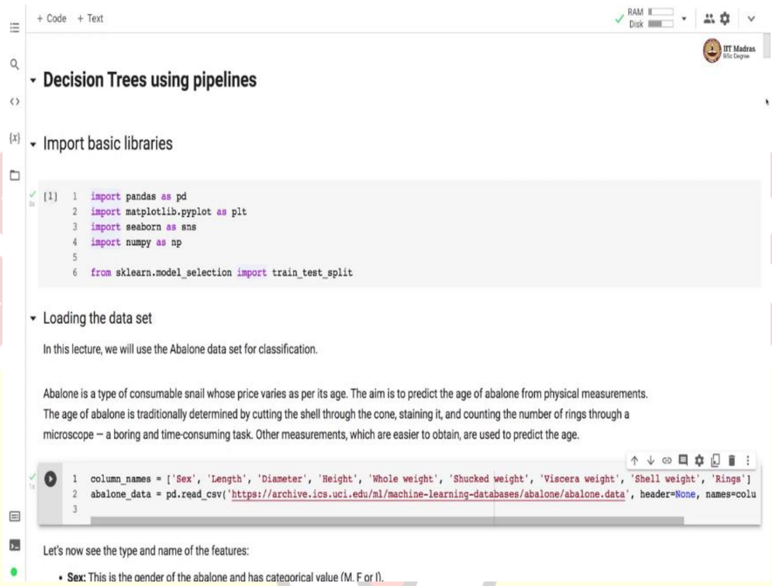
IIT Madras

ONLINE DEGREE

Machine Learning Practice
Professor Dr. Ashish Tendulkar
Indian Institute of Technology, Madras
Decision Trees for Classification – Abalone

(Refer Slide Time: 00:10)

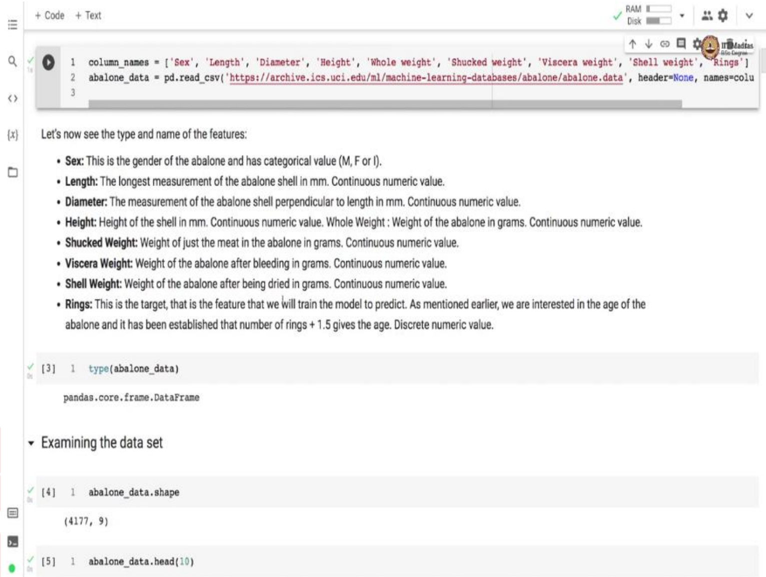


Namaste! Welcome to the next video of Machine Learning Practice Course. In this video, we will implement decision trees using pipeline. We import basic libraries like pandas. Then for plotting, we import matplotlib.pyplot and Seaborn. We also import NumPy. And then for model selection, we import train _test _split. In this video, we will be using Abalone data for classification.

So, Abalone is a type of consumable snail, whose price varies as per its age. Our aim is to predict the age of Abalone from the physical measurements. The age of Abalone is traditionally determined by cutting the shell through the cone, staining it and counting the number of rings through a microscope, which is a boring and time consuming task. Other measurements which are easier to obtain, can be used to predict the age of the snail, which is an alternative method.

And that is what we are going to explore in this collab using decision trees. So, in this dataset, we have columns like the sex, the length, diameter, height, whole weight, shucked weight, viscera weight, shell weight, and rings. We load the Abalone data from UCI machine learning repository. And this data is available in CSV format so we use pandas**.** read CSV function to read the data.

(Refer Slide Time: 02:00)



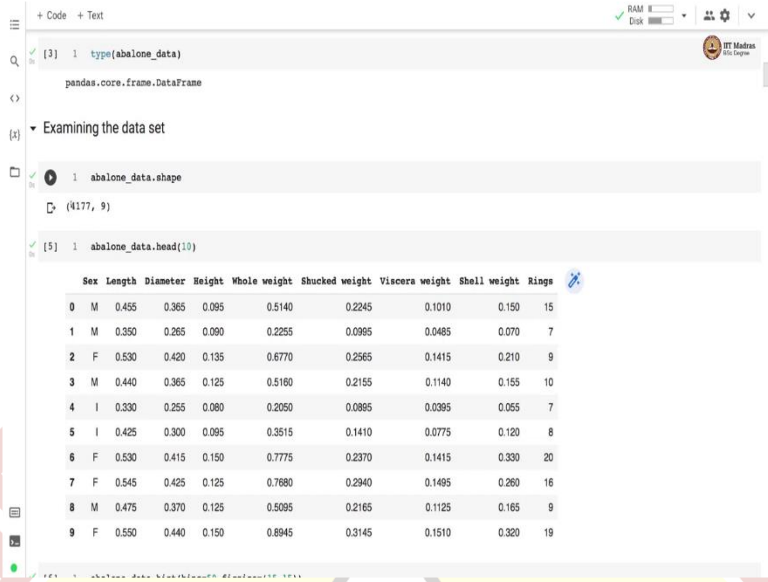
So, let us look at what each type of columns represent. So, we have sex that represent the gender of Abalone and has categorical values which are male, female, or infant. Then there is a length which is the longest measurement of the abalone shell in millimetre, it is continuous numerical value.

