**can we apply feature scaling right before splitting the dataset into the training and test sets?**

[Jili](https://www.udemy.com/user/jili-li/" \t "/home/ddxy/Documents\\x/_blank) · 2 天前

In tutorial 15, I am confused by "X\_test = sc\_X.transform(X\_test)". I still do not quite get the purpose of this line. Does it mean it uses the descriptive stats calculated based on the X\_train set and apply these descriptive stats to standardise X\_test?

Then I wonder if it would be more intuitive to do standardisation before splitting the dataset?

Thank you,

**关注的回应**

IMG_256

[Maiia](https://www.udemy.com/user/maiia-bakhova/" \t "/home/ddxy/Documents\\x/_blank) — 助教

 · 1 天前  答案

Hello Jili,

since Python always utilizes an Object-Oriented Programming then any method depends on an object to which it is applied. For example when fit()  method is applied to regressor or classifier it computes a machine learning model: Linear Regression, Decision Tree and such.

The fit\_transform()  method in our lectures is used to modify numpy arrays. It is paired with transform()  method (not with fit()   for regressors and classifiers). They both work with an object from StandardScaler()   class. As an object it may have attributes, and when it does we can use them to store some information. It is an object-oriented programming feature.

The fit\_transform()  method contains its own fitting and transformation.   
The fitting part here is used to analyse the data on which we apply the object (getting the mean, the min, the max, the standard deviation, outliers, etc.) in order to understand how the data is structured.   
Then once the object understands how the data is structured thanks to the fitting, the transformation part is used to apply a required transformation (like feature scaling for example).   
And since the test set and the training set have very similar structures, we don't need to create a new scaling object that we fit to the test set and then use to transform the test set, we can directly use the object already created and fitted to the training set, to transform the test set. This is why we apply to X\_test only transform() .

As for the order between splitting and transformation then the approaches here differ. Stricly speaking when we try to make our test set to imitate future data we should base pur scaling on training set, because we pretend not to know about the future numbers.

Nevertheless in real life our data sets are huge, so means and standard deviations on training and test sets are almost the same. In the same time pre-processing could be quite involved and require more steps. For example different columns may need different ways of imputing of missing data. Since we need to repeat all this steps on a test set, too, some people prefer to do all pre-processing together with scaling before splitting data.

Best,

Mya