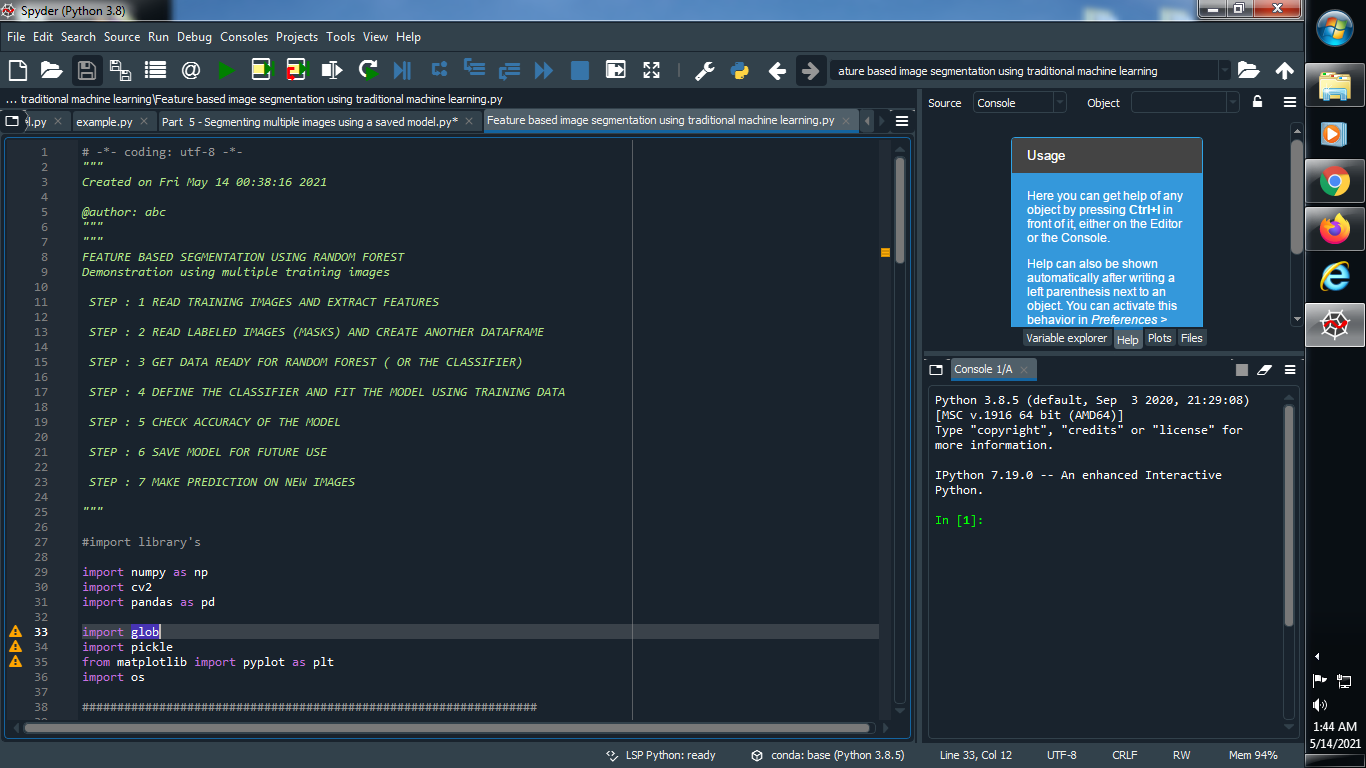
**→ Feature based image segmentation using traditional machine learning :**

