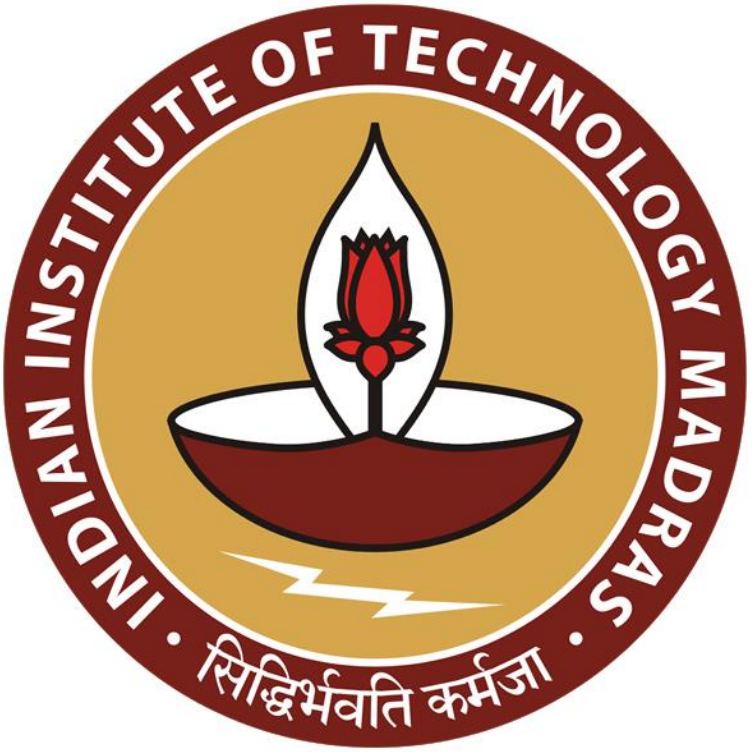


IIT Madras

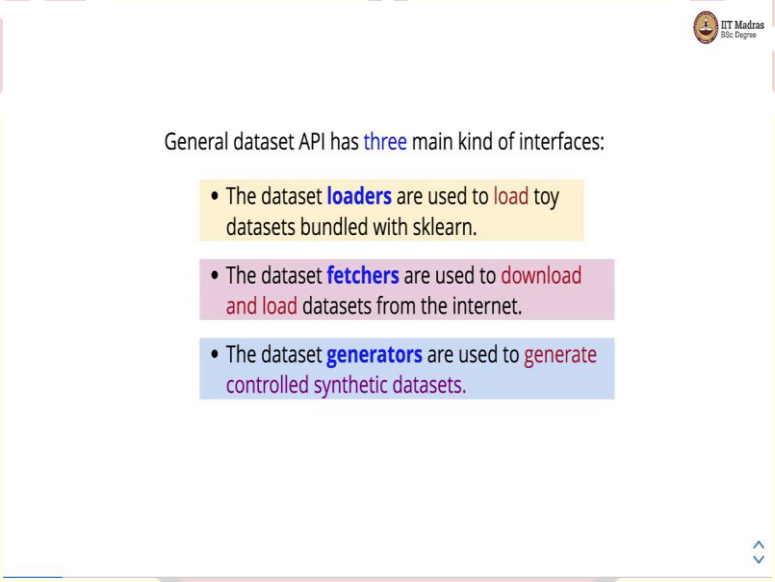
ONLINE DEGREE

Machine Learning Practice
Online Degree Programme
B. Sc in Programming and Data Science
Diploma Level
Dr. Ashish Tendulkar
Indian Institute of Technology – Madras

Data Loading

Namaste welcome to the next video of machine learning practices. In this video will discuss data loading functionality of the sklearn library. Data loading functionality helps us to load training data in different formats for our experimentation.

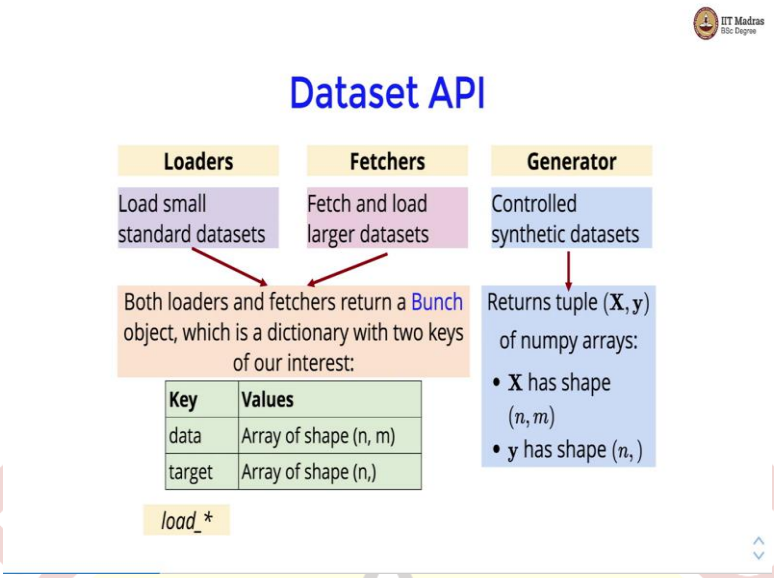
(Refer Slide Time: 00:30)



