

Understanding sklearn

A library to work with machine learning models like classification, clustering, regression, neural networks...

You need to import sklearn to work in it.

```
import sklearn
```

then you will also need some model to be imported ...

```
from sklearn.selector import model
```

here you can chose selector as per your model requirement like:-

```
from sklearn.cluster import kmeans
```

*use 'shift+tab' to know about the available selectors and models after '.'

We need to split the data as training set and testing set. This can be done by importing 'train_test_split' method from sklearn like this:-

```
from sklearn.model_selection import train_test_split  
train_data, test_data, train_label, test_label =  
train_test_split(X, Y)
```

Optionally you can specify the size of the test data as well by adding parameter size=0.3

Now you know how to split data for training and testing purposes.

Next step is to fit or train the model and it is done using `.fit()` method like this:-

```
model.fit(train_data, train_label)
```

Next step is to get the predictions done. It is done using `.predict()` method like this:-

```
model.predict(test_data)
```

You can then evaluate your model. The method depends upon the algorithms/ML model you are using.

You can import various available evaluation methods :-

```
from sklearn.metrics import r_score
```

remember

model.fit(): is used to fit training data

takes data and labels as arguments for supervised learning

takes only data as argument for unsupervised learning

model.predict(): used to predict labels of the new/test data set.

Returns the learned labels for each object in the test/new data array.

remember

model.predict_proba(): is used for classification problems. Returns the probability of category/label test set has.

model.score(): used in supervised learning. Returns score value between 0-1, larger score value means better fit.

remember

model.transform(): works in unsupervised learning. Takes train_data as an argument and transforms it into a new representation based on the unsupervised model.

model.fit_transform(): performs fit and transform on the train_data.