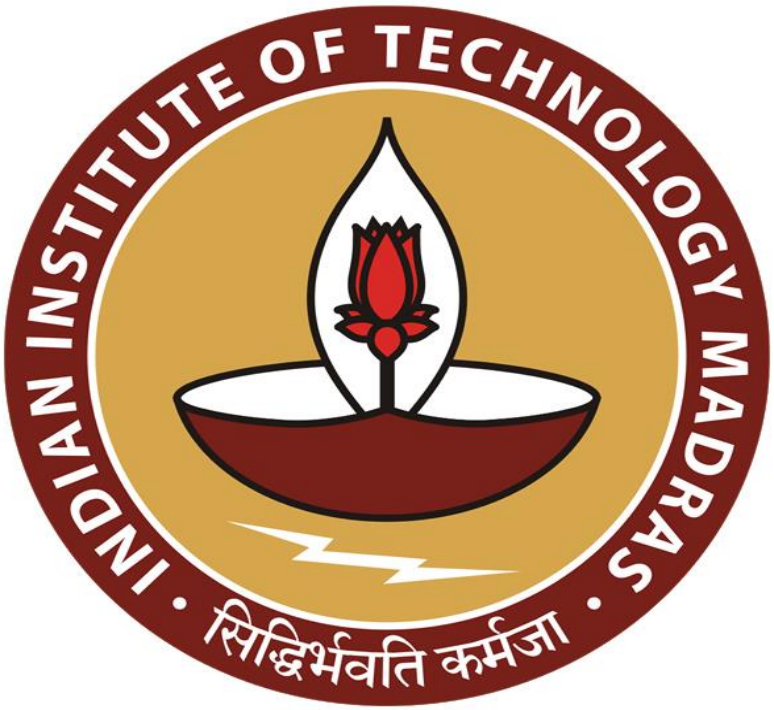


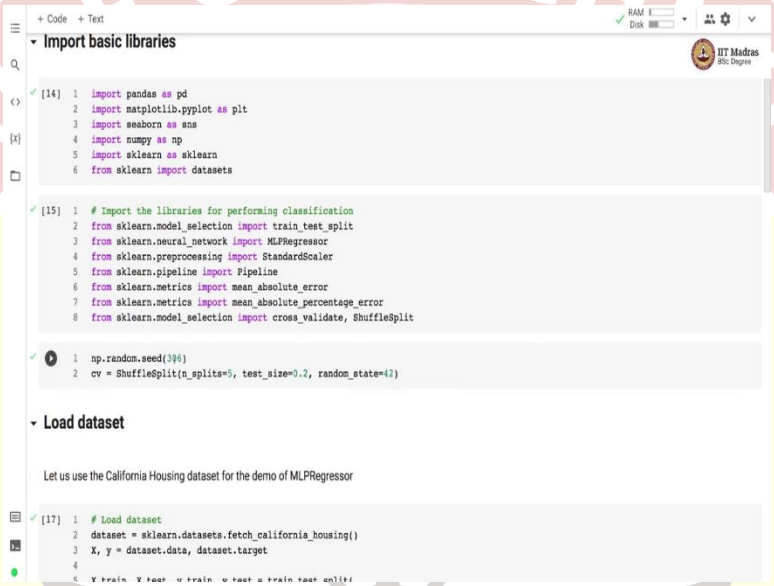
IIT Madras

ONLINE DEGREE

Machine Learning Practice
Bachelor of Science
Indian Institute of Technology, Madras
Multilayer perceptron regressor on California Housing Dataset

Namaste! Welcome to the next video of Machine Learning Practice Course. In this video, we will implement MLPRegressor on California housing dataset.

