Transformations

In the notebook on non-Homogeneous data (Becoming_a_successful_Data_Scientist.ipynb) we hopefully have motivated

- the need to process your features (and sometimes target)
- from "raw" to "synthesized" form

This process is called **feature engineering**.

Knowing how to perform feature engineering is a key skill of a Data Scientist.

We explore this topic in more depth in this module

Transformation: basics

Before diving into the many different types of transformations

- we want to establish some basics

Let's review the Mechanics of transformations (Transformations_Mechanics.ipynb)

For a refresher on implementation in `sklearn`

- Revisit our previous module Coding transformations in `sklearn` (Transformations_Pipelines.ipynb))

The why's of transformations

We illustrate several useful types of transformations below.

This is organized more as a "case study" than a taxonomy.

- Many transformations have multiple uses, making a hard taxonomy difficult

Missing features

Sometimes, your examples have all the "information" you need, but in the wrong form.

Creating new "synthetic" features from raw features is one way of making this information available to the model.

We have seen several illustrations of this in prior notebook section on Normalization (Becoming_a_successful_Data_Scientist.ipynb)

But sometimes, the problem is the *absence* of a relevant feature.

Let's look at the notebook Transformations: adding a missing feature (Transformations_Missing_Features.ipynb)

Scaling

There is a class of transformations that alter the *scale* (magnitude) of features or targets.

In the Recipe for ML, Scaling is treated as separate from the other transformations.

Perhaps this is because scaling is sometimes performed

- *Not* strictly because of the relationship between target and features
- But because of the mathematics of the *loss function*

Let's visit the notebook Transformations: scaling (Transformations_Scaling.ipynb)

Other transformations

For other transformations see the notebook Other Transformations (ransformations_Other.ipynb)

In [ ]:
```
print("Done")
```