INDIAN INSTITUTE OF TECHNOLOGY MADRAS

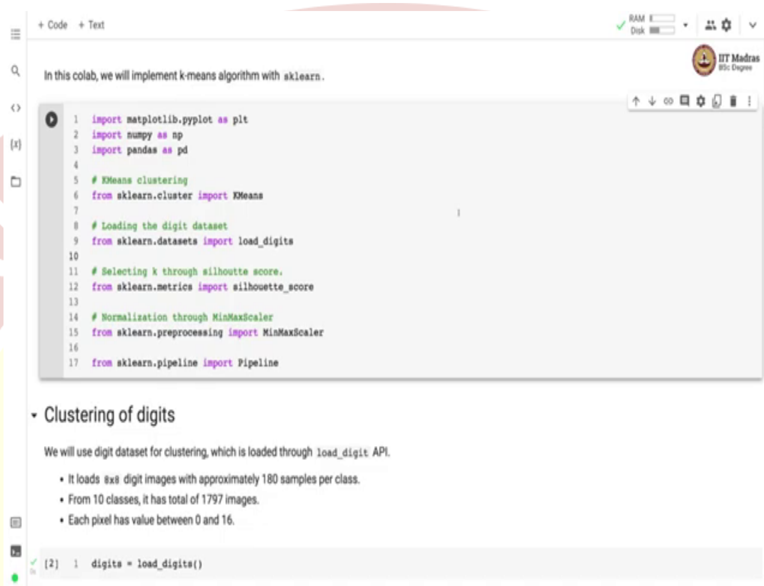
सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

Machine Learning Practice
Indian Institute of Technology, Madras
Programming and Data Science
K-means clustering on Digital Dataset

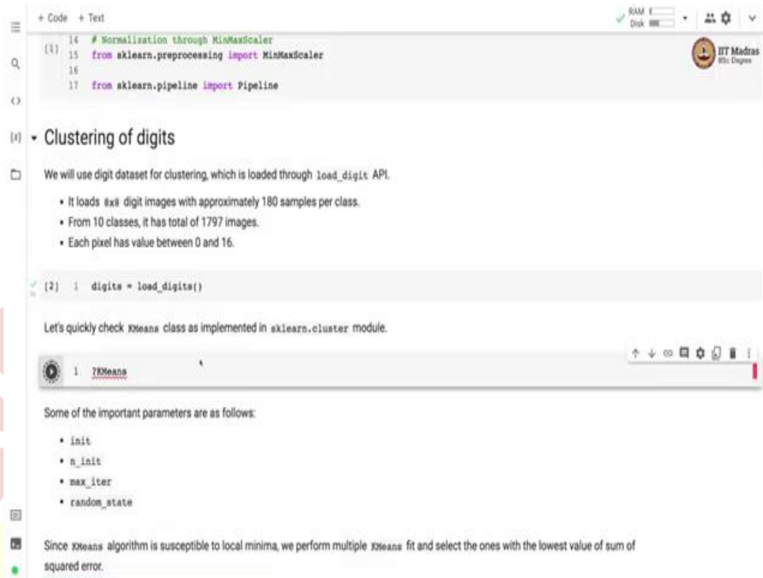
(Refer Slide Time: 00:10)

