

Training Day-12 Report:

fit(), transform() and fit_transform() Methods in Python:-

It's safe to say that scikit-learn, sometimes known as sklearn, is one of Python's most influential and popular Machine Learning packages. It includes a complete collection of algorithms and modeling techniques that are ready to be trained, including utilities for pre-processing, training, and grading models.

One of the most elementary classes in Sklearn is the transformer, which implements three different methods: fit(), transform(), and fit transform().

➤ **fit() Method:-**

In the fit() method, we apply the necessary formula to the feature of the input data we want to change and compute the result before fitting the result to the transformer. We must use the .fit() method after the transformer object.

If the StandardScaler object sc is created, then applying the .fit() method will calculate the mean (μ) and the standard deviation (σ) of the particular feature F. We can use these parameters later for analysis.

Let's use the pre-processing transformer known as StandardScaler as an example and assume that we have to scale the features of self-created data. The example dataset in the code below is created using the arrange method and then divided into the training and testing datasets. After that, we create a StandardScaler instance and fit the feature of the training data to it to determine the mean and standard deviation to be utilized for scaling in the future.

The significance of separating the dataset into the train and test datasets before using any pre-processing process, such as scaling, must be emphasized. Test data points represent real-world data. Therefore, we must only execute fit() to the training feature to prevent future data to our model.

➤ **transform() Method:-**

To change the data, we most likely use the transform() function, where we perform the calculations from fit() to each value in feature F. We transform

the fit computations. Hence we must use `.transform()` after we have applied the fit object.

When we make an object using the fit method, we utilize the example from the section above and place the object in front of the.

The scale of the data points is transformed using the transform and fit transform method, and the output we receive is always a sparse matrix or array.

➤ **fit_transform() Method:-**

The training data is scaled, and its scaling parameters are determined by applying a `fit_transform()` to the training data. The model we created, in this case, will discover the mean and variance of the characteristics in the training set.

The mean and variance of every feature reported in our data are calculated using the fit approach. The transform method transforms all features using the corresponding means and variances.

We wish scaling to be implemented in our testing data, but we also don't want our model to be biased. We expect our test set of data to be entirely fresh and unexpected for our model. In this situation, the convert approach is useful.