean centering
formula : $$x_i' = \frac{x_i - x_{mean}}{x_{max} - x_{min}}$$ ↳ scaling b/w factor (max - min).

Range → $[-1, 1]$

Note : { Rarely used (instead of this Standardization is used)
        { There is no separate class for this in Sklearn.

→ Max Absolute scaling :
                                    → used when we have
formula : $$x_i' = \frac{x_i}{|x_{max}|}$$ sparse data (data with many zeros).

[class in sklearn → MaxAbsScalar].

→ Robust Scalar :

$$x_i' = \frac{x_i - x_{median}}{IQR = q_3 - q_1}$$ ← Robust to outliers (works better with data having outliers)

[class RobustScalar].

Note : Can't say which technique works best for data, we have to try & choose.
(Most of the times standardization is used.)
                                    #_____