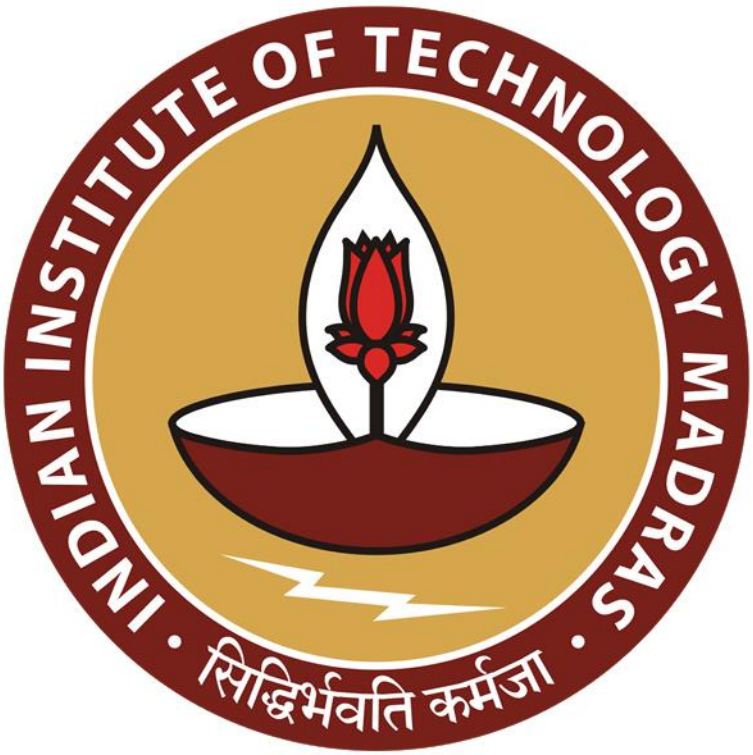
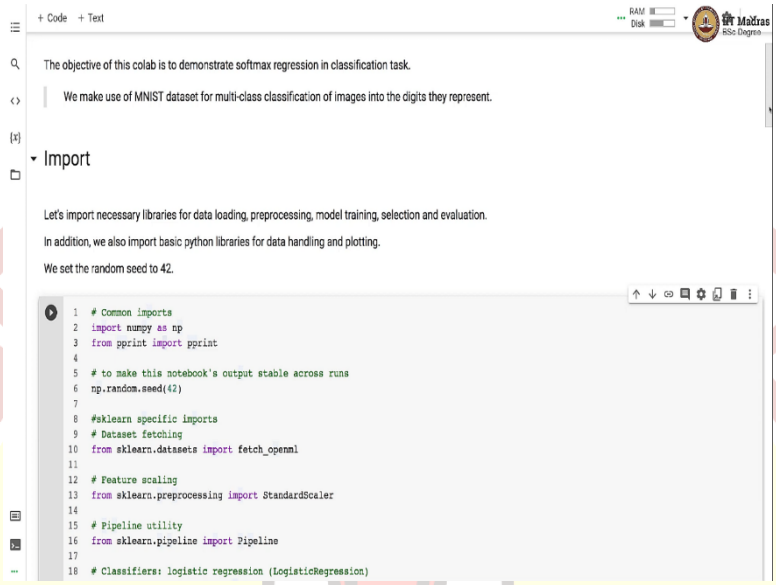


IIT Madras

ONLINE DEGREE

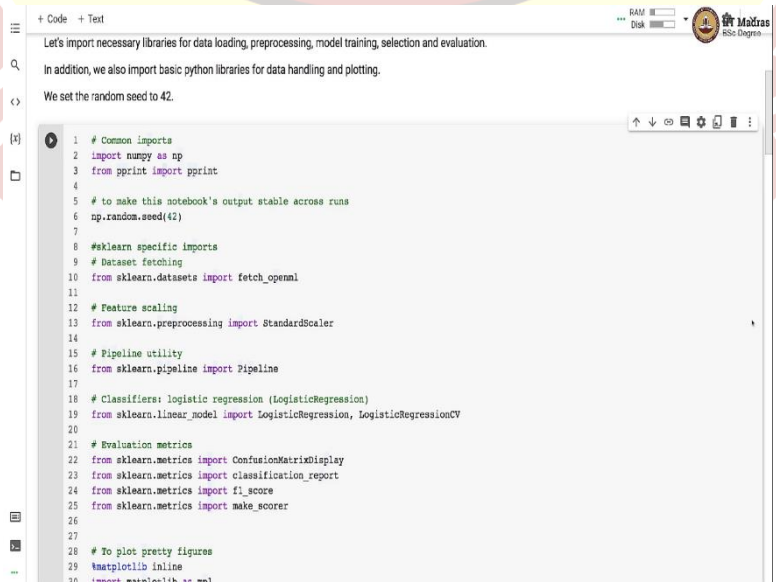
Machine Learning Program
Indian Institute of Technology, Madras
Demonstration - Softmax Regression with MNIST

