Feature Scaling

Features: Weight (100 kg)

- Magnitude (100)
- Units (Kg)

Feature Scaling

Few advantages of normalizing the data are as follows:

1. It makes your training faster.
2. It prevents you from getting stuck in local optima.
3. It gives you a better error surface shape.
4. Wweight decay and bayes optimization can be done more conveniently.

Hovewer, there are few algorithms such as Logistic Regression and Decision Trees that are not affected by scaling of input data.

Feature Scaling

Examples of Algorithms where Feature Scaling matters:

1. **K-Means** uses the Euclidean distance measure here feature scaling matters.
2. **K-Nearest-Neighbours** also require feature scaling.
3. **Principal Component Analysis (PCA)**: Tries to get the feature with maximum variance, here too feature scaling is required.
4. **Gradient Descent**: Calculation speed increase as Theta calculation becomes faster after feature scaling.

Note: Naive Bayes, Linear Discriminant Analysis, and Tree-Based models are not affected by feature scaling.

Feature Scaling (Before)

	Student	CGPA	Salary '000
0	1	3.0	60
1	2	3.0	40
2	3	4.0	40
3	4	4.5	50
4	5	4.2	52

Feature Scaling (After)

	Student	CGPA	Salary '000
0	1	-1.184341	1.520013
1	2	-1.184341	-1.100699
2	3	0.416120	-1.100699
3	4	1.216350	0.209657
4	5	0.736212	0.471728

Feature Scaling

- Distance AB before scaling => $\sqrt{(40-60)^2+(3-3)^2}=20$
- Distance BC before scaling => $\sqrt{(40-40)^2+(4-3)^2}=1$
- Distance AB after scaling => $\sqrt{(1.1+1.5)^2+(1.18-1.18)^2}=2.6$
- Distance BC after scaling => $\sqrt{(1.1-1.1)^2+(0.41+1.18)^2}=1.59$

Feature Scaling

- Normalization:

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

- Standardisation:

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}}$$