Sklearn provides a general dataset api that has three main kinds of interfaces. The dataset loaders, the dataset fetchers and the dataset generators. The dataset loaders are used to load toy datasets bundled with a sklearn. Then there are datasets on the internet which are large and are not bundled with a sklearn. So, dataset fetchers help us to download and load this dataset on the internet.

And then finally you have dataset generators that are used to generate controlled synthetic datasets with specific statistical properties.

(Refer Slide Time: 01:15)



So, we have loaders we have fetchers and we have generators. Loaders load small standard datasets that come bundled with sklearn. Fetchers fetch and load larger datasets and generators help us to generate controlled synthetic datasets with specific statistical properties. Both loaders and fetchers written a bunch of object which is a dictionary with two keys of our interest one key is the data and second is a target.

So, data the data key has a feature metrics of shape n, m and n is the number of examples and m is the number of fetchers. And we have target which is the label and the label is a vector which shape n. On the other hand, generators return a couple X, y where X is a feature metrics of shape (n, m) and y is a label vector with shape (n,) both of these both of these feature metrics and label vector n by arrays.

So, loaders generally start with load underscore that is what the load of function start with. The fetchers starts with fetch underscore and generator start with make underscore loaders and fetchers can also return tuples if we said return underscore x underscore y argument of loaders and fetchers functions the true.

(Refer Slide Time: 03:08)

Dataset Loaders

Dataset Loader	# samples (n)	# features (m)	# labels	Type
`load_iris`	150	3	1	Classification
`load_diabetes`	442	10	1	Regression
`load_digits`	1797	64	1	Classification
`load_linnerud`	20	3	3	Regression (multi output)
`load_wine`	178	13	1	Classification
`load_breast_cancer`	569	30	1	Classification

Note: These datasets are bundled with sklearn and we do not require to download them from external sources.

Let us look at dataset loaders. There are seven dataset loaders and you can see that these dataset loaders and loading datasets that are that are comparatively smaller in sizes. So, you can see that this is iris dataset which has got 150 samples with three features and a single label. This is a classification problem. Then we have diabetes dataset with 442 samples ten features and a single label.

And diabetes dataset is used to train regression models. Then digit dataset has 1797 samples each with 64 features a single label. Then we have then we have multi output regression dataset which is linear root dataset with 20y samples 3 fetchers and 3 labels. Then we have low wine dataset or wine dataset with 178 samples 569 samples 30 features each and a single label.

These datasets are bundled with a sklearn and we do not need to download them from the external sources. You can see that five of these datasets are used for classification like iris, digit, wine and the breast cancer dataset. Whereas diabetes dataset is used for regression and liberal dataset is used for multi output regression.

(Refer Slide Time: 05:08)

Dataset Fetchers

Dataset Loader	# samples (n)	# features (m)	# labels	Type
`fetch_olivetti_faces`	400	4096	1 (40)	multi-class image classification
`fetch_20newsgroups`	18846	1	1 (20)	(multi-class) text classification
`fetch_lfw_people`	13233	5828	1 (5749)	(multi-class) image classification
`fetch_covtype`	581012	54	1 (7)	(multi-class) classification
`fetch_rcv1`	804414	47236	1 (103)	(multi-class) classification
`fetch_kddcup99`	4898431	41	1	(multi-class) classification
`fetch_california_housing`	20640	8	1	regression

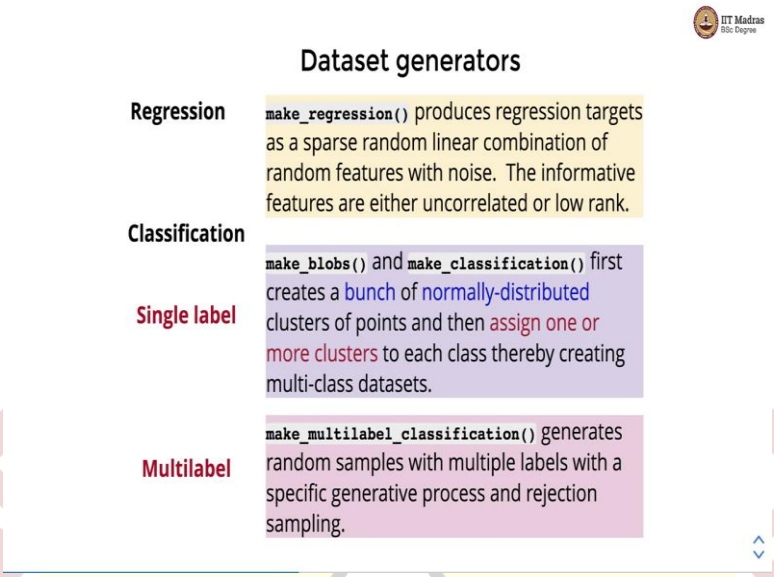
So, there are other larger datasets that are not bundled with a sklearn but they are available on the external sources like on the internet. And if you want to access these datasets generally we need to download them on the disk and then load them from there. So on a sklearn fetches us dataset fetch us is out that functionality for us. They download and load certain datasets from the external sources for us.