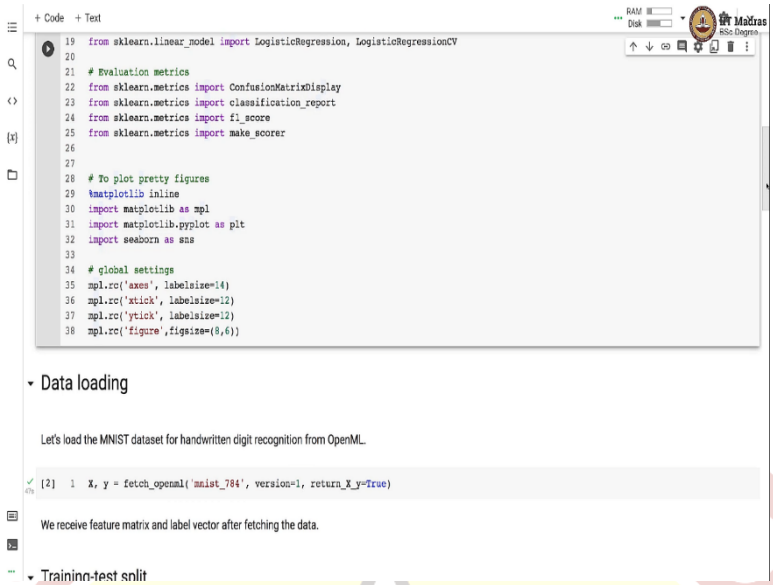
(Refer Slide Time: 00:10)



Namaste! welcome to the next video of Machine Learning Practice Course. In this video, we will demonstrate how to use softmax regression in the classification task. So, we make use of MNIST handwritten digit recognition dataset for the multi-class classification setup. So, as you know in MNIST data set there are 10 classes and the task is to classify an image of a digit to the digit that is present in the image.

(Refer Slide Time: 00:43)

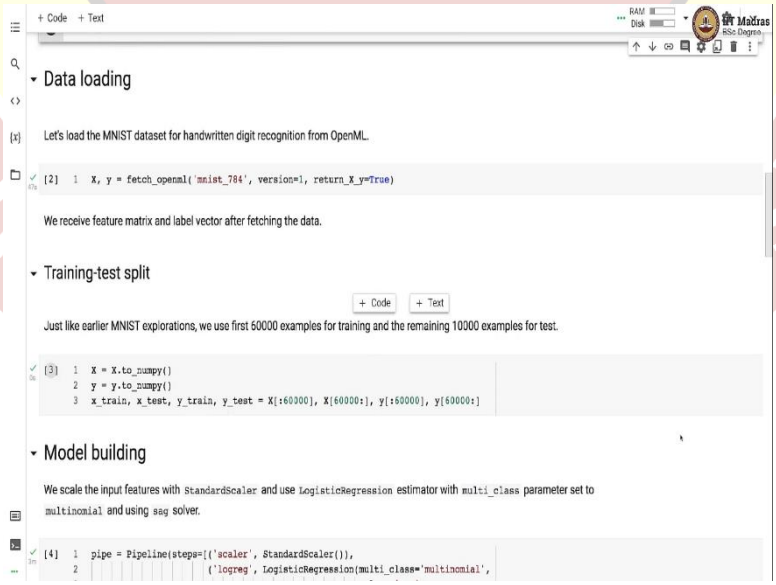




We import necessary libraries for data loading, pre-processing, model training, model selection and evaluation. We also import basic python libraries for data handling and plotting. Here, we import numpy then we import fetch _openml for loading the dataset, StandardScaler for pre-processing, Pipeline for setting up the pipeline of pre-processing and the model.