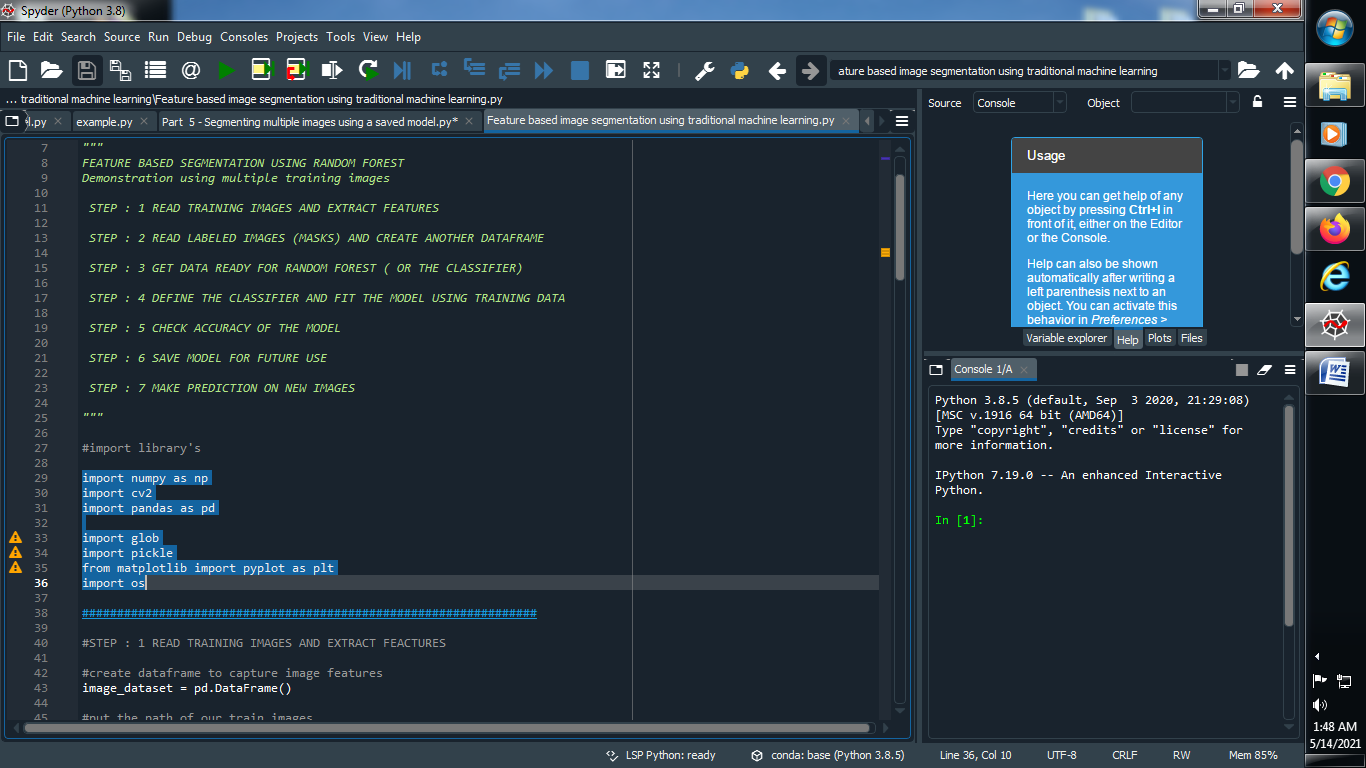
**(1) Steps for this tutorial :**

****

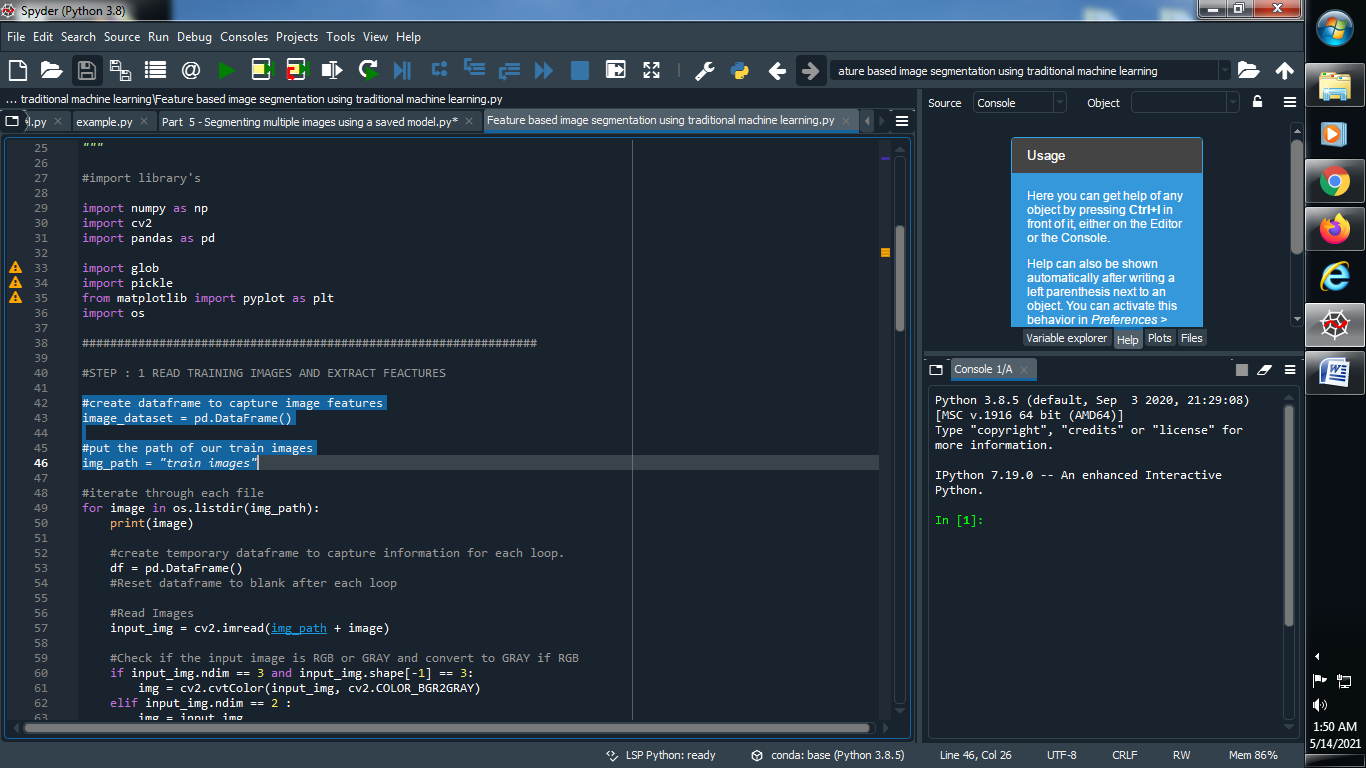
**####################################################################################################################**