So, there are datasets like faces datasets with 400 samples but very large number of fetchers like 4096 features and a single label the total number of labels are 40. So, it is a multi-class image classification problem. So, then we have 20 news group dataset which is quite famous dataset for text classification has 18846 samples and a single label the total number of labels are 20.

Then we have people dataset which is again an image classification dataset with 13000 odd samples with about 5800 features and a single label but the total number of labels are fairly large, since the large multi class image classification problem. So, these are some of the datasets that are available in the dataset fetchers. Then this is a California housing dataset which will be which we will use quite often in this course.

And this California housing dataset has 20640 samples each with 8 features and a single label and since we are predicting the price of the house it is a regression problem and California housing dataset should be used for demonstrating a bunch of regression task. Then we have other very large datasets like a kiddcup99 dataset with about 4.8 million samples each with 41 features and you know it is used for multi-class classification problem.

(Refer Slide Time: 07:41)



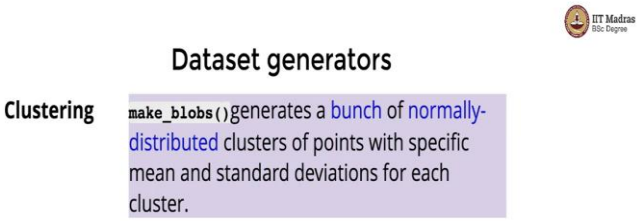
Sometimes we do not want to use the standard training set but instead you want to synthetic dataset with certain statistical properties. And sklearn on dataset generators help us to generate these kinds of datasets. So, there are functionalities to generate datasets for regression and classification also for clustering. So, make underscore regression function produces regression targets as a spa random linear combination of random features with noise.

And informative features are either uncorrelated or low rank. So, if you want to generate statistical control dataset synthetic dataset we can use make regression function and here we you can generate dataset with a single target or multiply target. We will see this function in action in the in the subsequent collapse. Then we have you know a bunch of functions for generating synthetic dataset for classification expense.

For single label we use make underscore blobs or make underscore classification function. They first create a bunch of normally distributed clusters of point and then assign one or more clusters to each class thereby creating multi-class datasets. We also have multi-level classification problem and for multilevel classification problem where each sample gets more than one label.

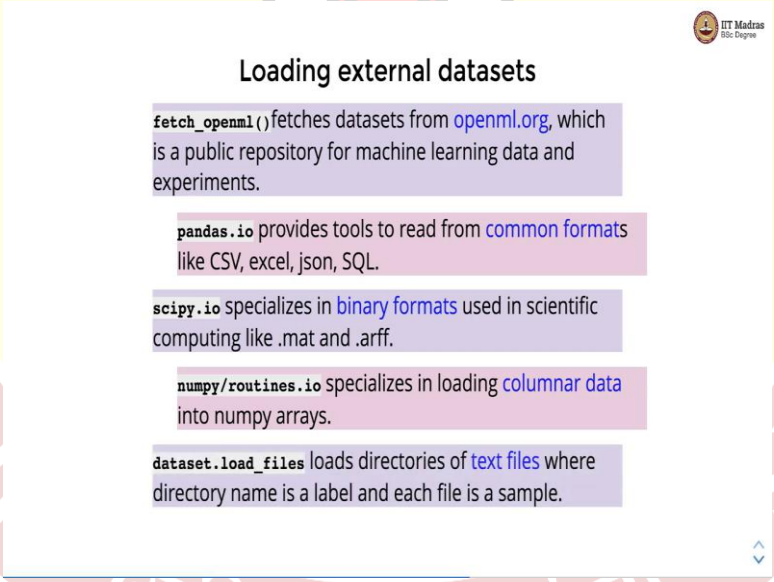
We have make underscore multi label underscore classification functions for generating random samples with multiple labels and here we use a very specific generative process and rejection sampling. So, that none of the sample gets zero label.

(Refer Slide Time: 09:33)



And for clustering we use make underscore blobs api for generating a bunch of normally distributed clusters a point specific mean and standard deviation for each cluster. So, these were three main interfaces provided by the dataset loader.

