Feature Transformations

Why do we need transformations?

To improve the model performance - as these transformations conform to model assumptions and which in turn amplifies model's predictive power and increases the quality of the model

1. It can even out the variance
2. To make the feature more normal
3. It can reduce the skew
4. It can linearize the relationship between the feature and target
5. Reduce the impact of outliers

More applicable on linear models(parametric models)
Another term is variable transformation which includes target column as well.

Problems with Feature transformations:
Less interpretability.
Finding the best transformation is tricky.
Add one more step in pipeline.

Log Transformation

Log transformation compress the larger values more than smaller values.
Works effectively on right skew data(pd.skew()=1 to 4).
Also reduces outlier impact.
To reduce heteroscedasticity
Invertible transformation.
Will not works on negative and zero values, when data is already normal, interpretability gets difficult.

Algorithms that benefit from Log Transform:

1. Linear Models
2. ANOVA
3. Time series analysis
4. K-Means
5. PCA
6. Gaussian Naïve Bayes
7. Training of Neural Networks

Squareroot Transformation

Similar to log transformation but log transformation is very extreme compression to big values.
(pd.skew()=0 to 1)

Square Transformation

When mild left skew data(pd.skew()= -1 to 0)
To linearize non linear relationship.

Reciprocal Transformation

When strong right skew data(pd.skew()=4 or greater)
When inverse hyperbolic relation

Reflect Transformation
For extreme left skewed,
Reflect, shift, mirror

```
g the exponential distribution
ot(np.log1p(-df['b'] + 396.900000001),
```

Pipeline of transformation (01:44:19)

Complex transformation

When you have lot of columns, you cannot apply individual feature transformation by checking each, since it would be very tedious task.

Box-Cox Transformation

$$Y'(\lambda) = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(Y) & \text{if } \lambda = 0, \end{cases}$$

Lambda from(-5,5)
Lambda use Maximum Likelihood estimation(MLE) in backend.

When to use:

1. Handle Skewness
2. Handle non-normal data
3. Heteroscedasticity
4. Can act as general case to many other transformations

When not to use:

1. Negative values
2. Interpretability is a concern
3. Data is already normally distributed
4. Categorical Data

Yeo Johnson Transformation

Improved version of Box-Cox Transformation, where negative values could also be applied.

For $y \in \mathbb{R}$ and $\lambda \in \mathbb{R}$, the transformed variable $y'(\lambda)$ is given by:

$$y'(\lambda) = \begin{cases} [(y+1)^{\lambda} - 1]/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[-y+1)^{2-\lambda} - 1]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

One can get more clarity after seeing graphs of the standard equations of these techniques(eg. y=x2)