Data Preprocessing

Feature Scaling

Feature Scaling

When the range of values are very distinct in each columns , we need to scale them to the common level.It is a technique of bringing down the values of all the independent features of our dataset on the same scale.

Type of Feature Scaling :

1. Normalization :

range(0 to 1):

Xnew = (X – Xmin)/(Xmax – Xmin)

2. Standardization

(mean=0 and std=1):

Xnew = (Xi – Xmean) / std

3. Robust Scalar :

It is best scaling technique when we have outliers present in our dataset. It scales the data according to the IQR (inter quantile range) q1 – 25 percentile and q3 – 75 quantile.

Xnew = (X – Xmedian)/ IQR

4. Gaussian Transformation :

When our dataset does not follow gaussian / normal distribution (bell curve) then we used gaussian transformation.

4.1. Logarithmic Transformation : This transformation is not applied to those features which have negative values. This transformation is not applied to those features which have negative values. This transformation is mostly applied to right-skewed data.

Data['feature_name'] = np.log(data['feature_name'])

4.2 Reciprocal Transformation:

This transformation is not defined for zero. It is a powerful transformation with a **radical effect**.This transformation reverses the order among values of the same sign, so large values become smaller.

```
Data['feature_name']  = 1 / data['feature_name']
```

4.3 Square Transformation:

This transformation mostly applies to **left-skewed data**.

4.4 Square Root Transformation :

This can be used for reducing the **skewness of right-skewed data**.

```
Data['feature_name']  = data['feature_name']  ** (½)
```

4.5 Exponential Transformation :

```
Data['feature_name']  = data['feature_name']  **(½)
```

4.6 Box-Cox Transformation :

```
Data['feature_name'] , parameters = stat.boxcox(data['feature_name'])
```