Data Transformation

Feature Scaling

In a dataset, Features can have different *measurement scales* like in the above data set MRP and Weight of items have a different scale. This makes a large difference in numbers. like below. Item_MRP is in hundreds and weight is in units and item_visibility is in decimals. This is a significant obstacle as some *machine learning algorithms* are highly sensitive to these differences in numeric features. Whereas some are totally insensitive to them.

Feature scaling is transforming data to bring multiscale features to the same scale

Algorithms like Neural Networks and logistic regression that use the *gradient descent* technique are very sensitive to feature scaling as they work on the same step size for all features and gradient convergence will be slow for multiscale data.

Gradient-based algorithm needs feature scaling to converge gradient faster and improve model speed

The Algorithms like KNN, SVM, and K-Means, are *distance-based algorithms* that are dependent on the distance between two data points and hence, can bias towards features with high numeric values. Hence, they are also sensitive to feature scaling.

Distance-based algorithms need feature scaling to avoid the Bias problem

Whereas *Tree-based algorithms* are insensitive to scale and units of feature as they split the nodes on a feature that increases the homogeneity of the node.

Hence these algorithms are insensitive to Feature Scaling and don't need it.
These two key methods/Techniques of Feature Scaling

1. **Data Normalization**
2. **Data Standardization**