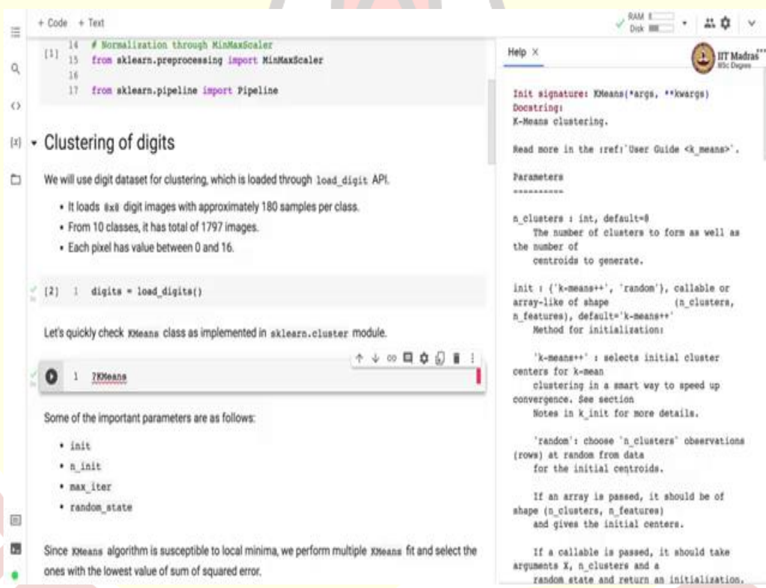


Namaste! Welcome to the next video of Machine Learning Practice Course. In this video we will implement k-means algorithm with a sklearn. The first import basic Python libraries like numpy and pandas and matplotlib.pyplot for plotting. K-means clustering is implemented as part of sklearn.cluster module. Then for this demonstration we use the digit dataset which is loaded to load _digit API from sklearn.datasets module.

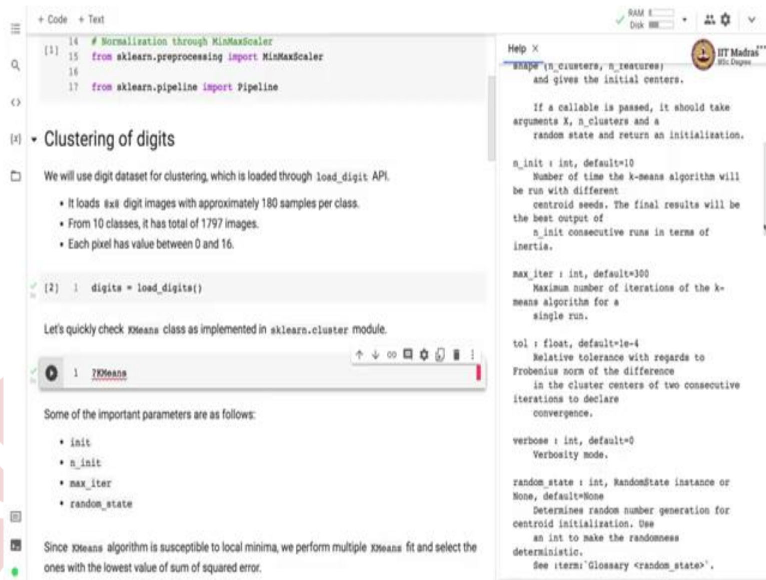
We will be selecting the optimal K through silhouette _score which is implemented as part of sklearn.metrics module and will normalize with MinMaxScaler. Finally, we use pipeline to combine the data pre-processing and clustering part.

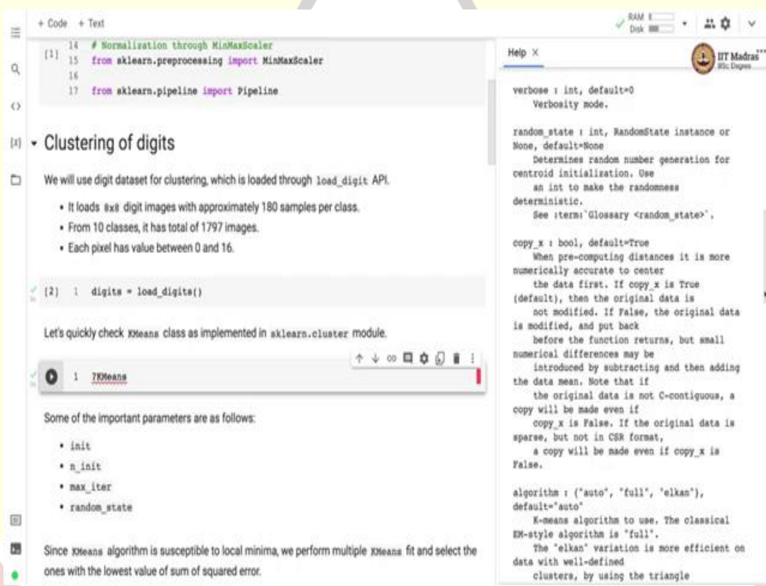
(Refer Slide Time: 01:01)





सिद्धिर्भवति कर्मजा





We will be using the digit dataset for clustering which is loaded through load _digit API. It loads 8 × 8 digit images with approximately 180 samples per class. There are 10 classes and it has got in all 1797 images. Each pixel has value between 0 and 16. Let us quickly check the k-means class as implemented at sklearn.cluster module. So, you can see the documentation of k-means class by calling k-means by prefixing it with the ?.

