LET THE DATA CONFESS
Understand | Learn | Code | Implement

All you need to know about

DATA SCALING TECHNIQUES

Ace your next data science interview
Practice Now: https://www.letthedataconfess.com/mock-intervie

Three popular Data Scaling techniques are:

Normalisation

Standardisation

Rescaling

Rescaling:

- **Rescaling** a vector means to add or subtract a constant and then multiply or divide by a constant, as you would do to change the units of measurement of the data.

- For eg: To convert a temperature from Celsius to Fahrenheit.

Normalisation:

- **Normalizing** a vector most often means dividing by a norm of the vector. It is also often known as Min-Max scaling, as it refers to rescaling by the minimum and range of the vector, to make all the elements lie between 0 and 1 thus bringing all the values of numeric columns in the dataset to a common scale.

- It is sensitive to outliers.

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

When to use Normalisation?

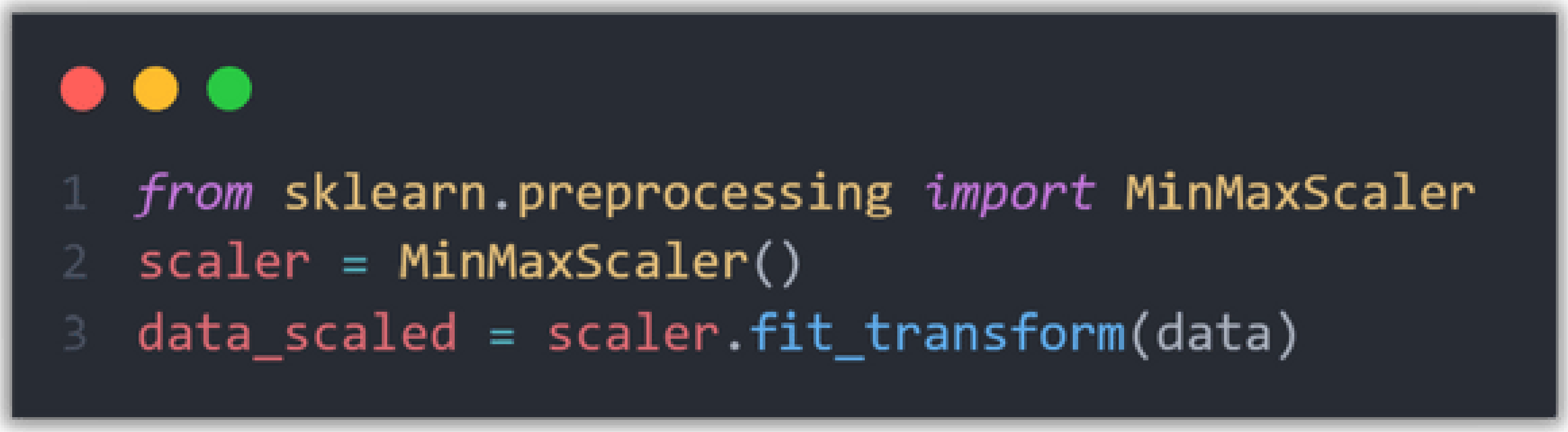
- For machine learning, every dataset does not require normalization. It is required only when features have different ranges. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

- **For eg:** Consider a data set containing two features, area of house and price of house, where area ranges from 100–500, while price ranges from 100,000–500,000 and higher. Price of house is about 1,000 times larger than area. So, these two features are in very different ranges. But in algorithms like multi-variate linear regression, price feature will influence the target variable more because of its larger value, even though it might not be more important independent variable. So we normalize the data to bring all the variables in the dataset to the same range.

- Good to use in algorithms which does not make any assumptions about the distribution of the data, like K-nearest neighbours or Artificial Neural Networks.

Code:

We can import MinMaxScalar from Scikit-learn and apply it to our dataset.



```python
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)
```

Standardisation:

- **Standardizing** a vector most often means subtracting a measure of location and dividing by a measure of scale. The values are centred around the mean with a unit standard deviation. For example, if the vector contains random values with a Normal distribution, you might subtract the mean and divide by the standard deviation, thereby obtaining a “standard normal” random variable with mean 0 and standard deviation 1.

- It is also highly influenced by outliers.