**(1) Step : 1 READ TRAINING IMAGES AND EXTRACT FEATURES :**

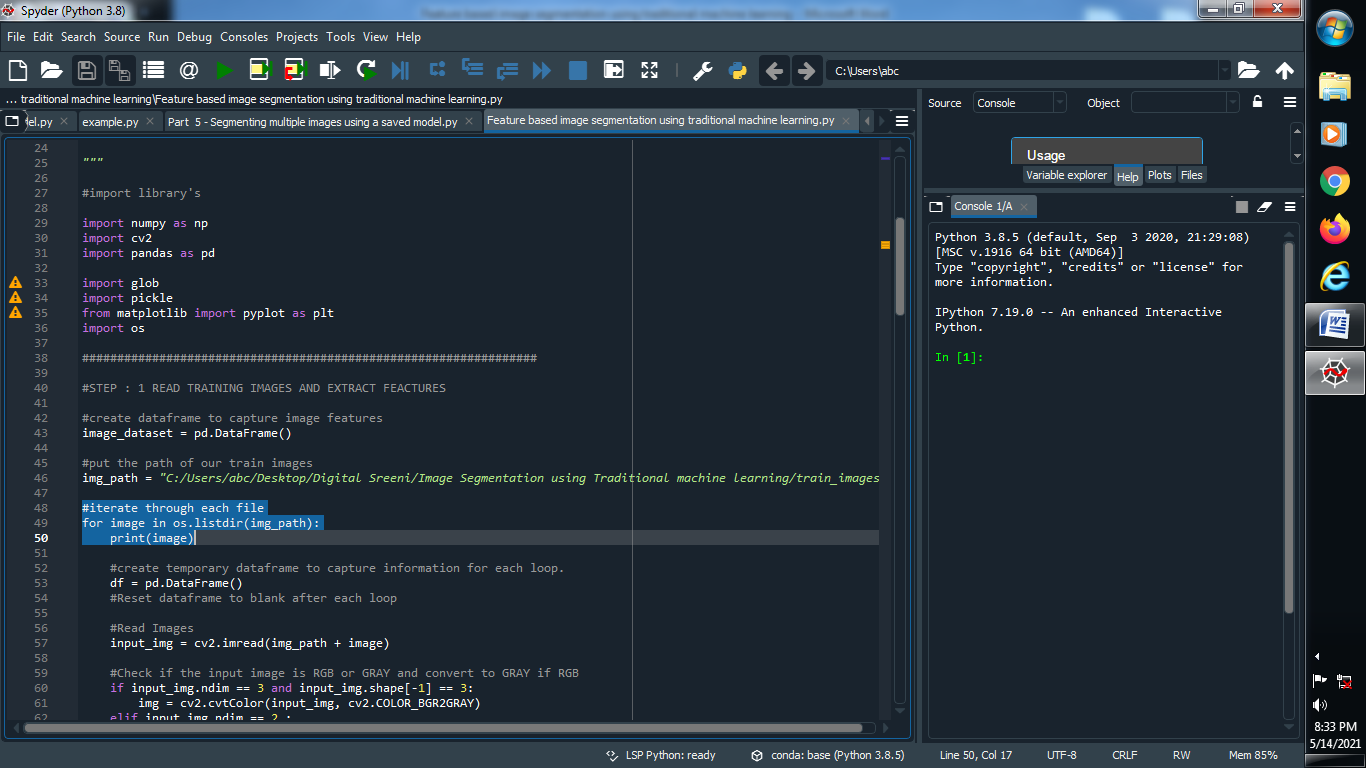
**→ Let’s import some library’s :**

****

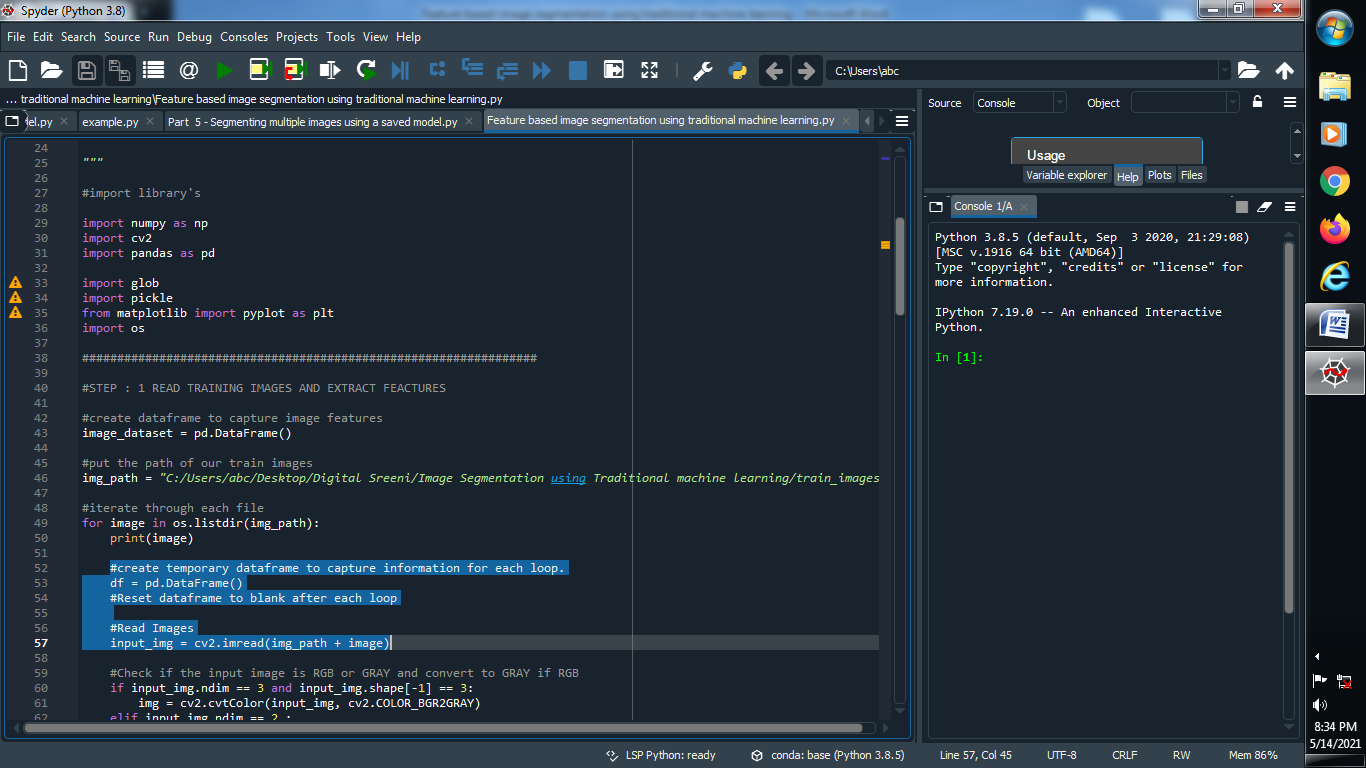
**(3) Create a data frame for capture features and give the path of our trained images :**

****

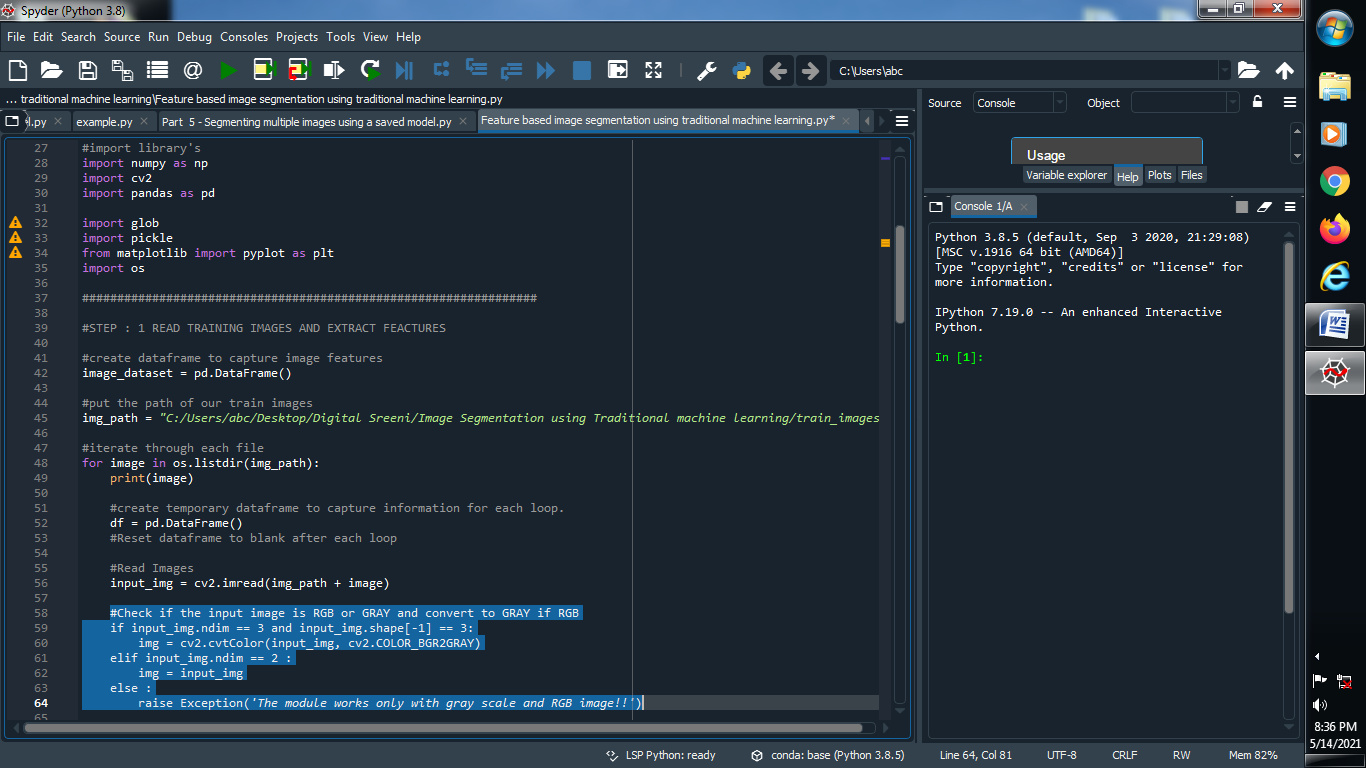
**(4) Create one function so that it iterate through each image :**