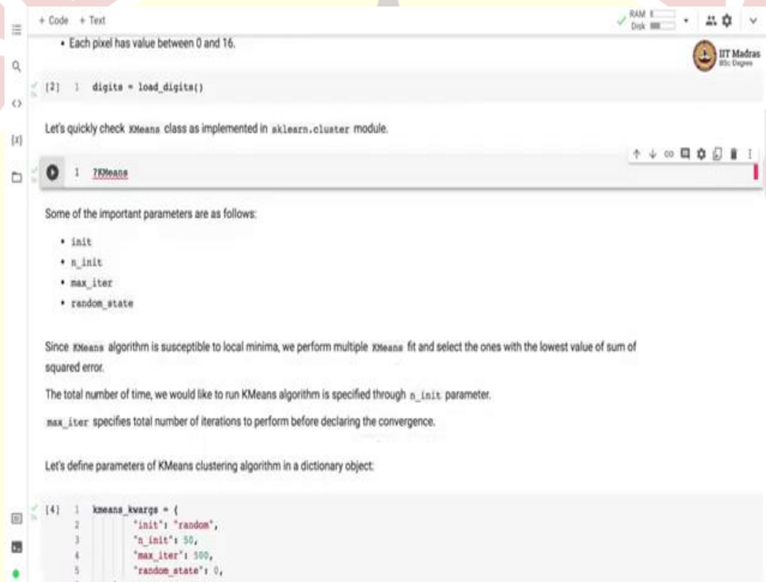
So, you can see that in k-means it takes parameters like number of clusters to form, then there is initialization parameter which could be k-means++ or random k-means++ selects initial cluster centers for k-means in a smart way to speed up the convergence, whereas random basically

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

initializes the cluster centers randomly. Then there is n _init parameter, which shows the number of times the k-means algorithm will be run with different centroid seats.

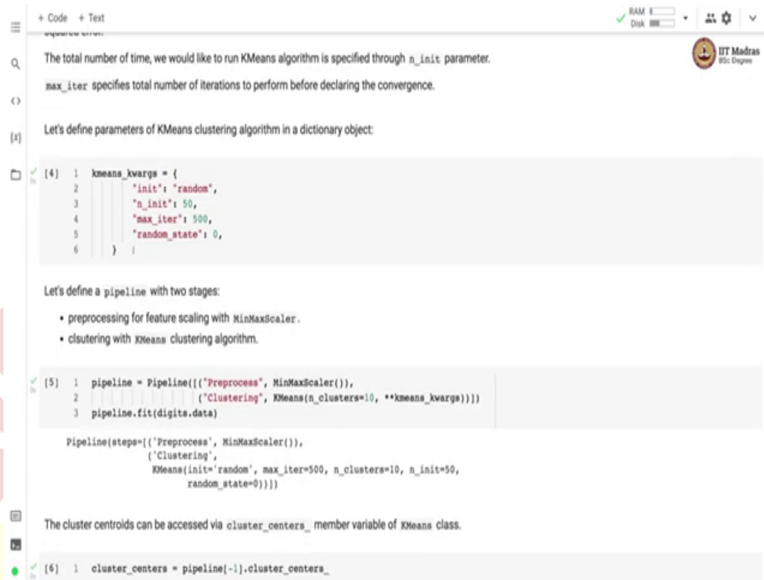
The final result will be the best output of the different runs in terms of the sum of squared error and max _iter specifies the maximum number of iterations that we want to run the single k-means algorithm for and finally, random _state that determines the random number generation for centroid initialization.

(Refer Slide Time: 02:40)



Since k-means algorithm is susceptible to local minima, we perform multiple k-means run and select the one with the lowest sum of squared error. The total number of time we would like to run k-means algorithm is specified through n _ init parameter. Max _ iter specifies the total number of iterations to perform before declaring convergence in any of these runs.