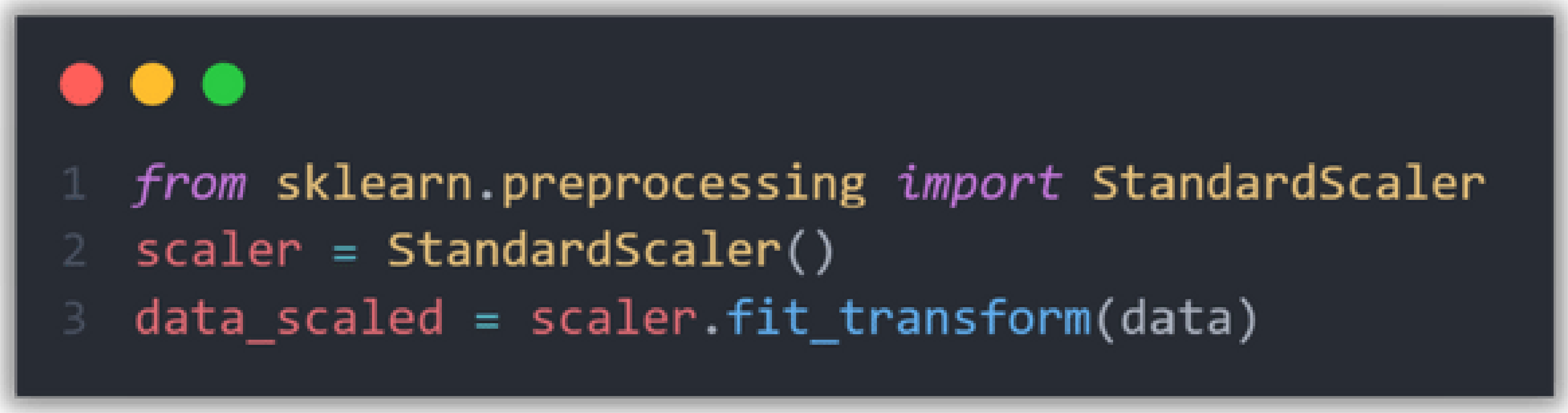
$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$$

When to perform Standardisation?

- We need to perform Standardisation when we compare measurements that have different units. This is to bring all the features on the same scale.
- **Eg:** There might be a features whose some values are in kms, while some values may be in cm. So by default, the algorithm will give more importance/weight to values in kms as compared to values in cm. This will result in variables that are measured at different scales not contributing equally to the analysis and thus, it will induce bias. To avoid this scenario, standardisation of variables is necessary.
- Good to use in algorithms which make assumptions about the distribution of the data, like linear regression, logistic regression, and linear discriminant analysis. Also, this technique is more effective if your data is normally distributed.

Code: </>

StandardScaler from sci-kit-learn library removes the mean and scales the data to unit variance. We can import the StandardScalar method from sci-kit learn and apply it to our dataset.