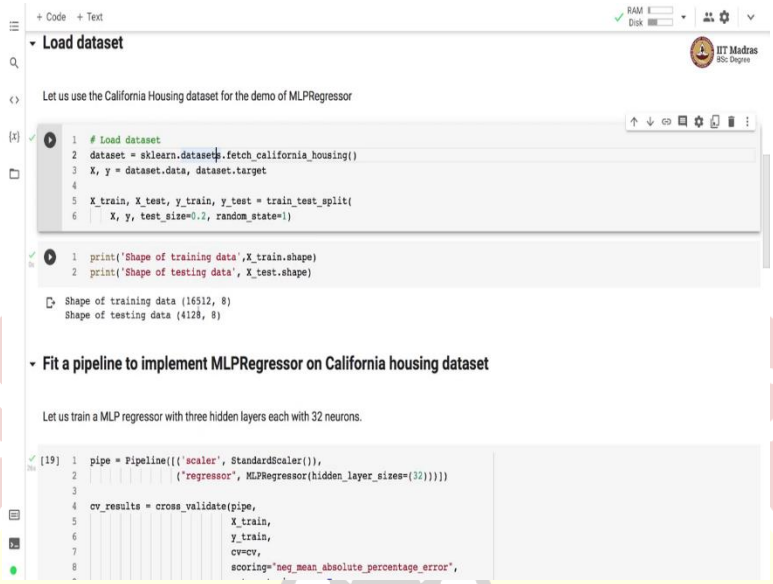
(Refer Time Slide: 00:25)



We begin by importing basic libraries like pandas and Numpy, and then matplotlib and Seaborn for plotting. We import datasets from sklearn. The training test split is carried out by train _test _split. The regression is carried out by MLPRegressor, which is implemented in sklearn**.**neural network module will perform feature processing like scaling using StandardScaler in sklearn**.**preprocessing module and the feature preprocessing and the regressor together are implemented in pipeline object, which is imported from sklearn**.**pipeline module.

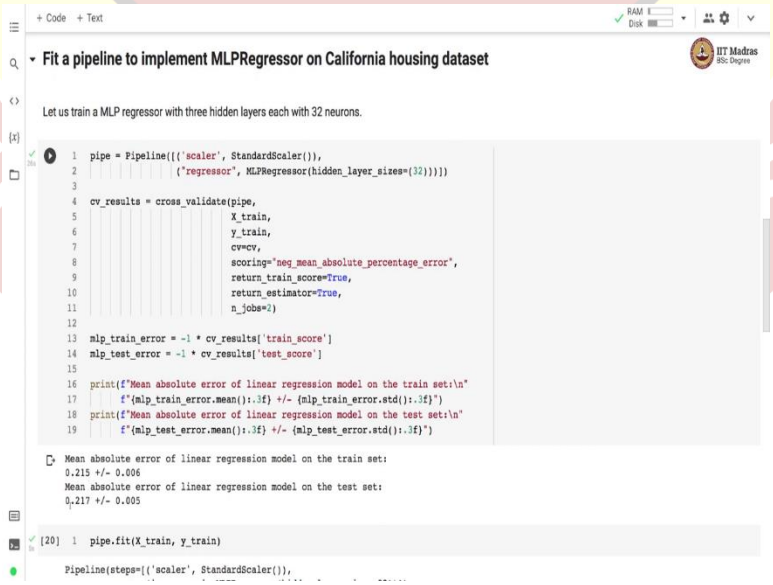
We evaluate the performance of classifier using mean _absolute _error and mean _absolute _percentage _error. And both these metrics are imported from sklearn**.**metrics module. We use cross _validate for training the classifier and ShuffleSplit for model selection. We begin by defining a random seed. We use ShuffleSplit cross-validation with number of splits = 5 and we set a size 20% examples as test examples.

(Refer Time Slide: 01:48)



We begin by loading the California Housing dataset as we have loaded earlier, and California dataset is loaded using fetch _California _housing from sklearn**.**datasets module. We set a size 20% examples as test examples to train test split. After training test split we obtained training and test feature metrics and training and test label vectors. You can see that 16,512 examples are used for training and 4,128 examples are used as test examples. Each example in training and test is represented with 8 features.

