Feature Scaling

Feature Engineering is a big part of Data Science and Machine Learning. Feature Scaling is one of the last steps in the whole life cycle of Feature Engineering. It is a technique to standardize the independent features in a data in a fixed range or scale.

Types of Feature Scaling :

- **Standardization:**
  - Standard Scaler
- **Normalization:**
  - Min Max Scaling
  - Mean Normalization
  - Max Absolute Scaling
  - Robust Scaling *etc.*

Standardization:
Standardization is a scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
df_new = pd.DataFrame(sc.fit_transform(df), columns=df.columns)
```

Normalization:
Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information.

Min Max Scaling : Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1. Min Max Normalization will perform best when the maximum and minimum value is very distinct and known.

```
from sklearn.preprocessing import MinMaxScaler
mm = MinMaxScaler()
df_new_mm = pd.DataFrame(mm.fit_transform(df), columns=df.columns)
```

Mean Normalization : It is very similar to Min Max Scaling, just that we use mean to normalize the data. Removes the mean from the data and scales it into max and min values.

Max Absolute Scaling : Scale each feature by its maximum absolute value. This estimator scales and translates each feature individually such that the maximal absolute value of each feature in the training set will be 1.0. It does not shift/center the data, and thus does not destroy any sparsity. This scaler can also be applied to sparse CSR or CSC matrices. Max Absolute scaling will perform a lot better in sparse data or when most of the values are 0.

```
from sklearn.preprocessing import MaxAbsScaler
ma = MaxAbsScaler()
df_new_ma = pd.DataFrame(ma.fit_transform(df), columns=df.columns)
```

Robust Scaling : This Scaler removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile). Robust Scaling is best for data that has outliers.

```
from sklearn.preprocessing import RobustScaler
rs = RobustScaler()
df_new_rs = pd.DataFrame(rs.fit_transform(df), columns=df.columns)
```