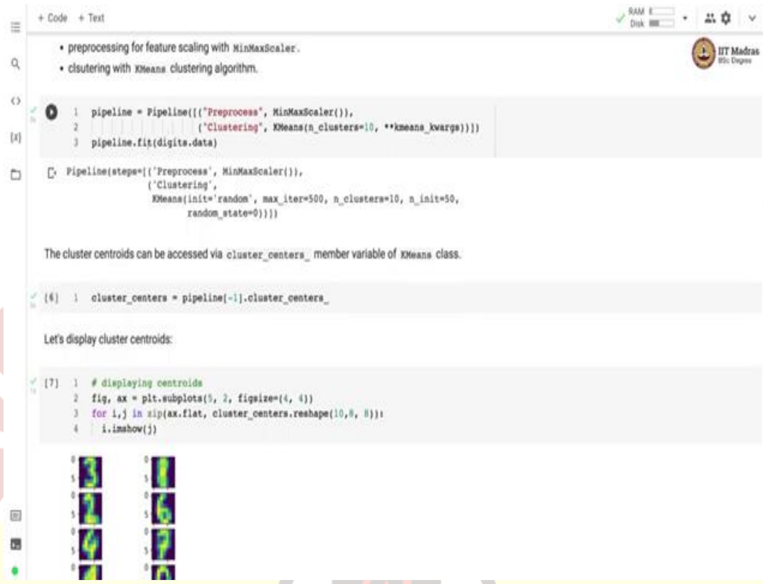
(Refer Slide Time: 03:04)



Let us first define all the k-means clustering algorithm parameter in a dictionary object. We will be using random initialization. We will be running k-means again, and again for 50 times, and in each iteration, we will be running k-means for 500 iterations, and random _state is set to 0. Let us define a pipeline with 2 stages. The first one is pre-processing for future scaling with MinMaxScaler. And second is clustering with k-means clustering algorithm.

Here we have selected number of clusters = 10. And we have specified the other arguments for k-means through this dictionary object. We call the fit function on pipeline with the digit dataset.

(Refer Slide Time: 03:52)



We can access the cluster centroids after training the k-means clustering algorithm. We can access them via cluster _centers_ member variable of k-means class. So, k-means class is available at the last stage of pipelines. That is why it is pipeline -1**.**cluster _centers_, give us the cluster centers. We display this cluster centroids in form of the digit images, as you can see over here. So, in this case, the number of clusters were known, hence, we set K = 10 and got these clusters.

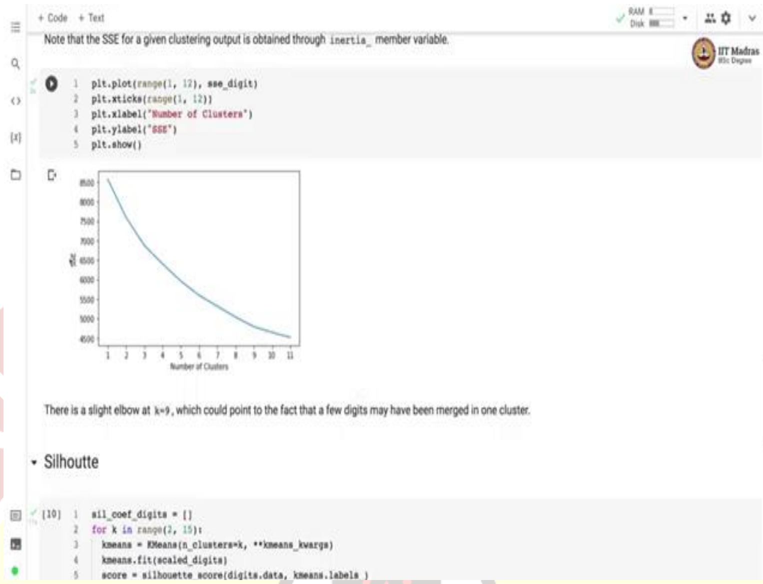
(Refer Slide Time: 04:39)

For deciding the optimal number of clusters, we generally use Elbow method and silhouette method. So, we basically trained the elbow and silhouette method and find out the actual number of K that comes out of these 2 methods. Here, for some time, we will pretend that we do not know the number of clusters and this will usually be the case when you are doing clustering in real life world, where we will not be knowing the total number of clusters, and we are supposed to come up with the optimal number of K.

