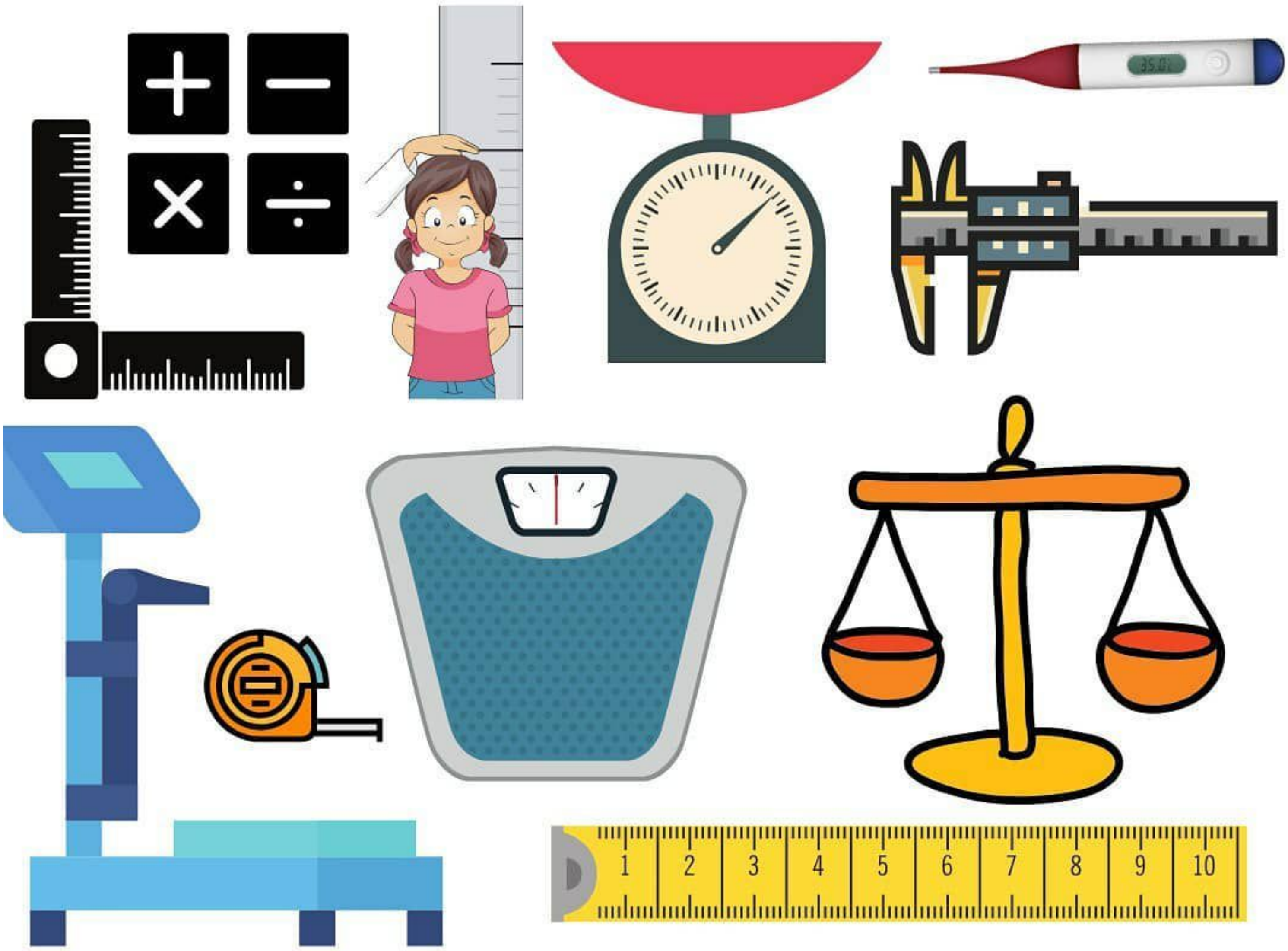
Feature Scaling



Swipe left >>>

Why???

Age	Income
25	20000
32	33000
35	40000
27	26000
45	56000
39	45000

- In the real world datasets we usually includes features that highly vary in magnitudes, units, and range.
- Let's see this with an example, consider below dataset with 2 features Age and Income. These two features vary a lot like income is 20X times more than Age.
- It is very important to scale the features because most of the Machine learning algorithms uses gradient descent or Eucledian distance.
- This Eucledian distance calculates the distance between two points and features which have the high magnitude will have more domination(more weight) than others (This is a big problem because we need to have equal weight for every feature)
- It is also important in gradient descent to make faster convergence as features will be in same variance range.

Swipe left >>>

How??

Standardisation (also called z-score normalization)

- values are centered around the mean with a unit standard deviation.
- Transforms your data such that the resulting distribution has a mean of 0 and a standard deviation of 1.
- The values are not restricted to a particular range.

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

$$\mu = \text{Mean}$$

$$\sigma = \text{Standard Deviation}$$

Swipe left >>>

How??

Normalisation

- Normalization means dividing the feature with the Norm of the the vector . We basically want the euclidean distance of the vector to be 1.

Mean Normalisation

- It bring values between -1 and 1 with μ=0.

$$z = \frac{x - \text{mean(x)}}{\max(x) - \min(x)}$$

Min-Max Normalisation

- This scaling brings the value between 0 and 1.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Swipe left >>>

Normalization vs Standardization

Normalisation

- It is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors and Neural Networks.

Standardization

- It can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

- **It is a good practice to fit the scaler on the training data and then use it to transform the testing data. This would avoid any data leakage during the model testing process. Also, the scaling of target values is generally not required.**

Swipe left >>>

Where???

- When ever you are using algorithms which calculates distance or expects data which is zero centric or assums normality

Examples

- Linear Discriminant Analysis(LDA)
- PCA
- KNN
- Gradient descent
- Tree-based models
- Naive Bayes
- Linear / Logistic regression (When regularized)
- Neural networks.

Swipe left >>>