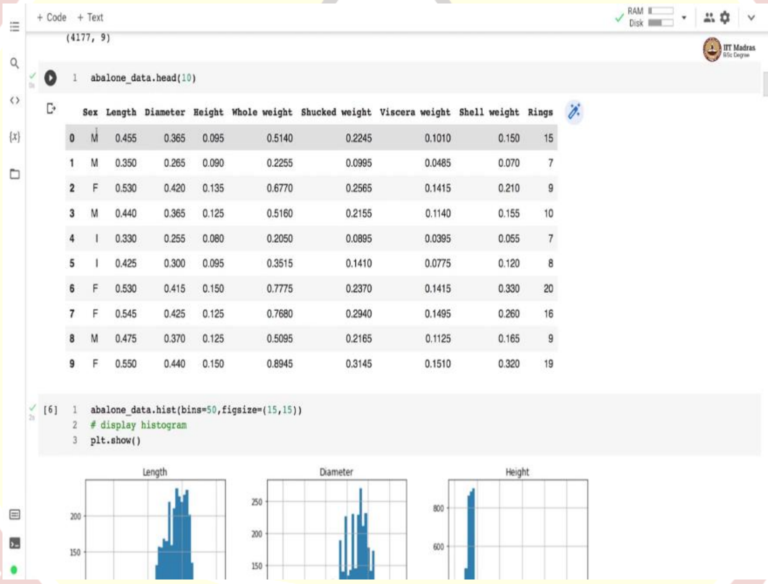
Then we have diameter which is the measurement of abalone shell perpendicular to the length in millimeter again a continuous numerical value than your height which is height of the shell in millimeter continuous numerical value. Whole weight is the weight of Abalone in gram which is again a continuous value.

Then there is a shucked weight which is weight of just meat in Abalone in grams. Viscera weight which is weight of abalone after bleeding in grams, then shell weight which is weight of Abalone after being dried in grams. And rings, this is the target, rings is a target variable for us. And we will train our model to predict the rings. And we are interested in age of Abalone.

And it is established that the number of rings + 1.5 gives the age of Abalone and it's a discrete numeric value. So, we have bunch of features which are in continuous, which are continuous numerical values. And our target is a discrete numerical value. So, even though the target is discrete numerical value, here, what we are going to do is we are going to treat this problem as a classification problem. And we will try to predict these number of rings for Abalone.

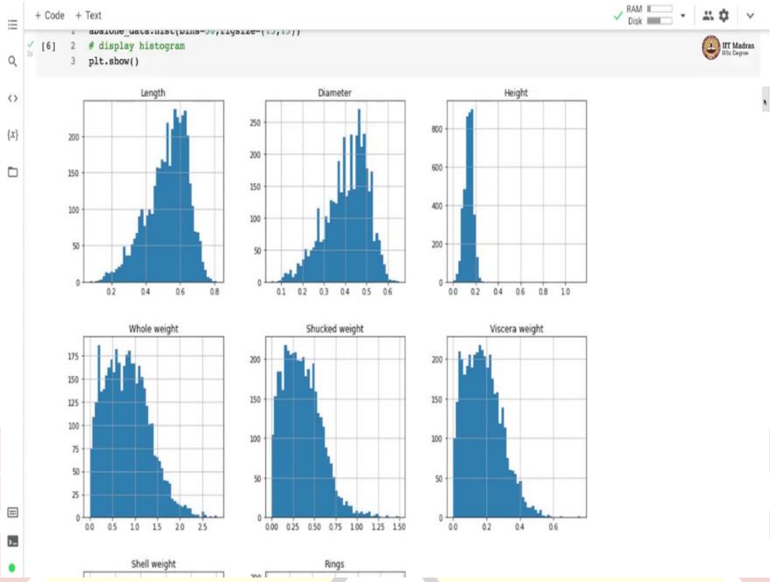
(Refer Slide Time: 03:57)

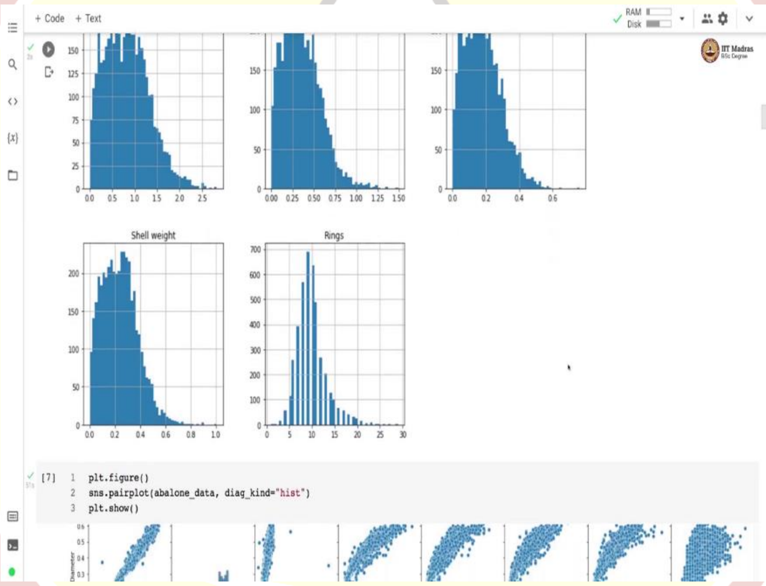




So, our data is in the format of the data frame. Let us examine the data. First thing to do is to look at the shape of the data. So, we have 4,177 examples in the dataset. And each example has got 9 features. Let us quickly examine the data you can see that the first 10 rows are printed on the screen. And you can check out the dataset or the first 10 sample rows on the screen.

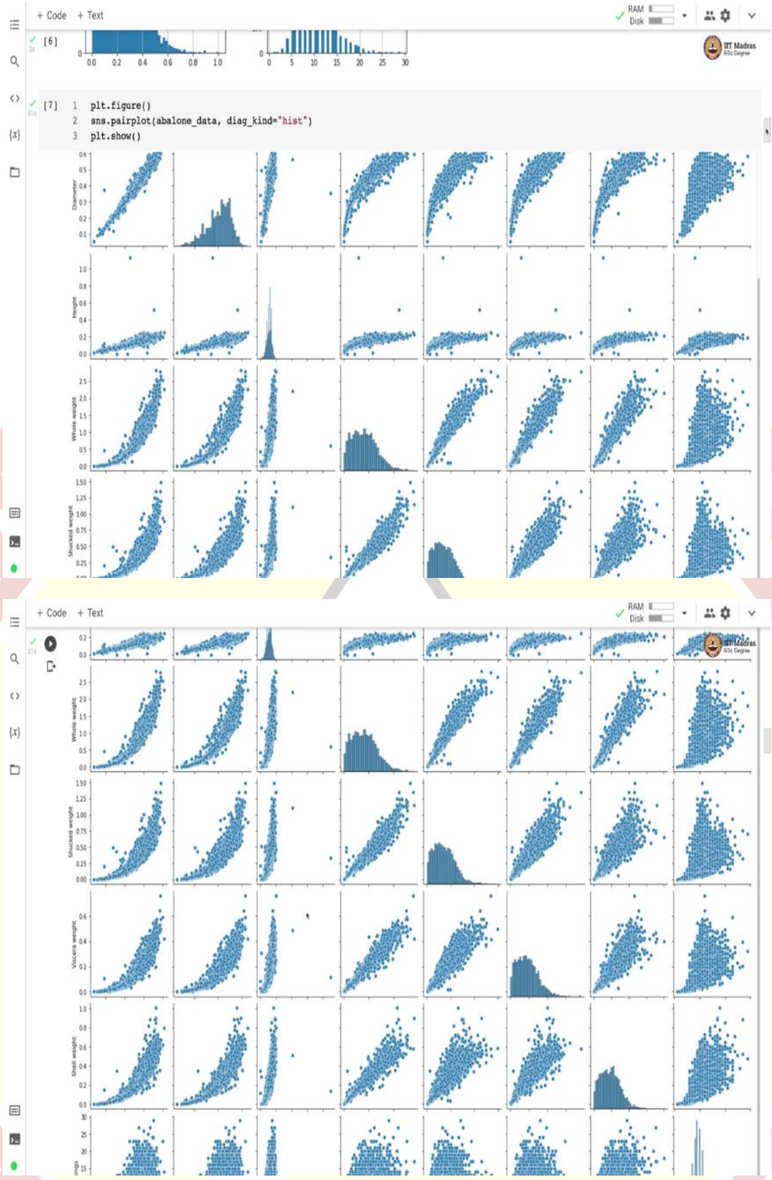
(Refer Slide Time: 04:34)