We will be using LogisticRegression and LogisticRegressionCV as estimators. We use ConfusionMatrixDisplay, classification report f1 _score and make _scorer from sklearn metrics for model evaluation. In order to plot figures, we use matplotlib and seaborn.

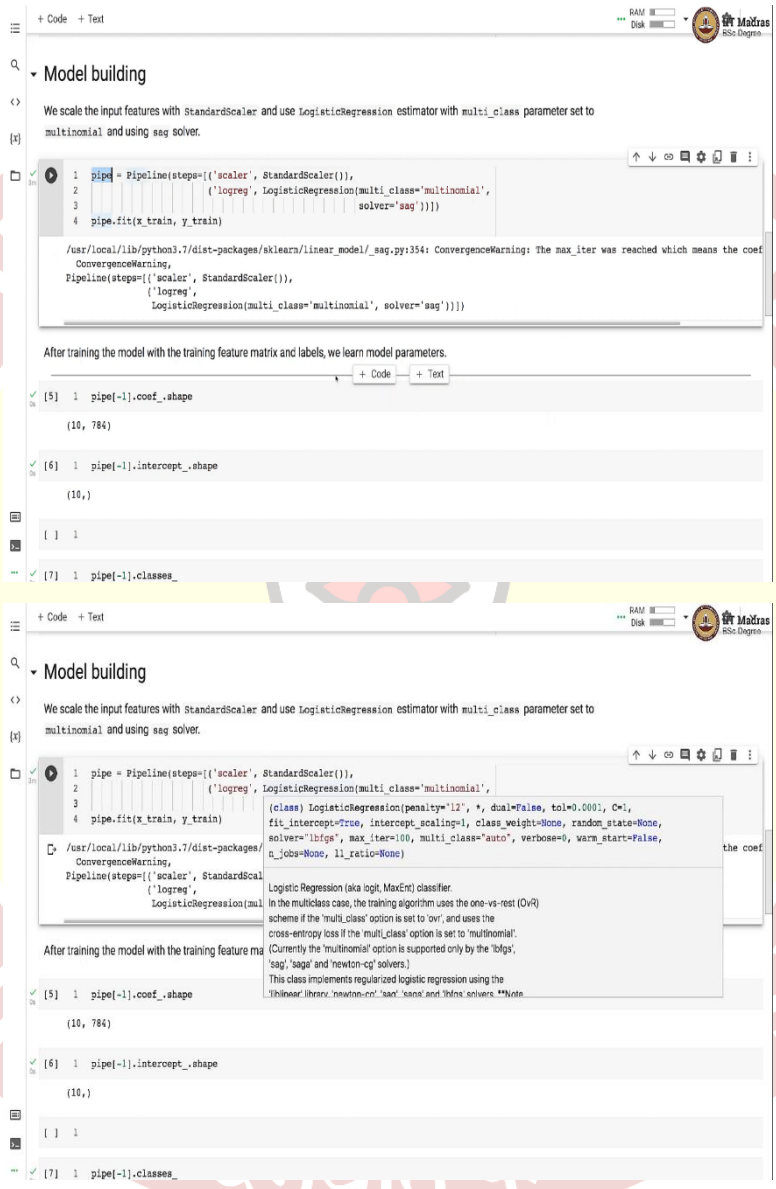
(Refer Slide Time: 01:34)



Let us load the MNIST dataset for handwritten digit recognition from openml. So, we use fetch _openml API. Specify the name of the dataset, which is mnist _784 and we also set the return _x _y flag to True. So, we receive feature matrix x and label vector y after fetching the data.

Just like earlier MNIST explorations, we use for 60,000 examples for training and the remaining 10,000 examples for test. So, we first convert the feature matrix and label vector to numpy and then we use the array indexing to split the data into training and test sets. We can also make use of train test split function over here, instead of the simple array indexing.

