What and why behind fit_transform() and transform() in scikit-learn!

Chetna Khanna Aug 25, 2020 · 3 min read

Scikit-learn is the most useful library for machine learning in Python programming language. It has a lot of tools to build a machine learning model and is quite easy to use too. Yet, we struggle at times to understand some of the very simple methods which we generally always use while building our machine learning model.

One such method is **fit_transform()** and another one is **transform()**. Both are the methods of class **sklearn.preprocessing.StandardScaler()** and used almost together while scaling or standardizing our training and test data.

Photo by Tekton from Unspalsh

The motivation to write this blog came from multiple questions posted on these methods in an online course on Machine Learning.

> ***The question is:***
>
> ***Why we use fit_transform() on training data but transform() on the test data?***

We all know that we call fit_transform() method on our training data and transform() method on our test data. But the actual question is why do we do this? My motive is to explain this simple yet confusing point in the simplest possible manner. So let's get started!

Suppose we are building a k-Nearest Neighbor model and we have to scale our features. The most common way to scale the features is through scikit-learn's StandardScaler class.

Note:

1. *Data standardization is the process of rescaling the attributes so that they have mean as 0 and variance as 1.*
2. *The ultimate goal to perform standardization is to bring down all the features to a common scale without distorting the differences in the range of the values.*
3. *In sklearn.preprocessing.StandardScaler(), centering and scaling happens independently on each feature.*

The magical formula which performs standardization:

Image by Author

Let's now deep dive into the concept.

fit_transform()

fit_transform() is used on the training data so that we can scale the training data and also learn the scaling parameters of that data. Here, the model built by us will learn the mean and variance of the features of the training set. These learned parameters are then used to scale our test data.

So what actually is happening here! 🤔

The fit method is calculating the mean and variance of each of the features present in our data. The transform method is transforming all the features using the respective mean and variance.

Now, we want scaling to be applied to our test data too and at the same time do not want to be biased with our model. We want our test data to be a completely new and a surprise set for our model. The transform method helps us in this case.

> *Related Article — Want to know about* ***Multicollinearity****? Read here*

transform()

Using the transform method we can use the same mean and variance as it is calculated from our training data to transform our test data. Thus, the parameters learned by our model using the training data will help us to transform our test data.

Now the question is why we did this? 🙃

Here is the simple logic behind it!

If we will use the fit method on our test data too, we will compute a new mean and variance that is a new scale for each feature and will let our model learn about our test

data which is the ultimate goal of building a model using machine learning algorithm.

This is the standard procedure to scale our data while building a machine learning model so that our model is not biased towards a particular feature of the dataset and at the same time prevents our model to learn the features/values/trends of our test data.

I hope this explanation will help you understand the simple logic behind these methods.

Reference:

sklearn.preprocessing.StandardScaler — scikit-learn 0.23.2 documentation

Standardize features by removing the mean and scaling to unit variance The standard score of a sample is calculated as...

scikit-learn.org

. . .

This is my very first blog. Please share your comments and suggestions to improve this blog post.

LinkedIn

Machine Lear Scikit Learn Data Science Python