****

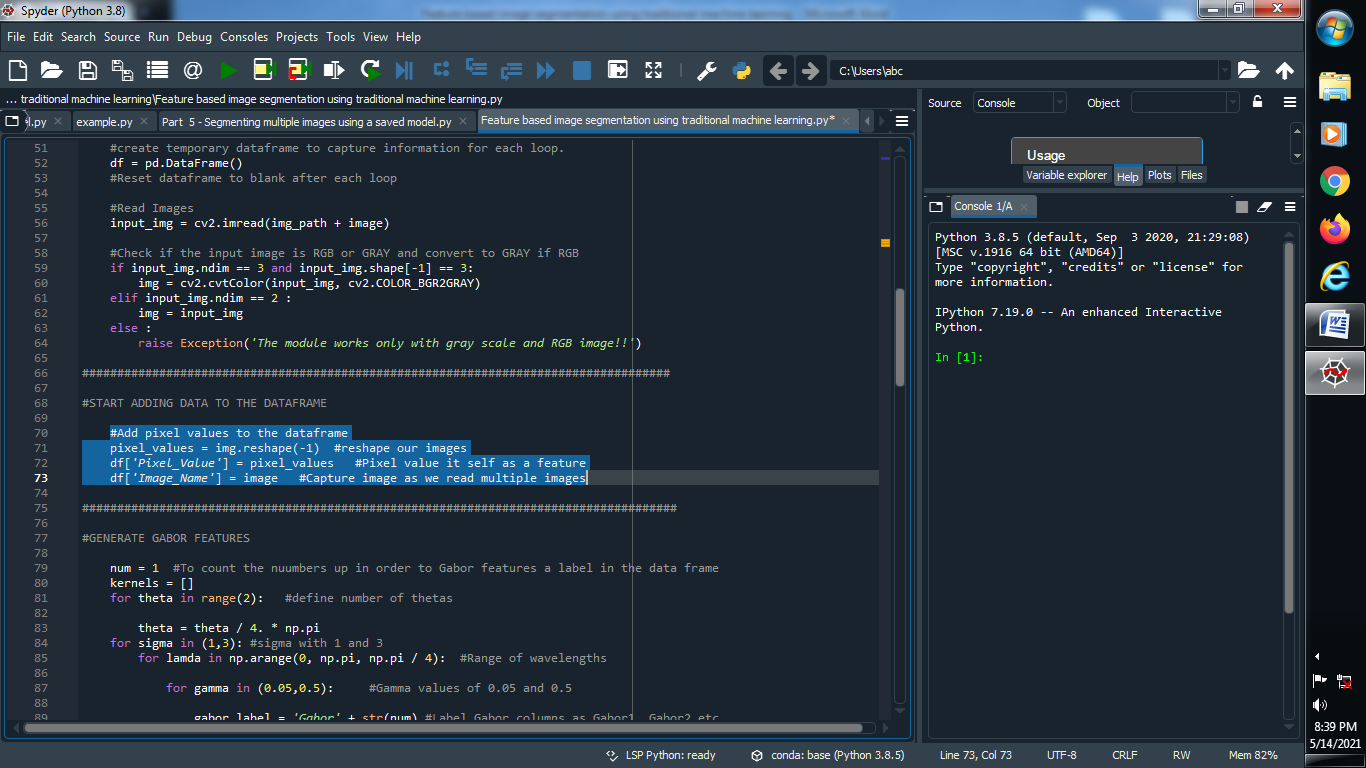
**(5) Create a temporary data frame to capture image through each loop and read that all images :**

****

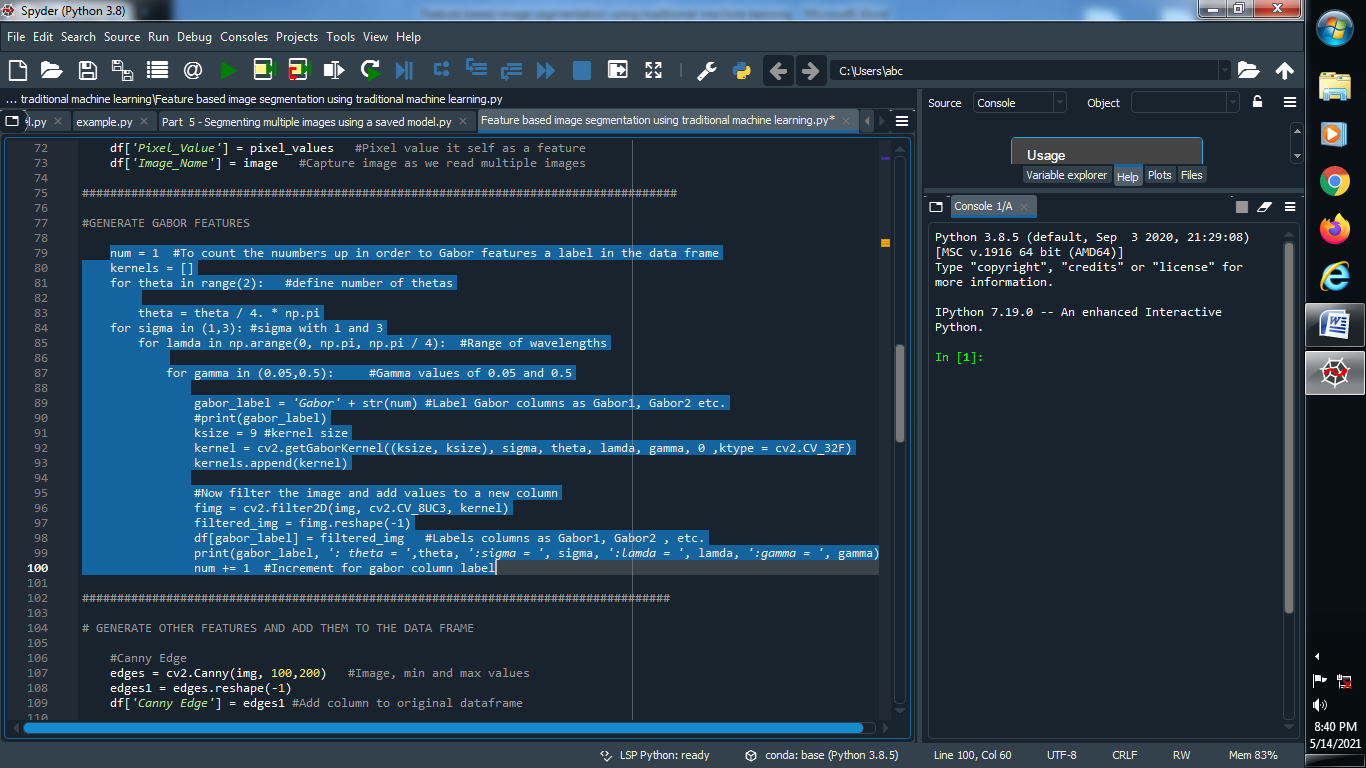
**(6) Check all the images if RGB than convert it onto gray image :**