Next what we do is we plot the histogram of different features. And you can see that different features have got different distribution and different scales. And this is the distribution of rings which is our target variable. There are a lot of Abalone with the number of rings around 10. They seem to be quite abundant in this dataset, whereas, Abalone with rings > 15 are much smaller in this dataset

(Refer Slide Time: 05:21)

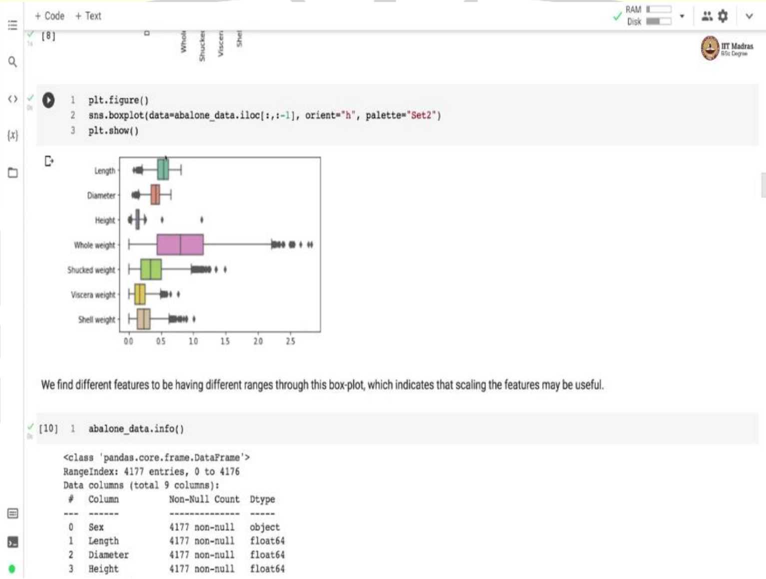


Then we perform pair plot between different features to see how these features are related to each other. And you can see that between certain features there seem to be very high correlation.

(Refer Slide Time: 05:38)

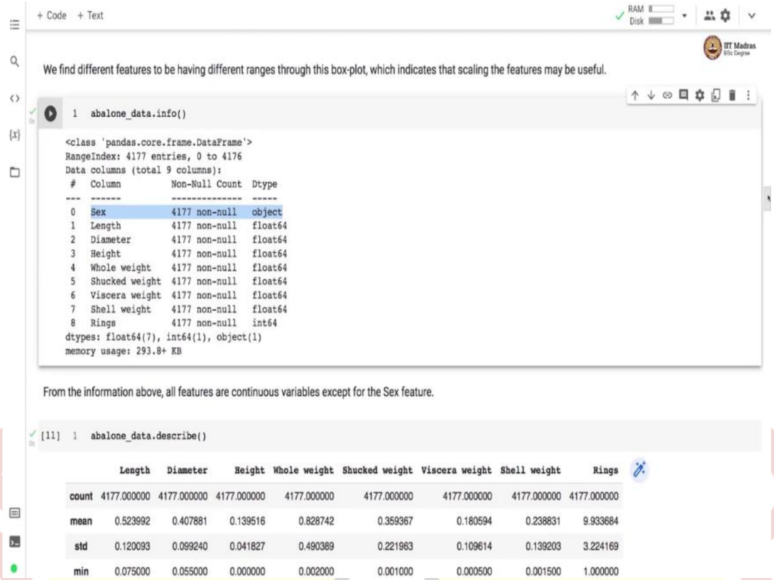
So, we can also establish correlation with heat map. And you can see that there is indeed a very strong correlation between certain features. For example, length and diameter there is a correlation of 0.99. Then between shucked weight and whole weight there is a correlation of 0.97. So, there is a lot of correlation between many features in this dataset.

(Refer Slide Time: 06:18)



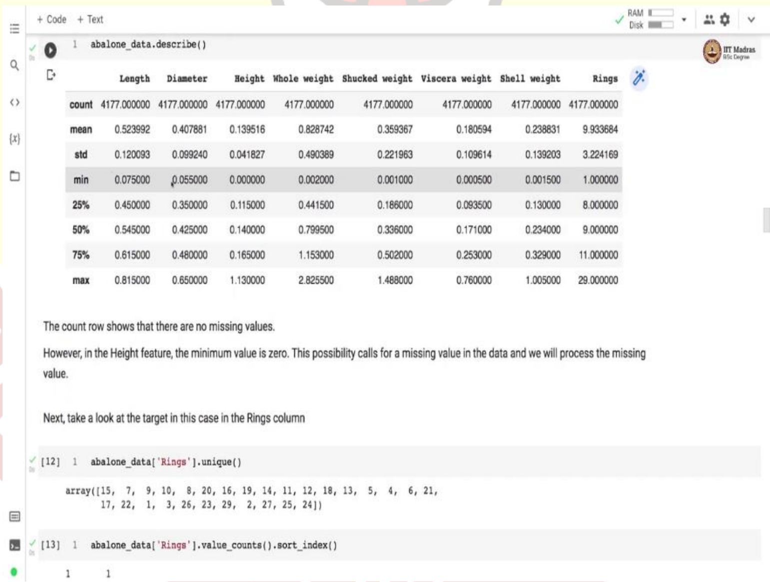
Then we plot the distribution of different features in slightly different format, this time in a boxplot format. And we can see that, again different features have different ranges. And there are certain outliers as well. So, for example, in whole weight there are a bunch of outliers in shucked weight. So, you can see presence of outliers in each of these features.

