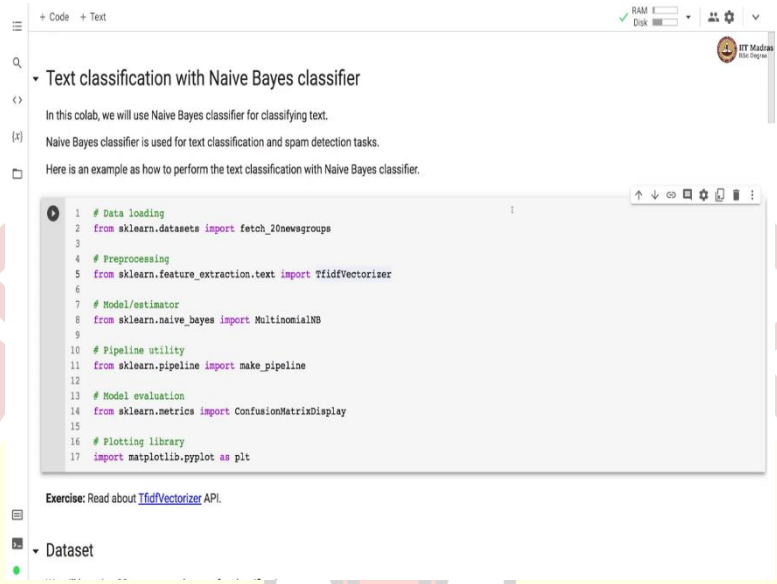
IIT Madras

ONLINE DEGREE

Machine Learning Techniques
Professor Doctor Ashish Tendulkar
Indian Institute of Technology Madras
Demonstration: Naive Bayes Classifier

(Refer Slide Time: 00:10)

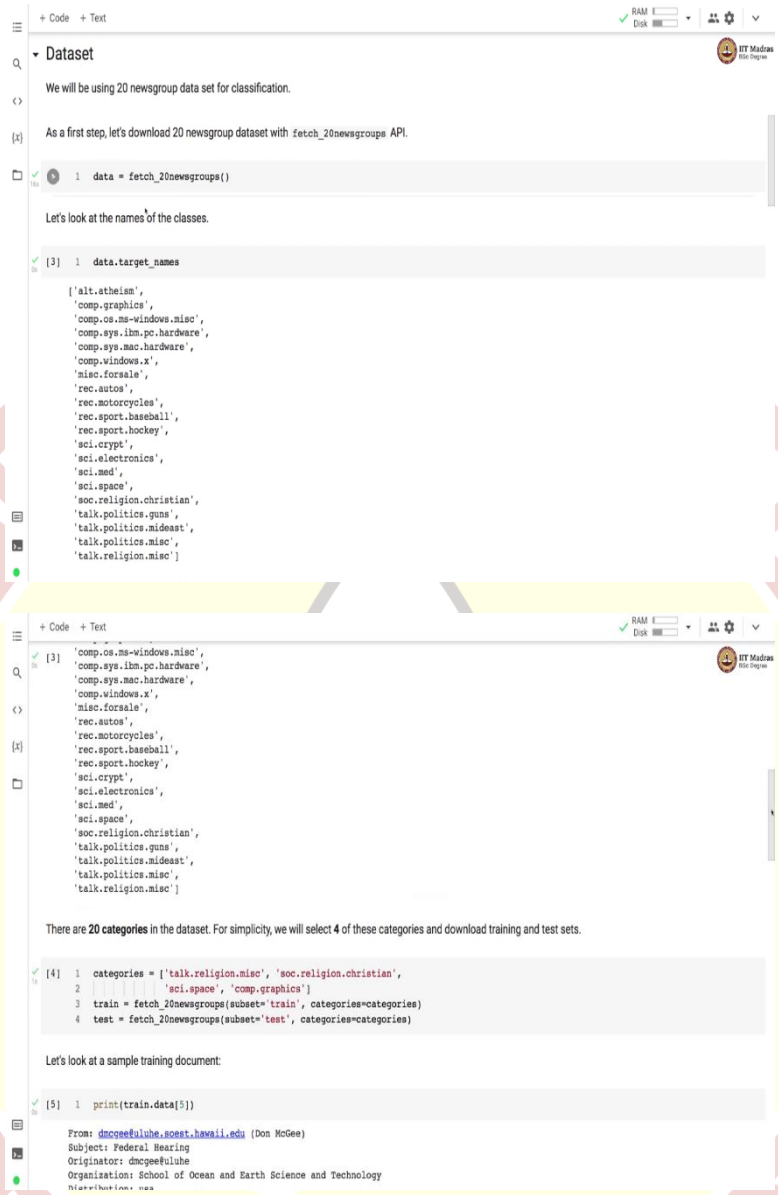


Namaste! Welcome to the next video of Machine Learning Practice Course. In this video, we will use Naive Bayes classifier for classifying text. As you know, Naive Bayes classifier is used for text classification, and spam detection tasks, here is an example as how to perform the text classification with Naive Bayes classifier.