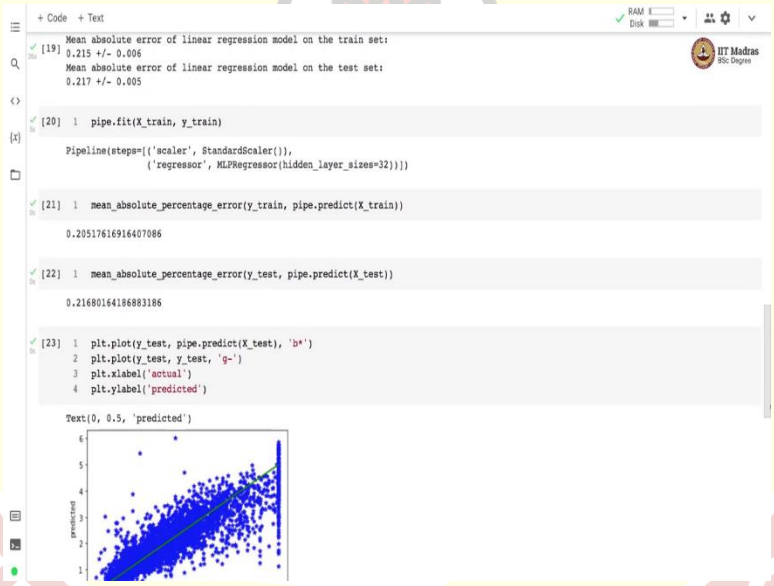
(Refer Time Slide: 02:31)



We define a pipeline object with feature preprocessing which is StandardScaler and then using regressor with MLPRegressor with hidden layers set to 32. So, we are defining MLPRegressor with 1 hidden layer and this hidden layer has 32 different neurons, we train the pipeline with cross-validation by supplying the training feature metrics and training labels. We use ShuffleSplit cross-validation strategy in the CV parameter and we use negative mean _absolute _percentage _error for scoring. We have set return train _score and little estimators to true.

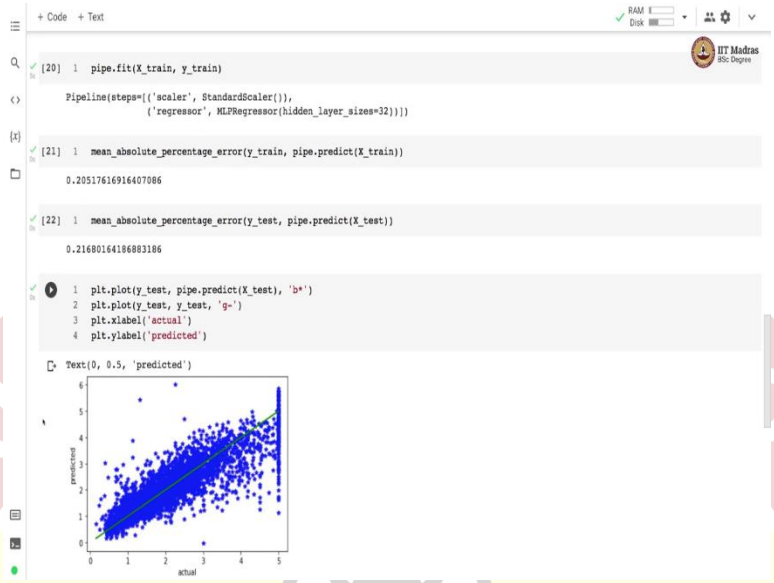
After training the model, we obtained mean _absolute _error on training set of 0.215 with a small standard deviation, the mean _absolute _percentage _error on the test set was 0.217. So, the mean _absolute _percentage _error is very similar on training and test set which indicates that the model is well trained and model is not underfitting or overfitting.

(Refer Time Slide: 03:49)



Then we obtain the mean _absolute _percentage _error on training and test set. So, you can see that on the training the mean _absolute _percentage _error is 0.205 And on test set it is 0.216.

(Refer Time Slide: 04:07)



We plotted the actual values and the predicted values. So, you can see that the actual value is represented with the green line which is at 45 degree and that is the line that shows the what should ideal prediction be. And you can see that the prediction is around this green line on either side. So, there is some error and which is which is what you see with mean _absolute _percentage _error.

So, this is the predicted values are represented with blue stars. So, there are some miss classification or there are there are many wrong predictions for this value 5, and you have seen this earlier that the actual value of 5. This is more like a truncated value for all the houses with value > 5 million and that explains why our regressor is not working well, in this particular region and the rest of the region the regressor is performing satisfactorily.

So, in this video, we trained MLPRegressor on California housing dataset and demonstrated how to use MLPRegressor in regression problems.