(Refer Slide Time: 06:49)



So, we have already seen that in the dataset there are 4,177 entries, and there are 9 columns including the target variable which is rings for us. And all the, so most of the features are continuous numerical values. And 6 is one feature which is a categorical feature.

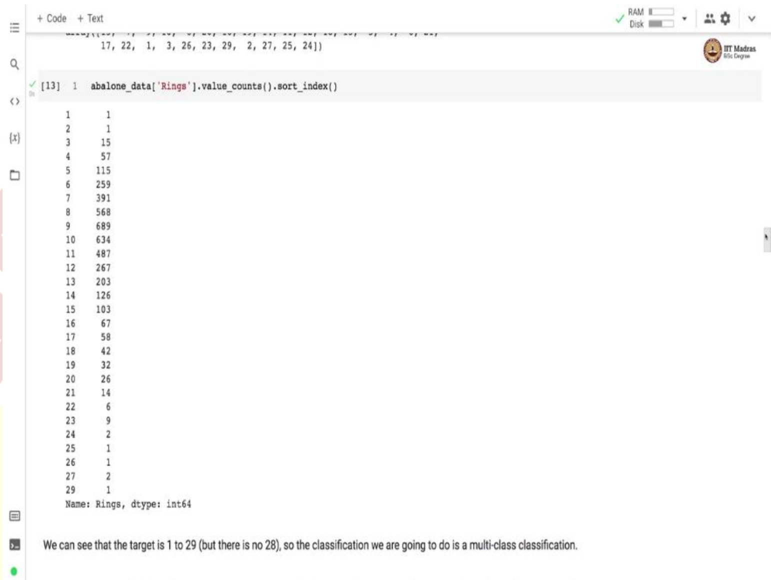
(Refer Slide Time: 07:17)



When we call describe on the data frame, and obtain different quartiles, we obtain mean, minimum, maximum and 25%, 50% and 75%. And we can examine this to check or reconfirm the presence of outliers. So, you can see here that the max over here is far higher than the 75^{th}%, there are outliers in height, there are outliers in shucked weight, then also in shell weight.

And there are certain outliers probably on the other side where the values are much smaller than the 25th%. And you can see that in height the minimum value is 0. So, there is a possibility of a missing value in the data. And we need a special provision for processing these missing values.

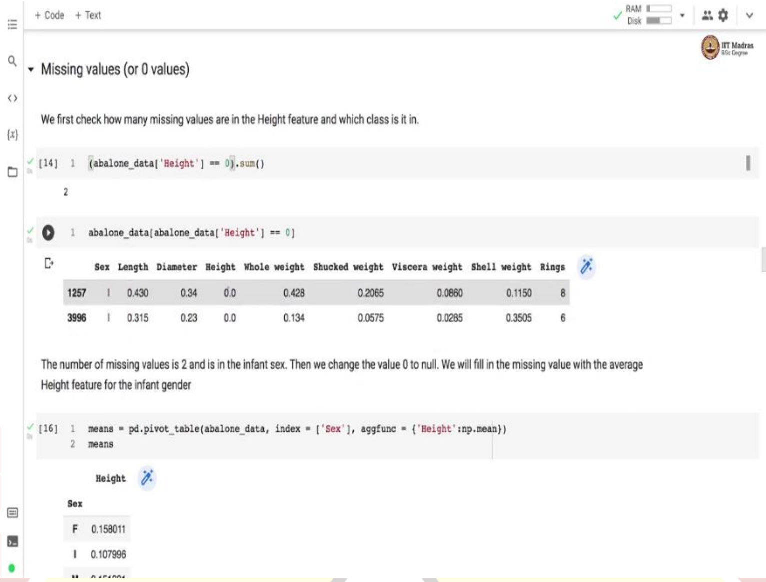
(Refer Slide Time: 08:16)



Then, let us look at a target column in this case, which is rings, and we count the number of rings. And we have seen this already in the plot that there are rings with count 10 or around 10, they form some kind of a majority in the class, but beyond 15 the number of rings are much lesser.

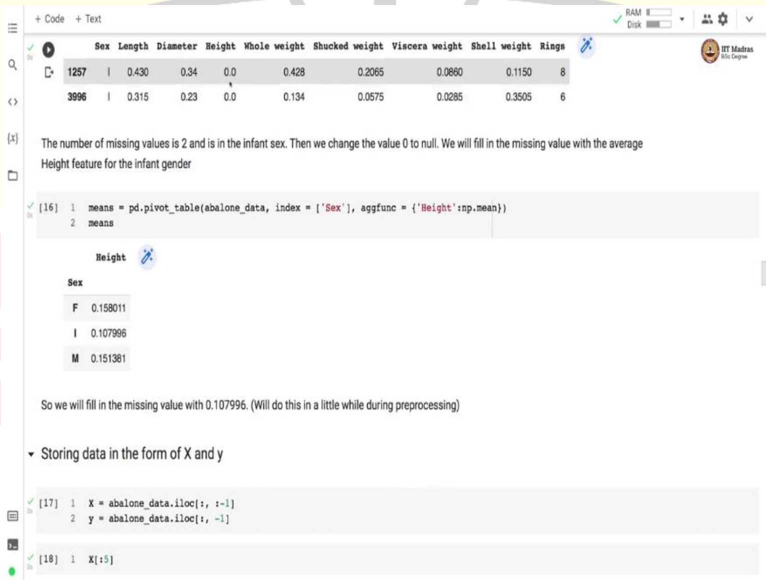
And here there is there are no rings present with number 28. So, there is another observation that you can make from this count. So, we are going to attempt a multi-class classification on this where there are 29 different classes and we will predict one of these 29 classes based on physical measurement of the Abalone.

(Refer Slide Time: 09:17)



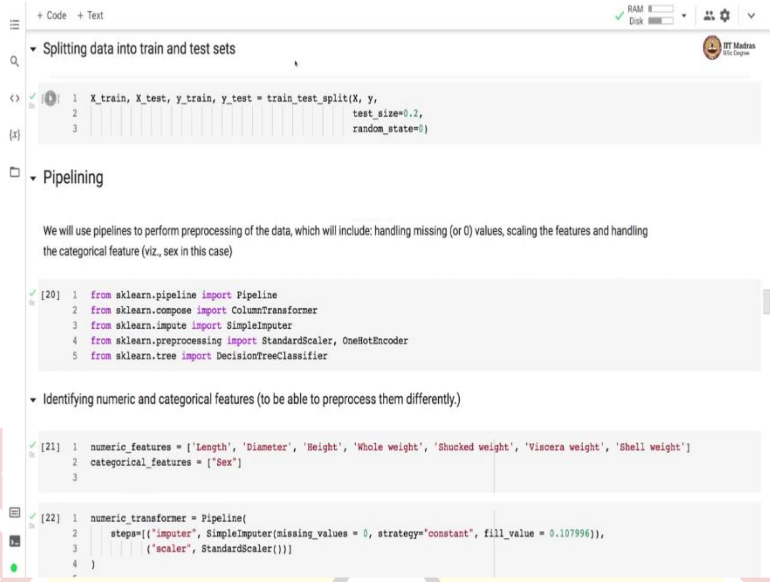
So, since there are missing values, we first handled the missing values, we found that in Abalone height column there are missing values and those missing values seem to be for 6 = infant and there are two missing values that you see here. So, we can fill up these missing values with average height feature of the infant gender.