****

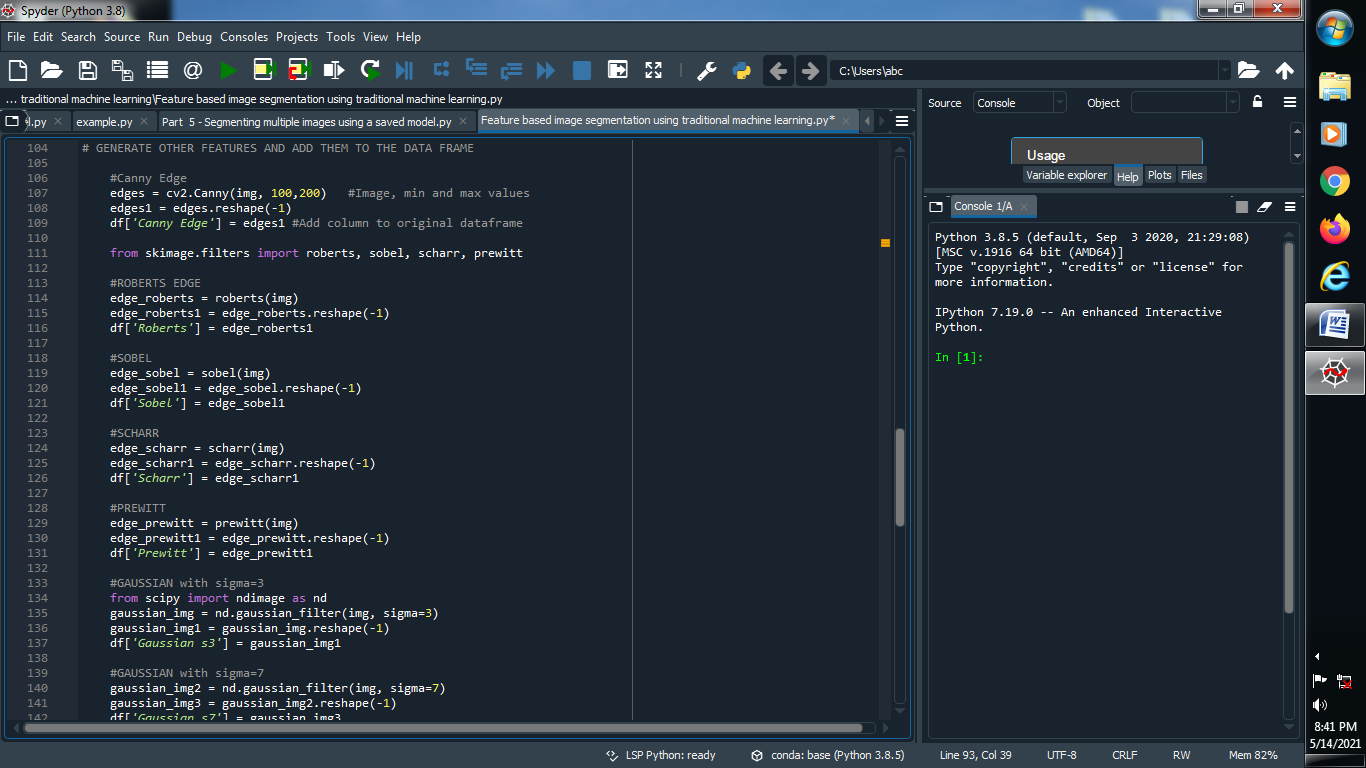
**(7) Start adding data to data frame :**