(Refer Slide Time: 09:55)



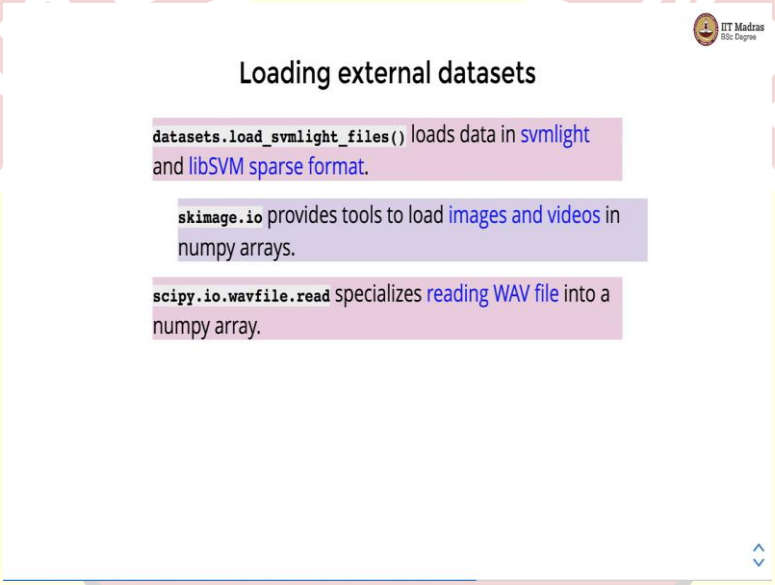
Apart from that there could be other excellent datasets that are directly supported by fetchers. So, for example there is a there is a repository called open email dot org which has got a lot of machine learning datasets and expense that are uploaded by different people. And we can use fetch underscore open ml api for fetching datasets from openml dot org.

Sometimes we might have our data in common formats like csv, excel, jason or sql and pandas provides tools to read data in this particular in this formats. And sometimes the data might be in binary formats like dot mat or dot arff and mainly used in scientific computing and in such

cases we should be using cypi api's for loading the binary formats. If we; have columnar datasets and if you want to load those columnar data into numpy arrays we use numpy or routines functionality.

Then you have dataset dot load files that is used for loading the text files and this specifically useful for text data where the directory name is a label and each file is a sample index classification problem. So, dataset dot load underscore files are the api that should be used to load such kind of text datasets.

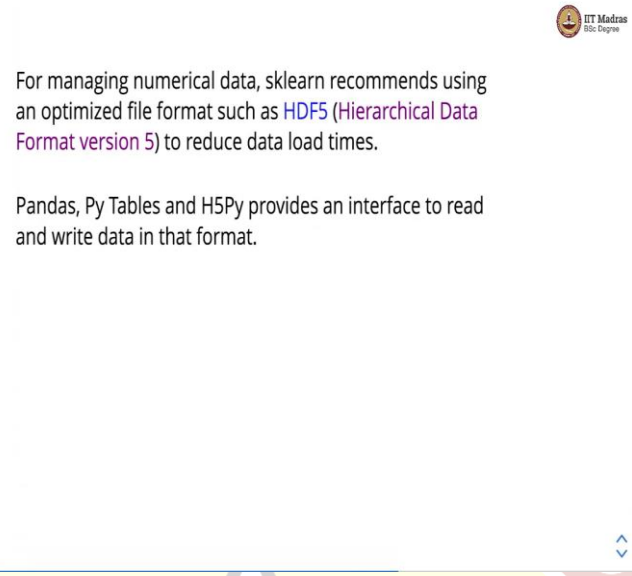
(Refer Slide Time: 11:41)



So, it could be learning support vector machine as one of the important classification technique and support vector machines generally work with very large feature space which is also spas. So, SVM uses some kind of a sparse format to store the training data and we have dataset dot loader underscore svmlight. We have datasets that load underscore svmlight underscore files api to load datasets in svmlight and lib svm format.

And apart from text and numerical data we might have data in form of images and videos or also in form of audios. And sklearn also provides functionality to load this variety of datasets. For example, skimage provides tools to load images and videos in numpy array and scipy has a wave file reader that that helps us to read a wave file into numpy array. So, we can use these various functionalities for loading dataset depending on the format of the training data.

(Refer Slide Time: 13:06)



So, if you are managing your own numerical data sklearn data recommends using an optimized file formats such HDF5. HDF5 stands for hierarchical data format version five and the HDF formatted dataset takes lesser time to load and you know pandas pi tuples and hfipy provides interfaces to read and write data in HDF5. So, that is it from the data loading functionality of sklearn.

We will implement some of these functions in colab where you will get an idea how to use this api for loading training data from different formats. I hope you enjoyed and learned how to load training data in sklearn learn from various formats. In the next video we will look at the demonstration of some of these concepts through colab, thank you, Namaste.