(Refer Slide Time: 09:46)



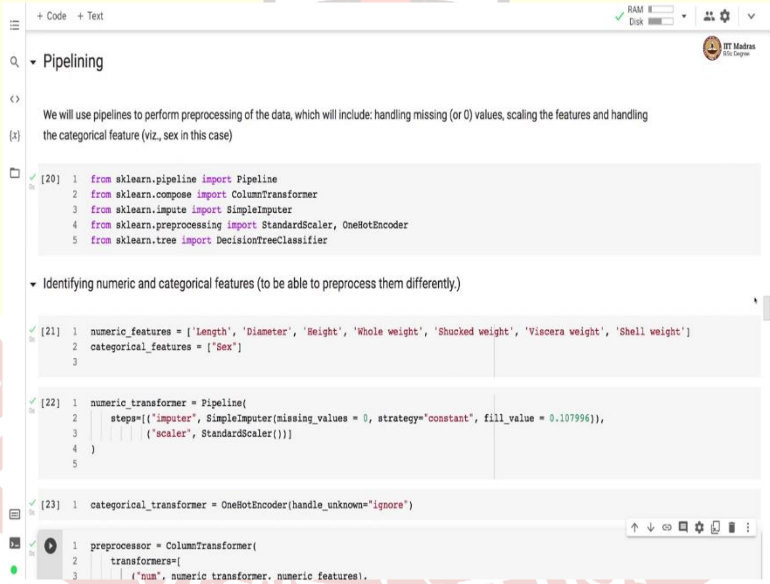
So, here we calculate the mean height for different genders. And since there are missing values for gender I, we use its mean value to fill up these missing values.

(Refer Slide Time: 10:04)



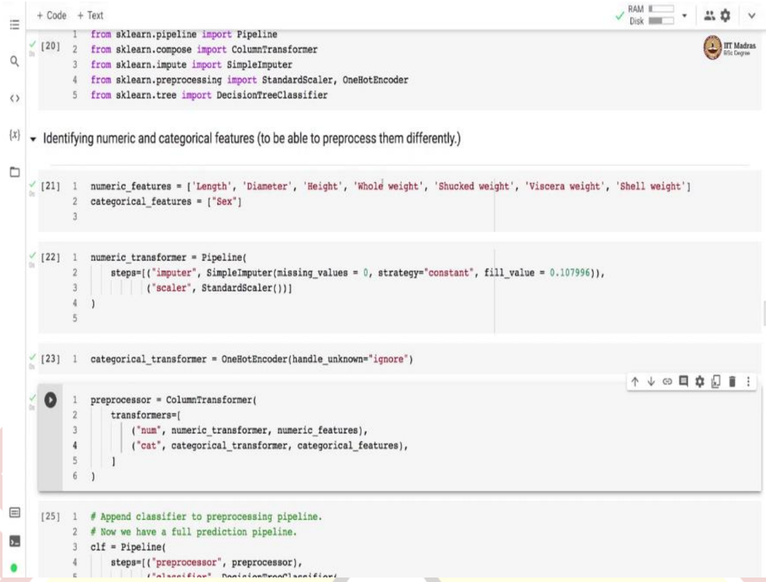
Next thing that we do is we split the data into training and test, we set aside 20% examples as test examples. And now we will define a model training pipeline.

(Refer Slide Time: 10:19)



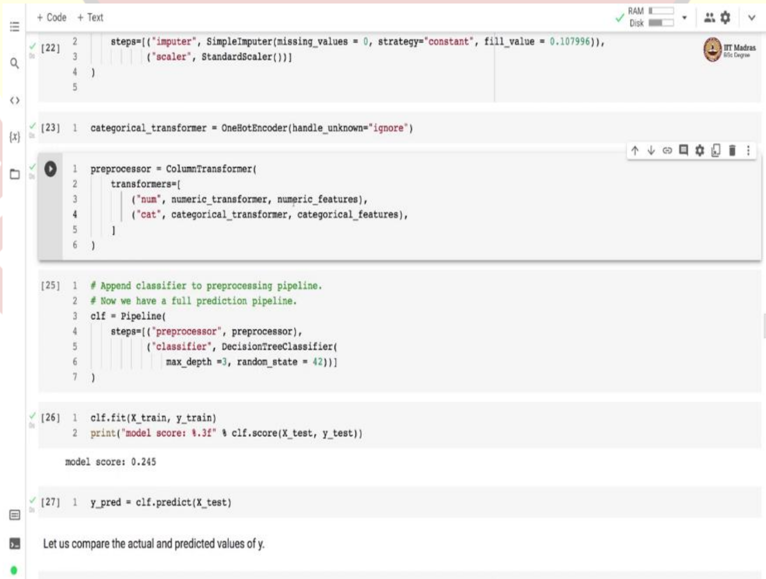
We will use pipeline construct to perform pre-processing of the data. This includes handling missing values, scaling the features, and handling categorical features. So, here we have mixed type of features that are majority of the features which are numerical features and we have a categorical feature as well. So, for that, we will be using ColumnTransformer. So, first identify numerical and categorical features and list them out. So, we have numerical features listed in this list and categorical features are listed in the other list.