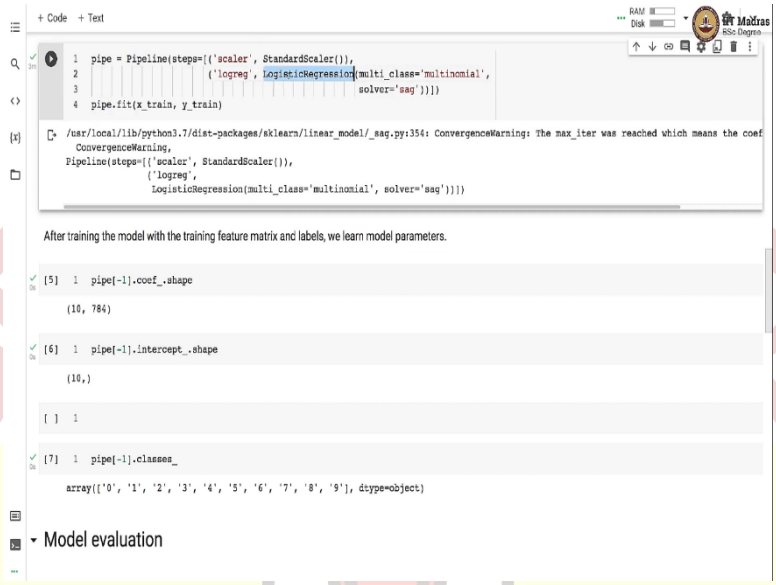
(Refer Slide Time: 02:30)



We scale the input features with StandardScaler and use LogisticRegression estimator with multi _class parameters set to multinomial and using sag as the solver. So, what is the difference between the LogisticRegression that we saw in earlier in this course. So, here, since we are solving the multi _class classification problem, instead of solving this problem in one versus rest or one versus one setup, we solve it directly with softmax regression and for that we have to set this multi _class parameter to multinomial.

So, when we set this to multinomial, the logistic regression uses softmax activation. So, we train the model by passing the train feature matrix and train label vector to the fit function of the pipeline object. And after the model is trained, we learn model parameters.

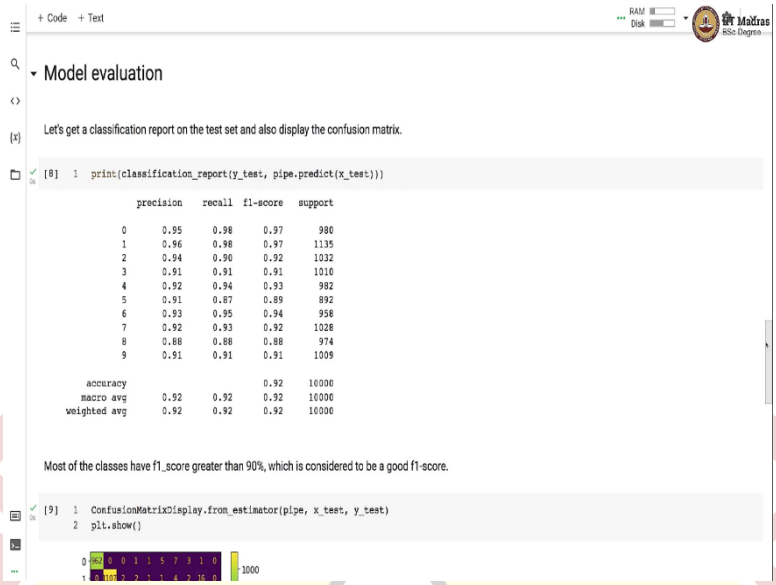
(Refer Slide Time: 03:34)



Let us look at the shapes of the model parameters. So, coefficients store the coefficients or the weights of the parameters. And we examine the shape of that and the shape is $\frac{10}{784}$ or 10 ,784. So, there are 10 classes and for each class there are 784 parameters. In the similar manner, we can examine the shape of the intercept. And remember, here we are using pipeline -1.