Let us begin by importing necessary libraries. In this case, we will be using 20 newsgroup data, and we will fetch it with fetch _20newsgroup api from sklearn**,**datasets library. Then we will use TfidfVectorizer in order to convert the text document into a set of numbers, and TfidfVectorizer is part of sklearn**.**feature _extraction**.**text library.

Then we will be using multinomial Naive Bayes classifier that we that we load from sklearn**.** naive _bayes library. Then we have pipeline utility and confusion matrix display for model evaluation. We will be using matplotlib**.**pyplot for plotting the confusion matrix. So, I would urge all of you to read a bit more about TfidfVectorizer API in the sklearn user guide.

(Refer Slide Time: 01:51)

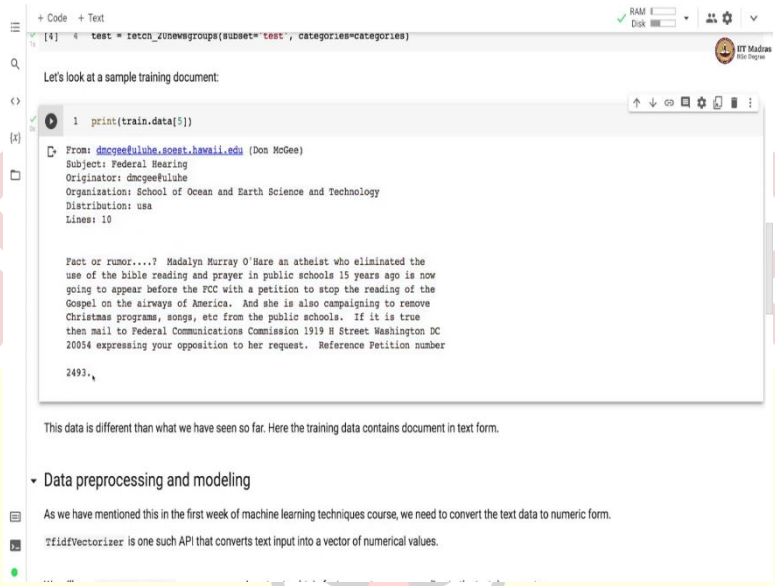


Here we will be using 20 newsgroup dataset for classification. As a first step, let us download the 20 newsgroup dataset with fetch 20 newsgroup API. So, we simply call fetch _20newsgroup and we get the data. Now, let us look at the names of the classes. So, data**.** target _names will give you names of different classes. So as name suggested, 20 newsgroup there are 20 categories in this dataset.

For simplicity, what we will do is we will select four of these categories at random and download training and test dataset corresponding to those four categories. So, we have selected four categories which are talk**.**religion**.**misc, soc**.**religion**.**Christian, sci**.**space, and comp**.**graphics.

These four categories we are selecting, and what we will do is we will fetch training data corresponding to these four categories using fetch _20newsgroup API and by specifying the categories. So, we are we are specifying to subset train and test. So, we get dataset for training, as well as for testing for these four categories.