(Refer Slide Time: 10:57)



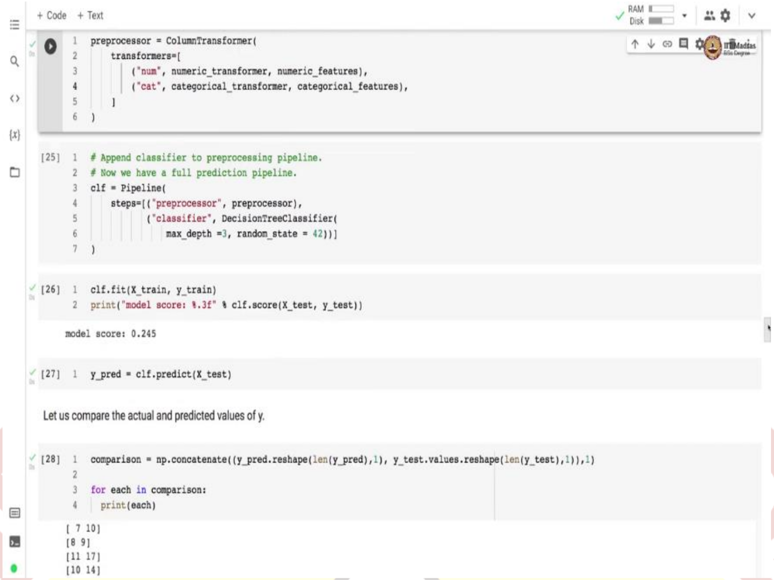
And we define numeric transformers and categorical transformers. So, numeric transformers are used to process the numerical features and categorical transformers will process the categorical features. So, here in numeric transformers, we have SimpleImputer and StandardScaler as two steps in the pre-processing transformation, the SimpleImputer uses the constant strategy and fill up the missing values with the values specified in fill _value parameter. In categorical transformer, we perform OneHotEncoder.

(Refer Slide Time: 11:41)



Then we define a ColumnTransformer which has got two transformers one is the numerical transformer which is applied on numeric features and categorical transformer which is applied on categorical features.

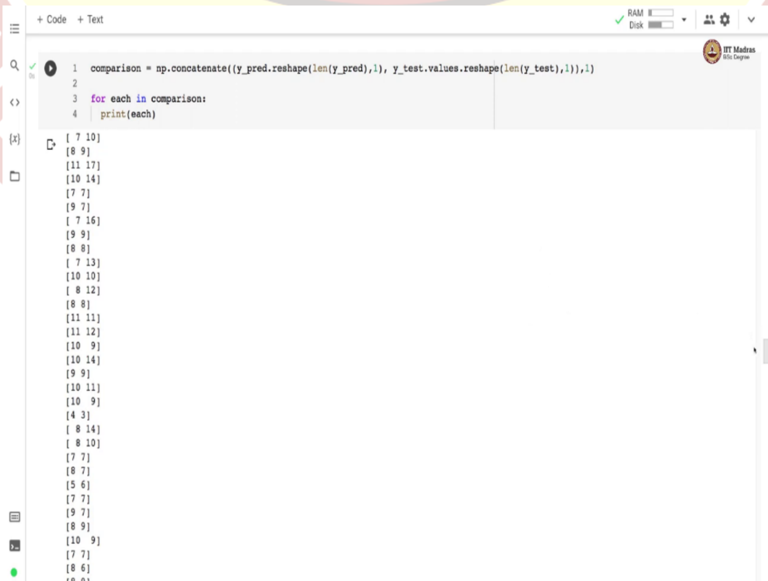
(Refer Slide Time: 11:56)



Now, we will define our model along with the pre-processor in the pipeline construct. So, we have a pipeline with two steps one is a pre-processor and second is a DecisionTreeClassifier with max _depth set to 3. So, we will be building a tree with a maximum depth of 3. So, we train the tree by calling the fit function on the pipeline object with feature metrics and training label vector.

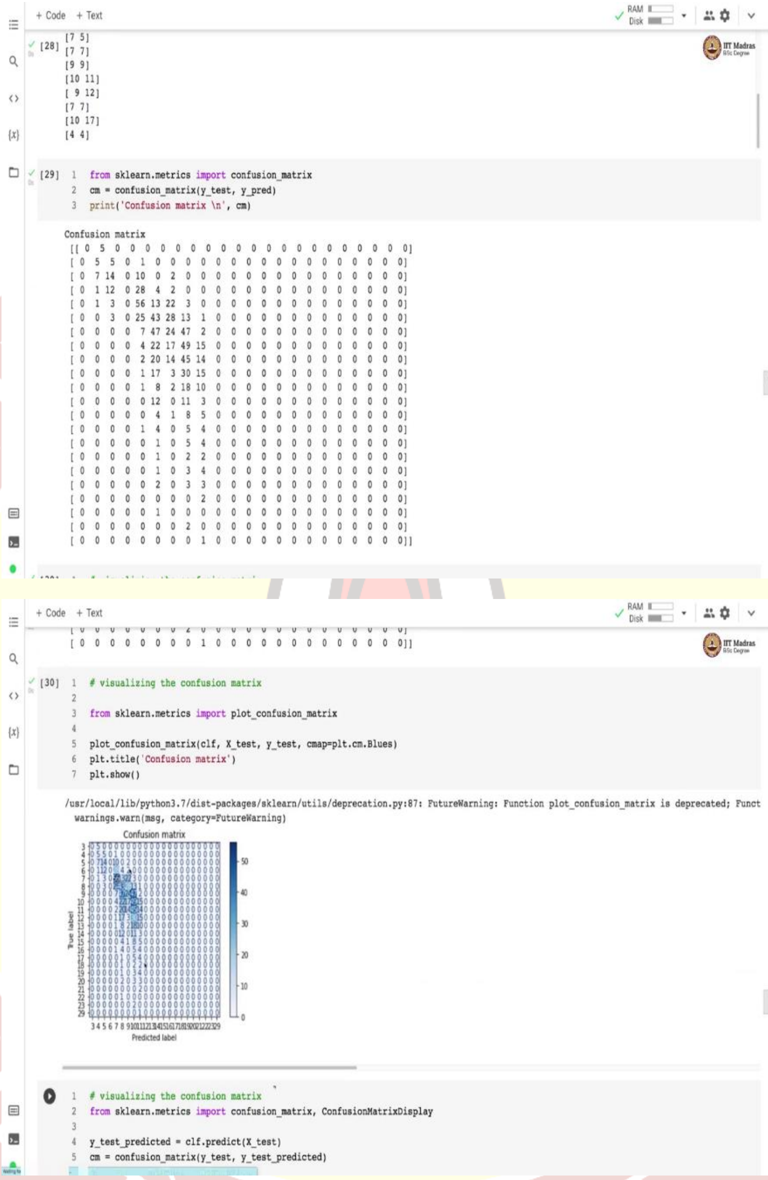
We can see that on the training set, we obtained the score of 0.245. We perform the prediction by calling the predict function on the pipeline object by supplying the test feature matrix and we obtain the predictions for each of the example in the test set.