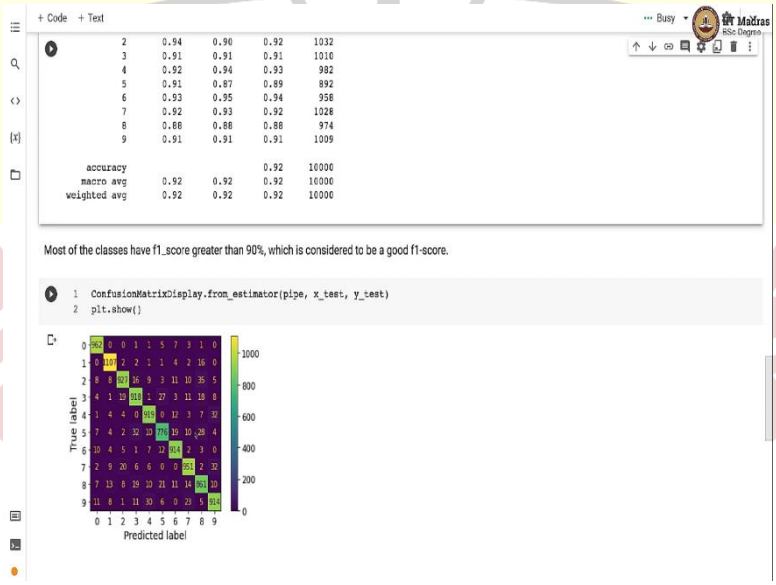
So, all these parameters that we are examining are from LogisticRegression because this is the last stage in the pipeline. And here you can see that the shape of the intercept of the intercept member variable is 10. So, this is a vector. So, for every class there is one intercept and we can see that there are 10 different classes that are present in the model, which are 0 to 9.

(Refer Slide Time: 04:39)



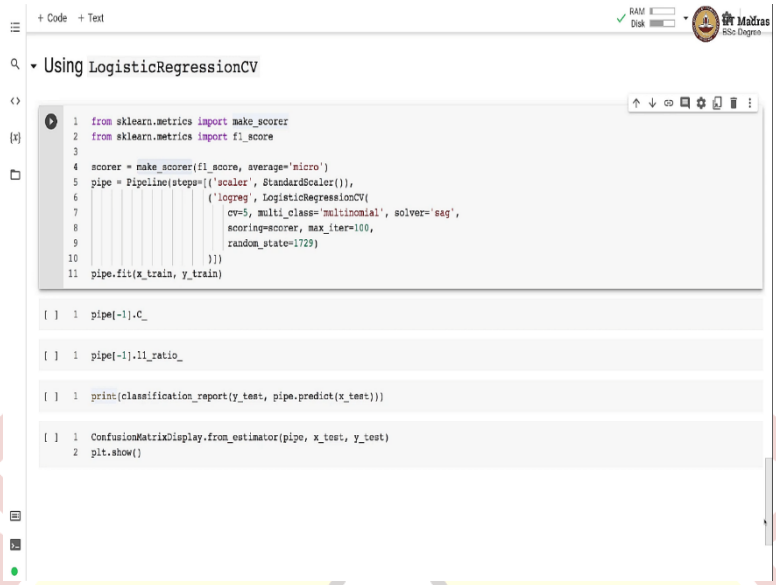
Now, that model is trained. Let us get a classification report on the test set and also display the confusion matrix. So, we get a classification report by passing the actual label of the test set and the predicted labels on the test set. So, most of the classes have f1-score > 90 and this f1-score is considered to be a good f1-score for any model.

(Refer Slide Time: 05:06)



And this is a confusion matrix for the test setup. So, here you can see that the label 3 is most confused with label 5. Label 8 is also most confused with label 5. Label 4 and 9 are also moderately or there is some confusion over there. Similar manner, there is a confusion between 5 and 3 or 5 and 8.

(Refer Slide Time: 05:46)



As an exercise, I would like you to use the LogisticRegressionCV in order to find out the optimal values of C and l1 _ratio for the LogisticRegression classifier. For that make use of make _scorer function from sklearn metrics and use the f1 _score for scoring with average equal to micro.

Use number of cross-validation fold equal to 5. Set the multi _class equal to multinomial and use the solver as sag. Then fit the LogisticRegressionCV model on the training feature matrix and label vector. After the model is trained, print the value of C and l1 _ratio that were obtained through to the LogisticRegressionCV.

Then obtain the classification report and examine the confusion matrix and also compare whether the LogisticRegressionCV gets better result than the plain logistic regression. In this video, we demonstrated how to use softmax regression in sklearn. We use MNIST handwritten digit classification dataset for this demonstration.