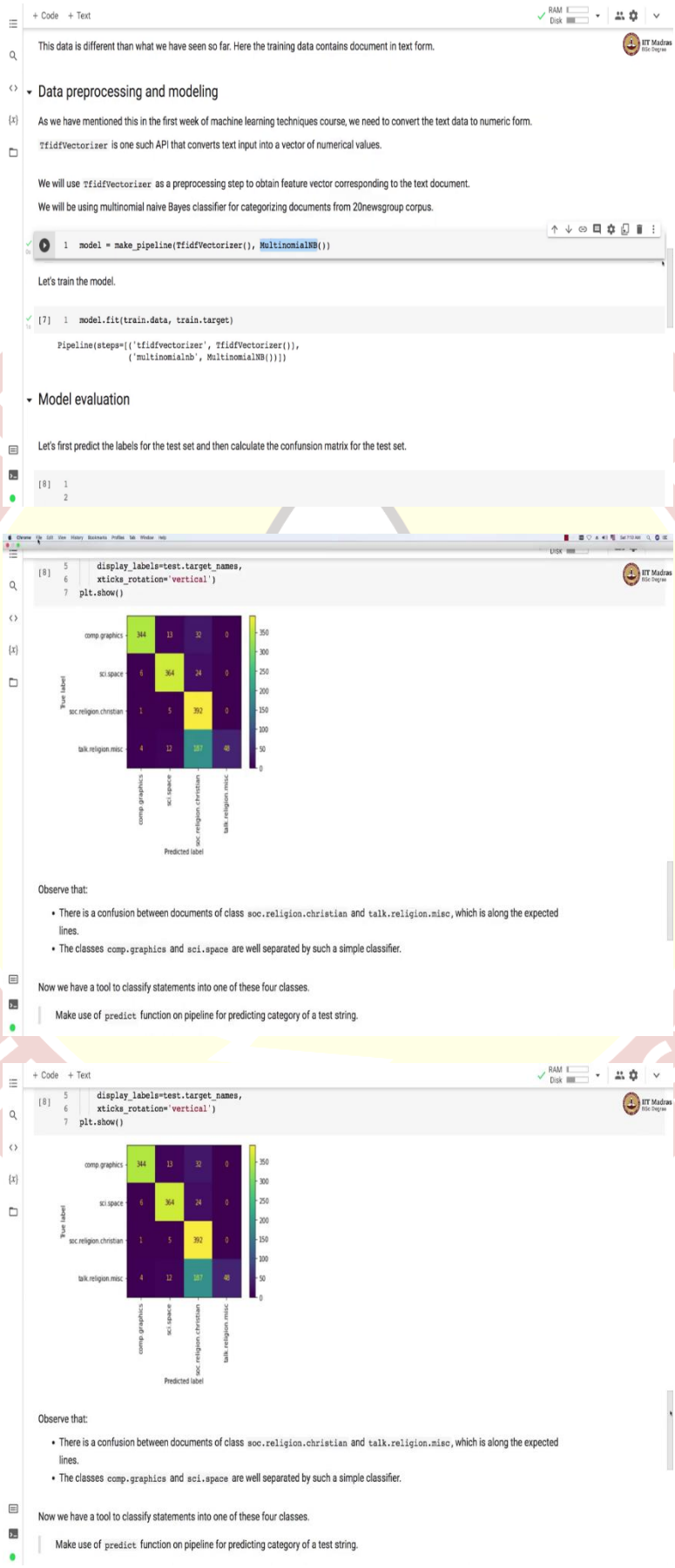
(Refer Slide Time: 03:17)



Let us look at a sample training document. So, we look at the sixth example in the training set. And you can see that it is in the form of a text. And this data is different what we have seen so far or what we are used to. Here the training data contains the text. So, this is one training example and it has got text. Then there is some kind of a header, and then there is some number in present in the document.

Now how do we handle it? So, in the first week of machine learning techniques course, we talked about different kinds of datasets like text and images. When we are presented with any non-numeric dataset, we need to perform some kind of pre-processing to get that data into numeric form. So here we will be using TfidfVectorizer to convert text input into a vector of numerical values.

(Refer Slide Time: 04:32)



So, what we do is, we define a pipeline object with two steps. The first step is TfidfVectorizer that obtains feature vector for a given text document, and then, we pass that feature vector through a multinomial Naive Bayes classifier for learning the classification model for these four categories. In order to train the model, we call the fit function on the pipeline object by passing the training data and the labels.

And once the model is learned, we use that to predict the label from test set and calculate the confusion matrix for the test set. So here we make use of confusion matrix display API and we use from _estimator method of this object of this class. So, now here, we pass the name, the object corresponding to the model.