(Refer Slide Time: 12:53)



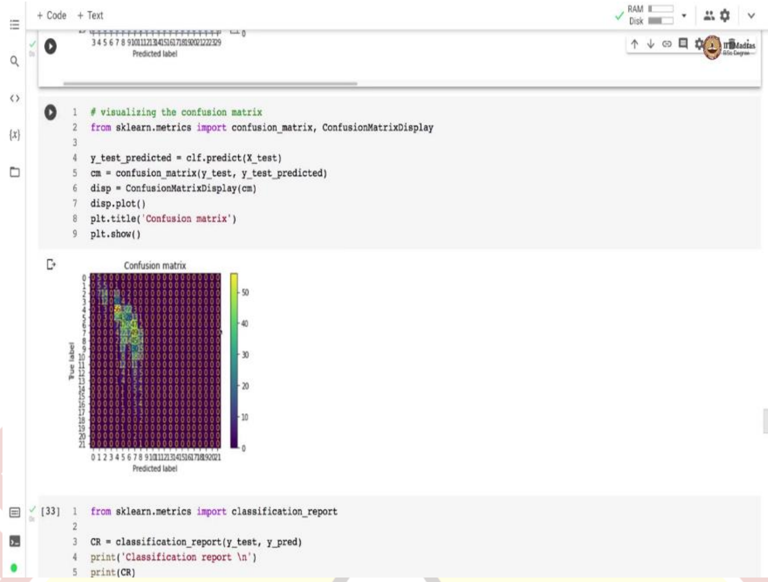
So, here, we have printed out, here we have printed out the actual label and the predicted label.

(Refer Slide Time: 13:03)



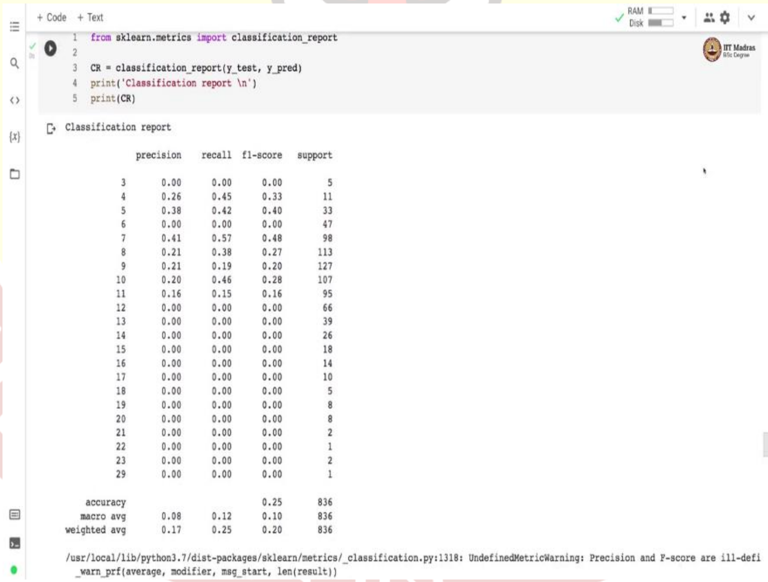
And we will actually calculate a confusion matrix and visualize the confusion matrix. So, you can see that there seem to be some kind of confusion in this particular part between different types of rings.

(Refer Slide Time: 13:25)



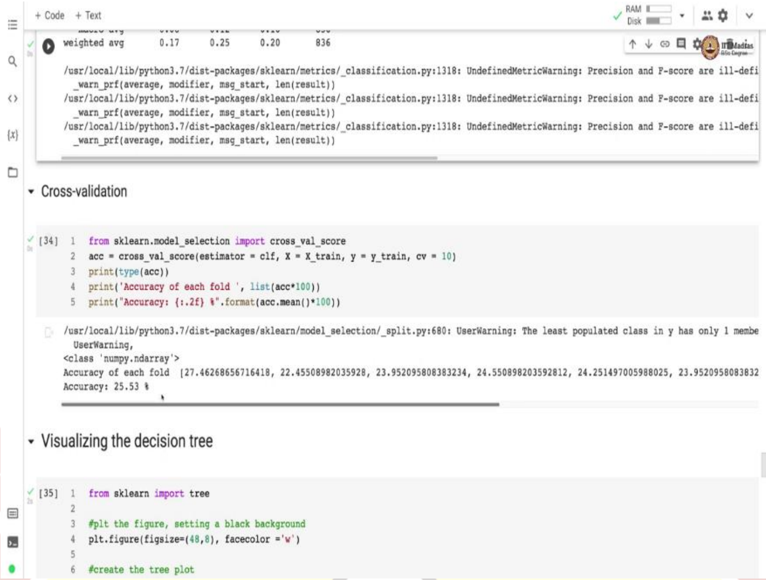
So, let us visualize the confusion matrix. And this visual confusion matrix shows the amount of confusion and you can see that in this particular region there is confusion that a lot of zeros because test set does not contain any examples from these ring types.

(Refer Slide Time: 13:50)



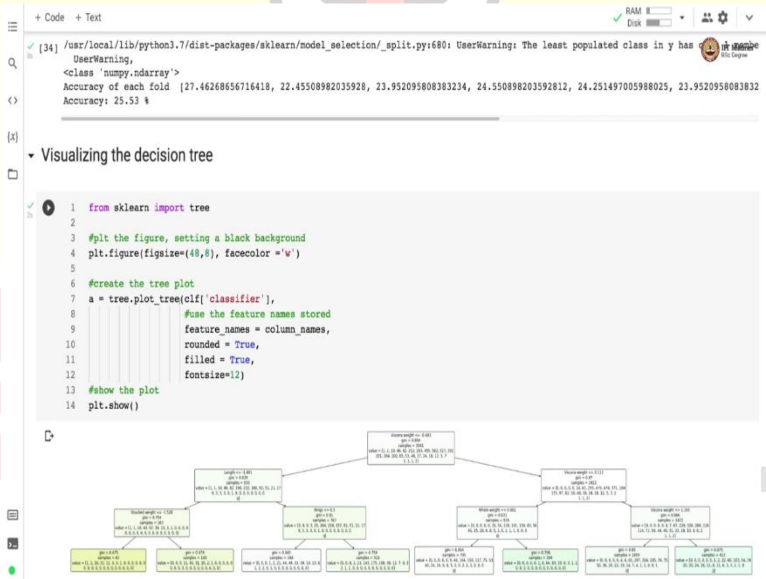
So, let us get a classification report and get precision recall and f1-score by the classes. You can see that bunch of classes have got 0 precision recall. The highest f1-score is obtained for number of rings equal to 7. And overall accuracy is 25%.

(Refer Slide Time: 14:24)



Let us also perform cross-validation and opt in the cross-validation score. So, the accuracy in cross-validation is 25.53%. Now, next what we will do is we can visualize the decision tree that was learned.