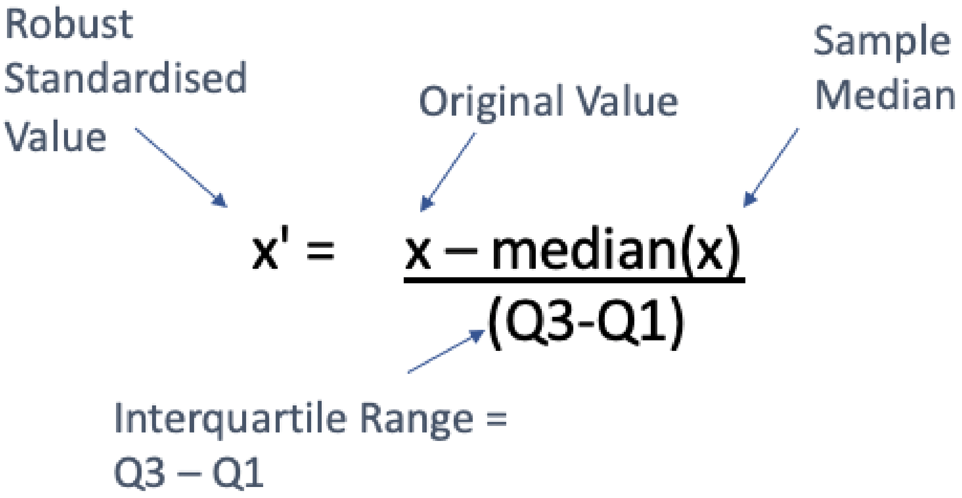
```python
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

Robust Scalar:

- Since normalisation and standardisation are sensitive to outliers, this technique scales features using statistics that are robust to outliers.

- Here, we use median and quantiles for scaling and the technique consists of subtracting the median to all the observations and then dividing by the interquartile difference.

- The interquartile difference, known as IQR is the difference between the 75th and 25th quantile.



Robust Standardised Value

Original Value

Sample Median

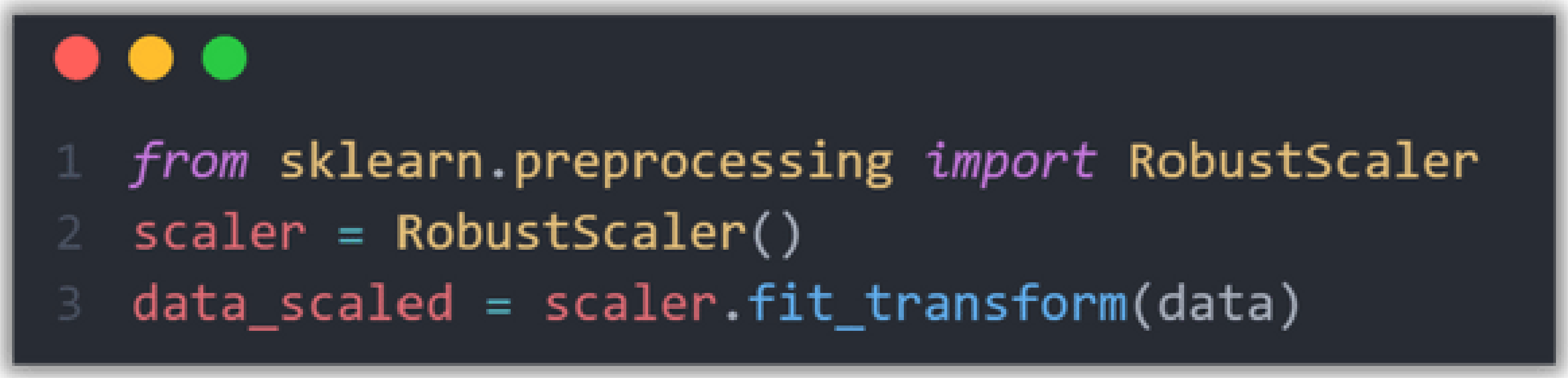
$$x' = \frac{x - \text{median}(x)}{(Q3\text{-}Q1)}$$

Interquartile Range = Q3 – Q1

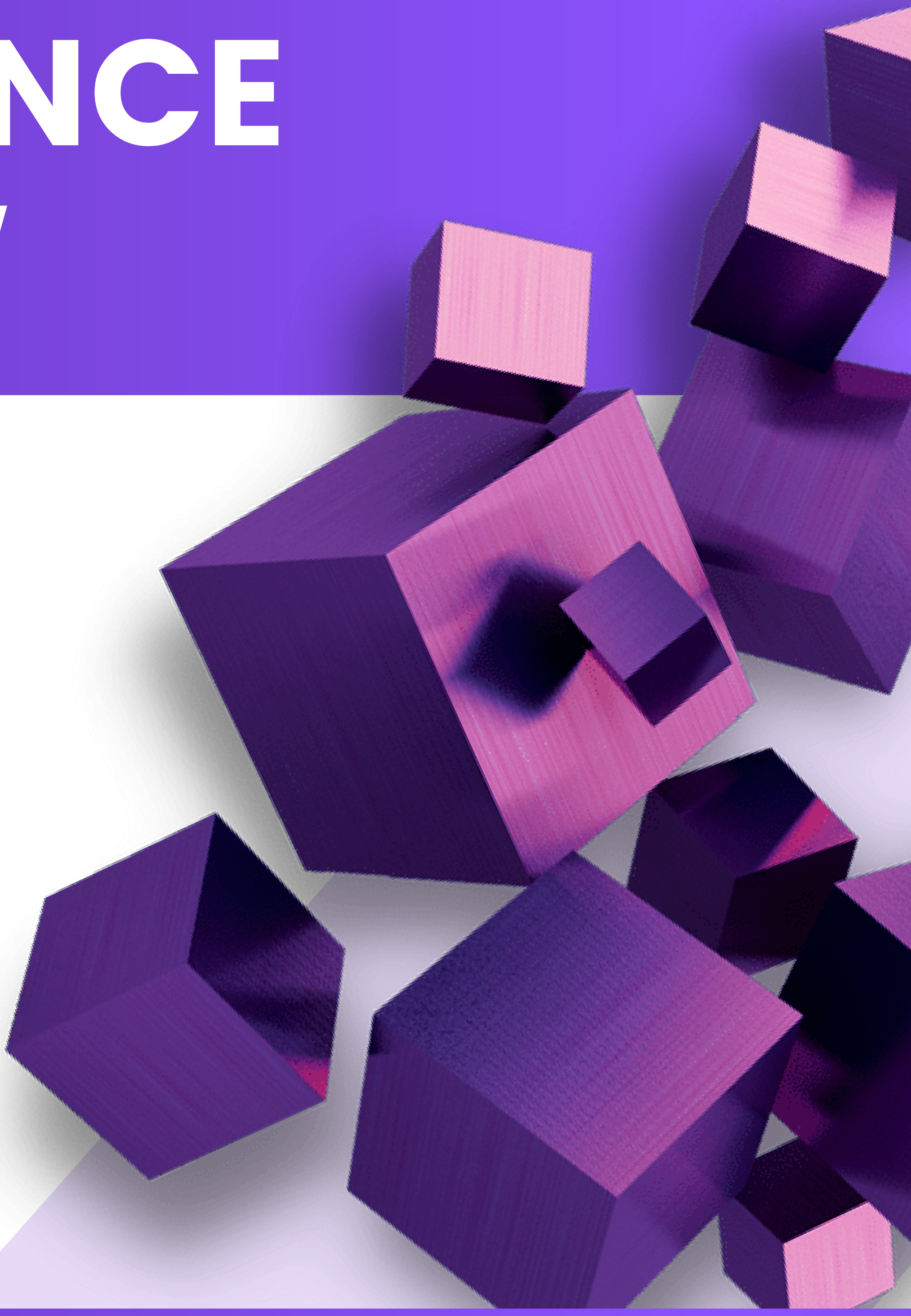
Code:

</>

We can import RobustScalar from Scikit-learn and apply it to our dataset.



```python
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
data_scaled = scaler.fit_transform(data)
```



www.letthedataconfess.com

LET THE DATA CONFESS

Understand | Learn | Code | Implement