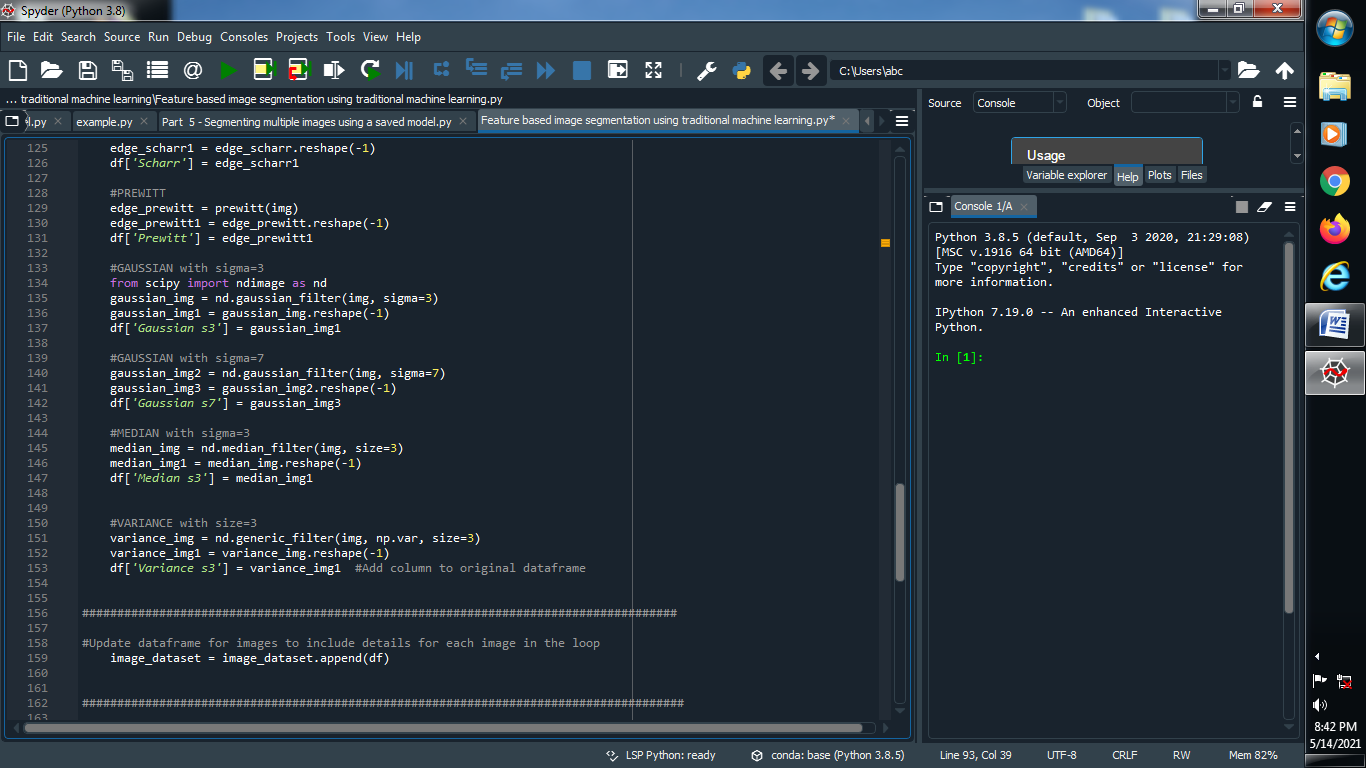
****

**(8) Apply Gabor Filter :**

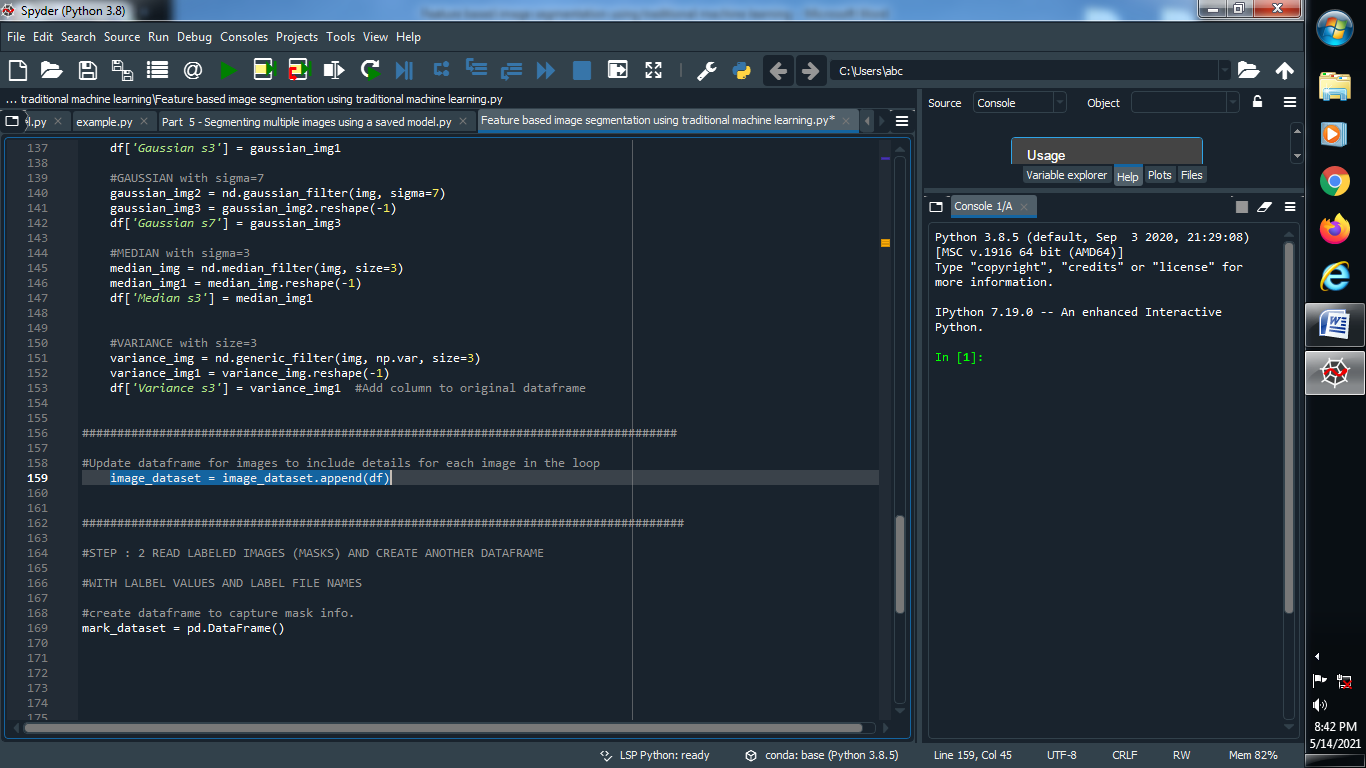
****

**(9) Apply and generate another filter and add them to the data frame :**

****

****

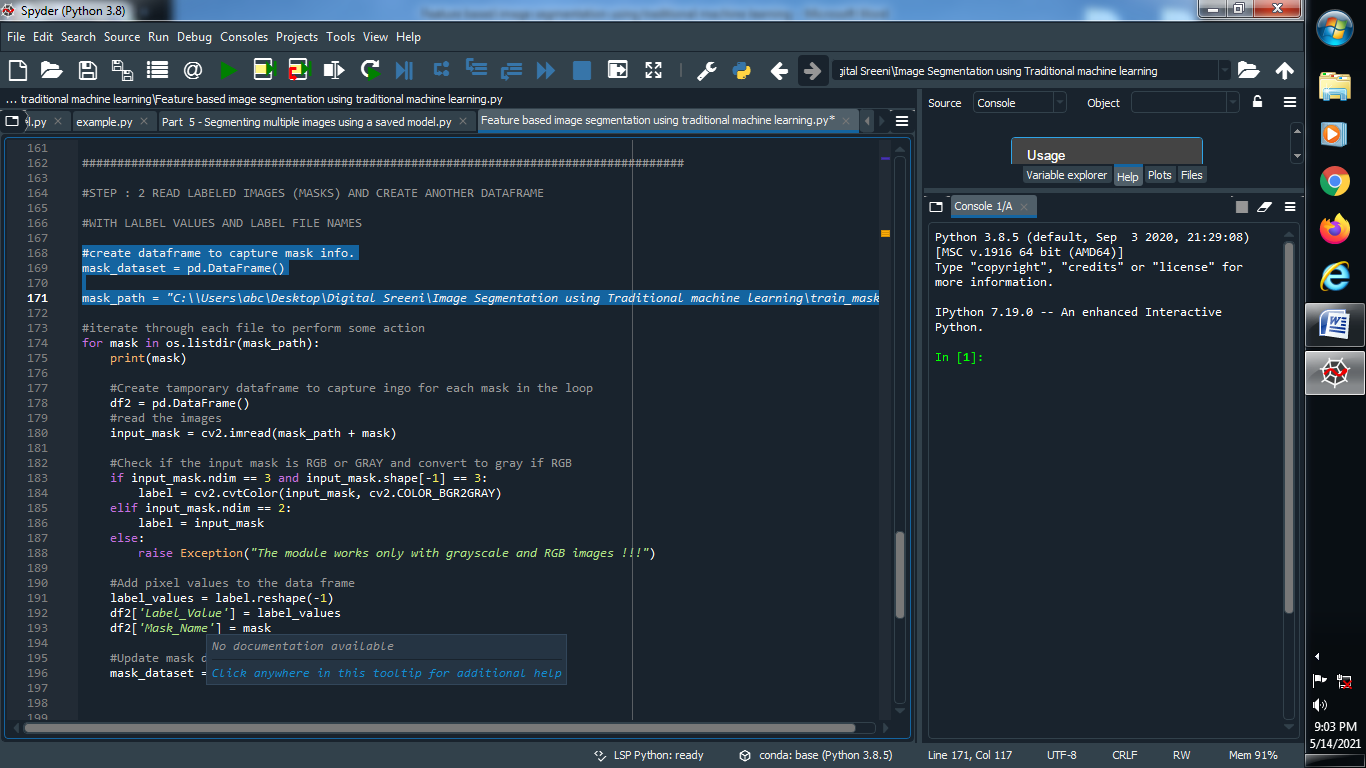
**(10) Update data frame to include images of each loop :**