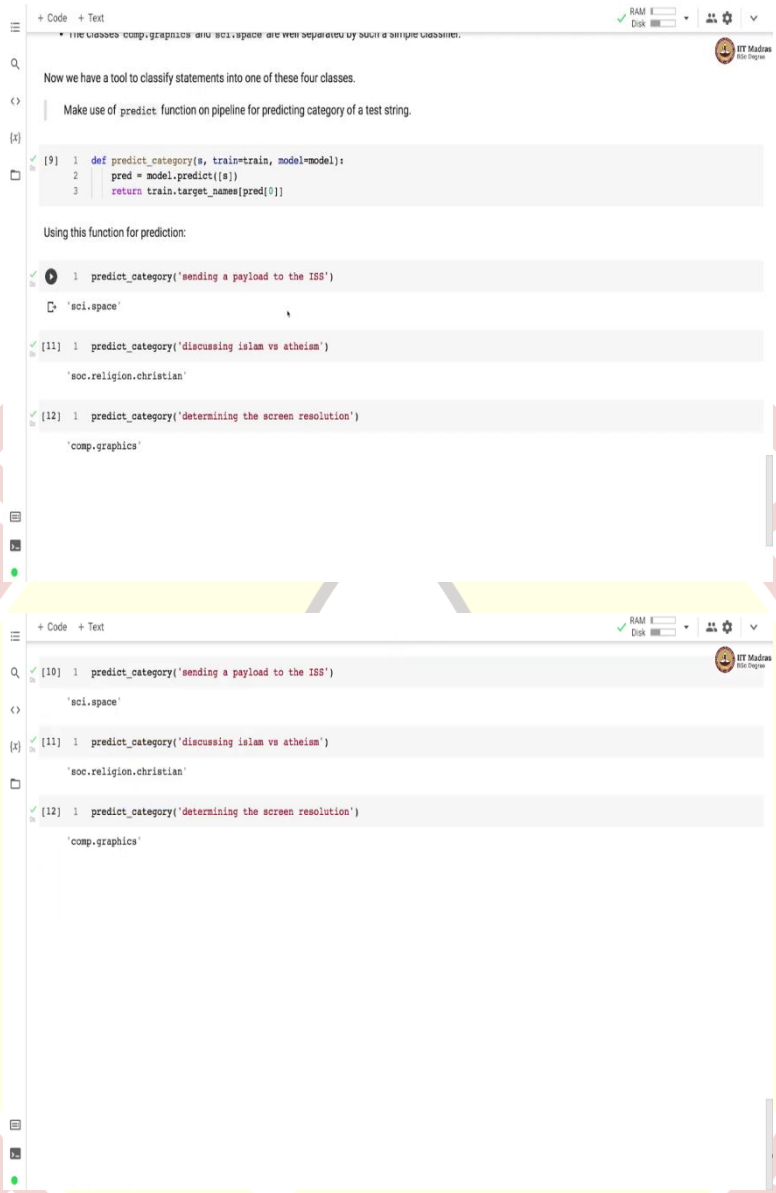
The test feature matrix test feature, test labels and then the names of the classes. And we also specify that on the x axis, the class name should be rotated so that they are in the vertical format. So, here on your screen, what you see is a confusion matrix between four different categories that we have selected out of 20 different categories in 20 newsgroup dataset.

So, what we see is, the com.graphics is classified quite well, because there is much less confusion with other classes, and so is the case with sci . space class. But if you observe the soc.religion.Christian and talk.religion.misc, there is some amount of confusion right. So, especially, for talk.religion.misc, it is getting confused with the members or with the example strong soc.religion.Christian.

So, soc.religion.Christian as such is again classified fairly well because the examples from this class are rarely misclassified. Only six examples are misclassified into the other classes. Whereas, in talk.religion.misc 187 examples or majority of examples are misclassified into soc.religion.Christian.

So, there is a confusion between these two classes, which is along the expected lines because they are talking about some religion-related content. The classes which are com.graphics and sci.space are well separated by such a simple classifier.

(Refer Slide Time: 08:00)



Now, we have a tool to classify statements into one of these four classes. So, for that we make use of predict function on the pipeline object for predicting category of the test string. So, here we define a small function called predict _category that take a statement, the training set and the name of the model.

And remember, we have set these two parameters to their default values. So, we first called the predict function on the model by using the statement as an argument, we get a prediction. And for this prediction, we find out what is the name of the class through train**.**target _names. So, let us use this function for prediction.

So, you want to predict category of this statement which is sending a payload to ISS which is International Space Station, and this is clearly you know, a statement from sci.space class. Then, we use another statement discussing Islam versus atheism. This is clearly related to religion, so that is why it is classified into religious content.

And the third statement, which is determining the screen resolution is part of com graphics. So, this is how the statements are classified into respective classes. So, in this video, we implemented Naive Bayes classifier in order to classify text into different categories. And we can use it in different tests classification task.