And for that we use elbow and silhouette as 2 methods, so one of the 2 methods you need to employ. So, let us see how to use Elbow method. So, in Elbow method, what we do is we keep track of sum of squared error in a list. And here what we do is we first scale the digits by applying MinMaxScaler and this scale digits are used for clustering with k-means clustering algorithm, but number of clusters now varies in the range between 1-12.

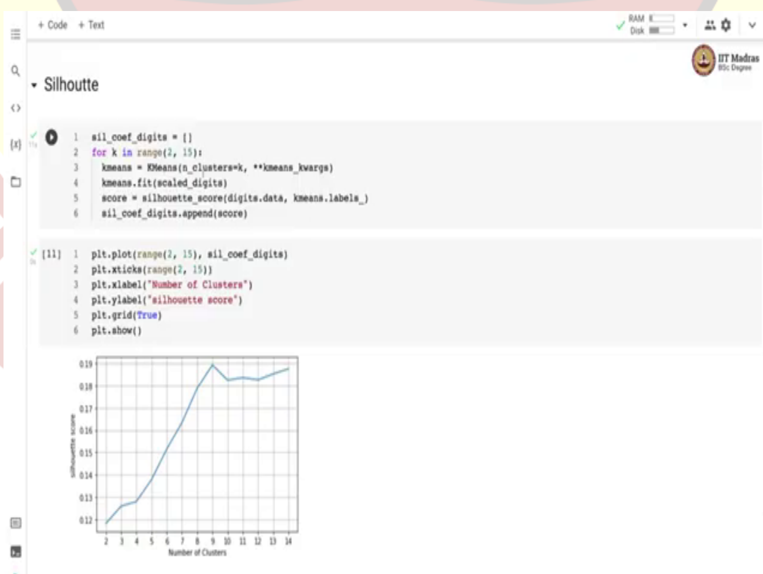
And the sum of squared error that we obtained for different values of K is appended to this particular list. And sum of squared error is obtained by accessing inertia_ member variable of the k-means object.

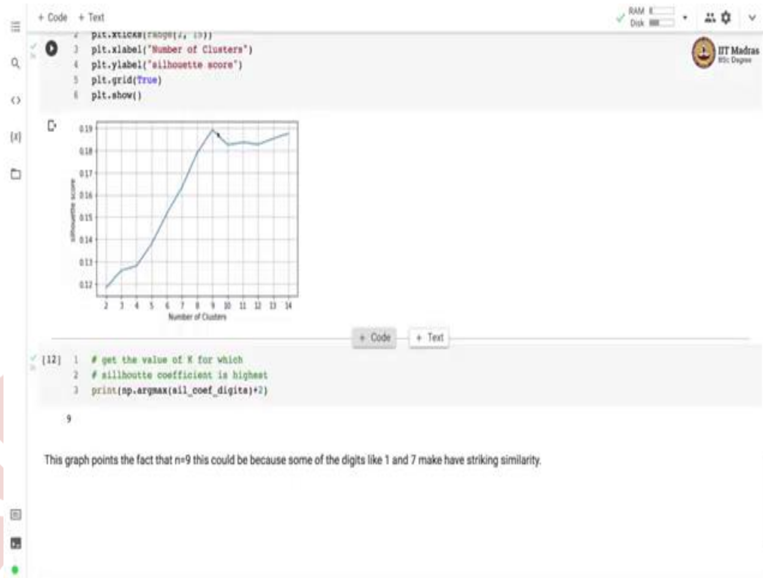
(Refer Slide Time: 06:01)



Now, you can see that, we have plotted the sum of square error against the number of clusters. So, we have number of clusters on x-axis and sum of square error on y-axis, and you can see that there is a slight elbow at K = 9. And this could point to the fact that few digits may have been merged in a single cluster.

(Refer Slide Time: 06:25)





The second method is a silhouette method and in silhouette method we basically calculate the silhouette _score based on the data and the labels that we obtained to k-means clustering algorithm. So, here again, we trained a k-means clustering algorithm with different number of clusters between in the range between 2-15 and then we calculate silhouette _score for each clustering solution. And the silhouette _score is appended to the list.

So, this graph also points the fact that K = 9, maybe the optimal number of clusters, which was also revealed by Elbow Method. And this could happen because some digits like 1 and 7 have striking similarity, and they have been merged in the single cluster. So, this was a short demonstration of how to use k-means clustering algorithm and find out optimal value of K through elbow and silhouette methods.