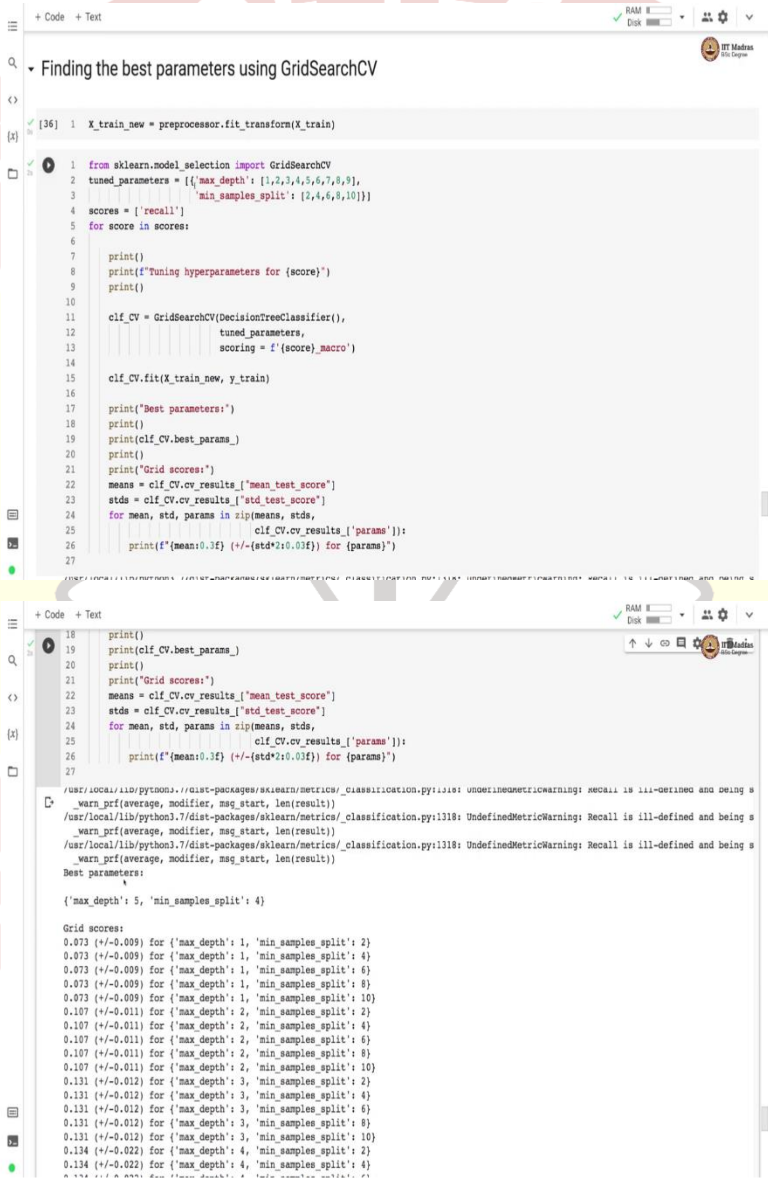
(Refer Slide Time: 14:41)



So, for that we use a tree API from sklearn and called plot _tree on the tree class. So, the learn tree is on the screen, so you can see that a top-level node is viscera weight and we have printed the gini index as well as the number of samples and the values over here. So, as you go down the sub trees, the gini index has improved and you can see that the number of samples.

So, here we got a split where 930 samples were pushed on the left and the remaining 2,411 samples are pushed to the right. So, in the second step, we use length as a feature where we want to perform the split, here we use viscera weight, again as a feature to perform the split and so on. So, you can see that the leaf node, the first leaf node has got 63 samples, second one has got 100 samples and so on, and you can see the values of the target in each of the leaf node which is also printed over here.

(Refer Slide Time: 16:14)

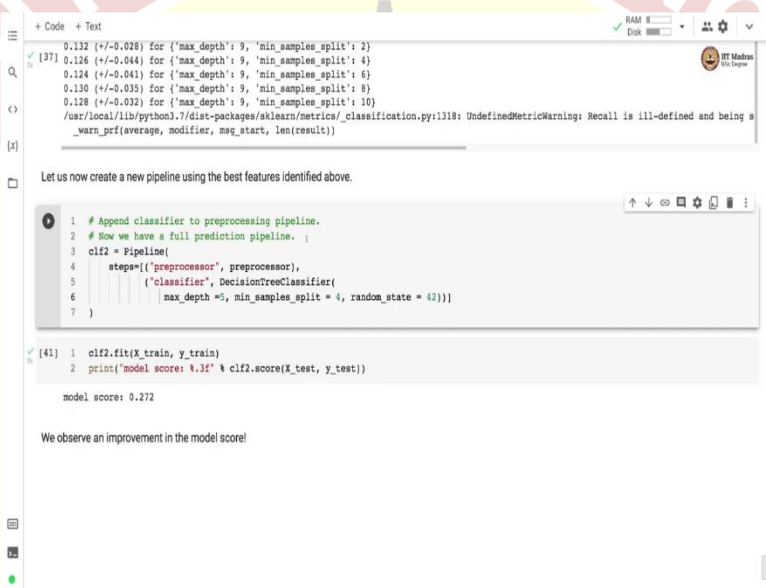


Now, as usual, there are certain configurable parameters in decision tree, like maximum depth and then minimum number of samples to split. So, the depth we want to expand with the depth between 1 to 9 and the min _samples _split between 2 to 10. We are going to use recall as a major.

So, here we perform a grid search by defining GridSearchCV object with DecisionTreeClassifier as the estimator, then the grid, which is specified in tuned _parameters. And using the recall _macro, as a scoring measure. We call the fit function on the GridSearchCV object with the training feature matrix and label vector as input.

The base parameters are printed with by accessing best _params _ member variable of the GridSearchCV object. You also print the mean _test _score under standard deviation of the test score. So, you can see that the best parameters are obtained with max_depth equal=5 and min _samples _split = 4.

(Refer Slide Time: 17:45)



Now, we create a new pipeline with the best features identified above. So, we set the max _depth = 5 and min _samples _split = 4. And we define the pipeline object and we fit the pipeline object with the training data. Here we obtained the model score of 0.27 which is improvement from what we obtained earlier which was 25%. So, here we have got the score of 0.27 as compared to earlier score of 0.25.

So, we observe an input improvement in the model score. So, as an exercise you should perform the evaluation of this classifier on the test set. Other interesting exercise to do is to also attempt this Abalone problem as a regression problem and see if you get a better model. In general, the accuracy is quite low on this dataset. So, this was a demonstration of decision trees with continuous and discrete features. We use decision tree to predict the number of rings in Abalone.