****

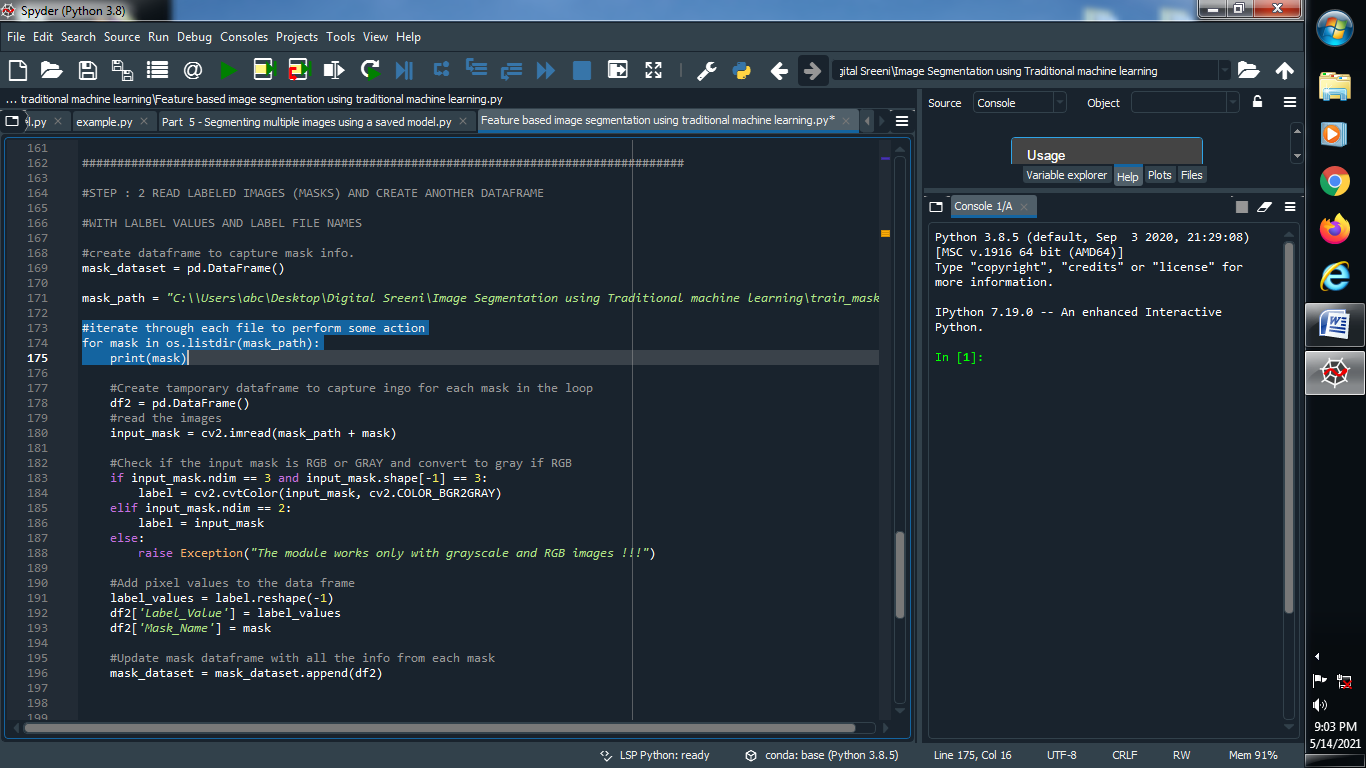
**########################################################################################################################**

**→ STEP : 2 READ LABELED IMAGES AND CREATE ANOTHER DATAFRAME**

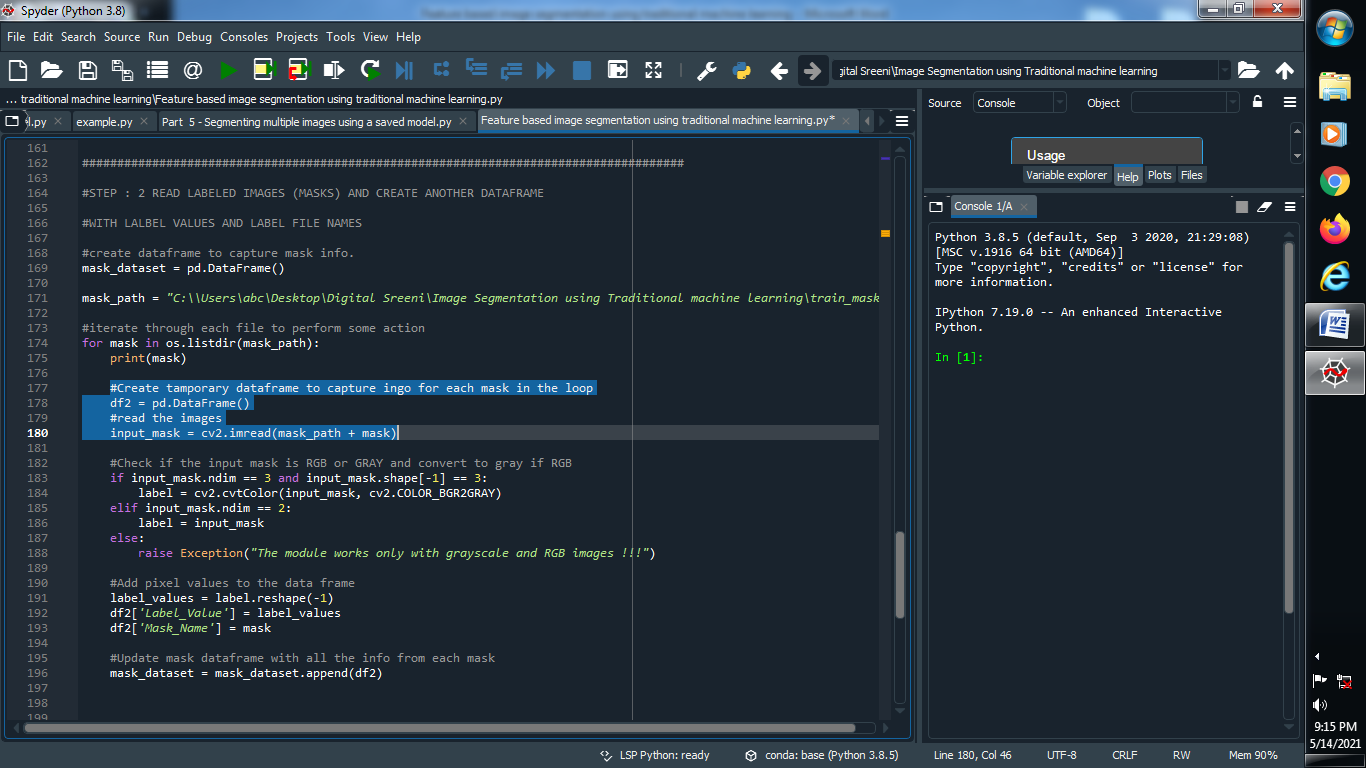
**(1) Create a mask data frame and give the path of it :**

****

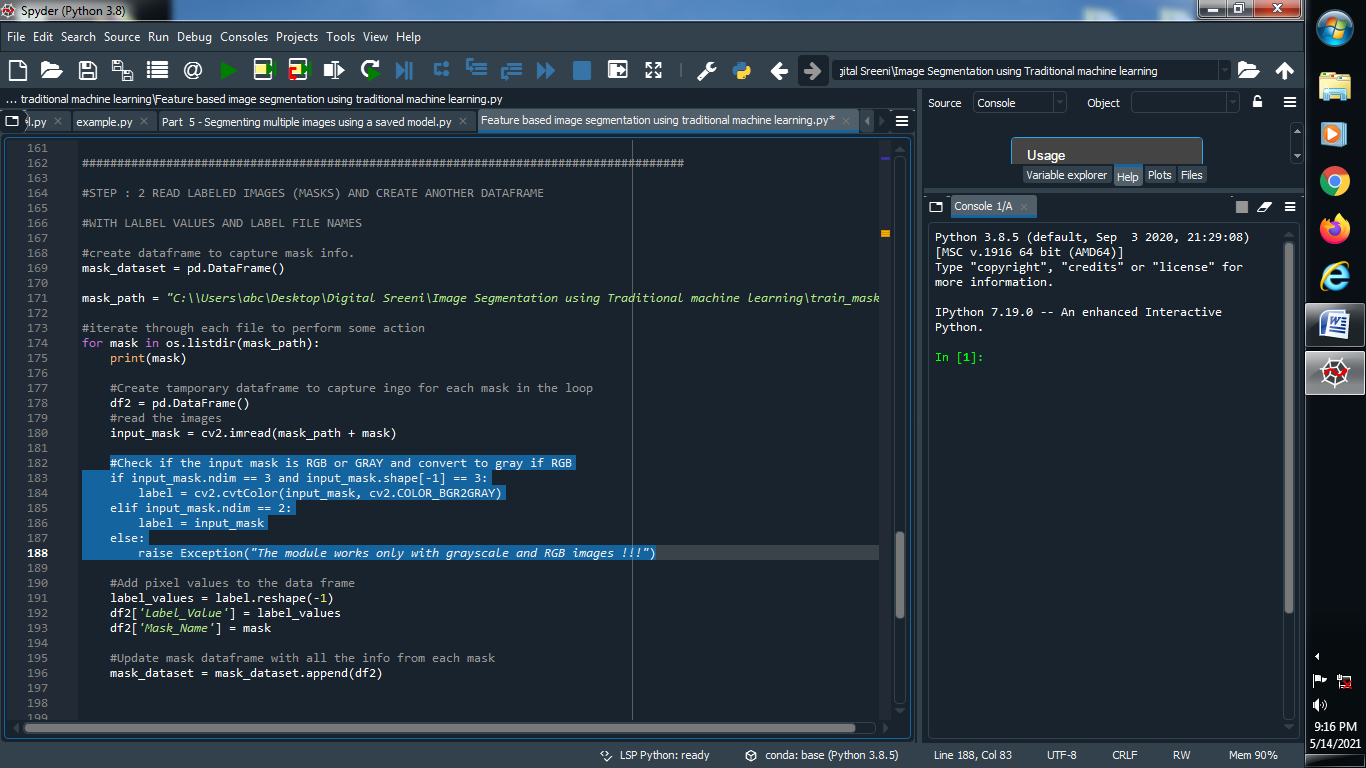
**(2) Create one function for iterate through each images :**

****

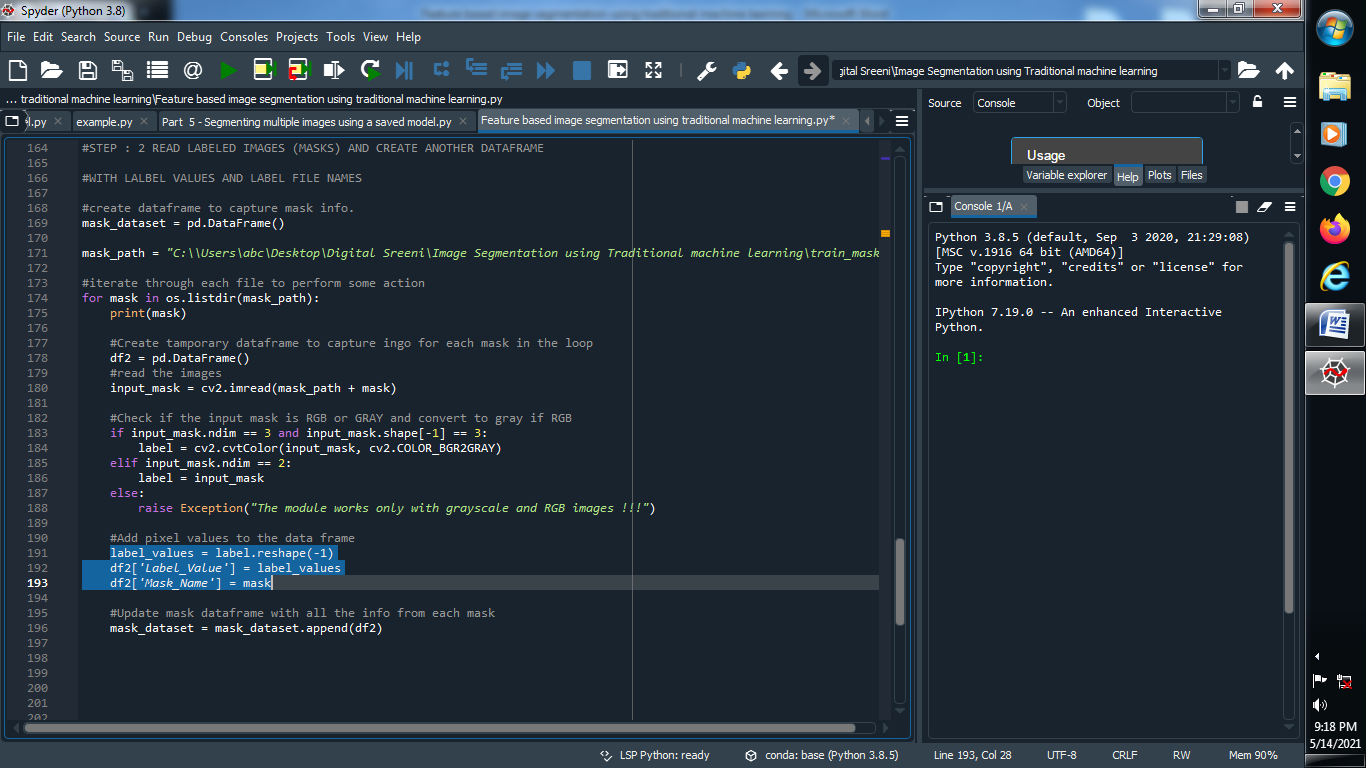
**(3) Create a temporary data frame to capture information of each mask and read that images :**

****

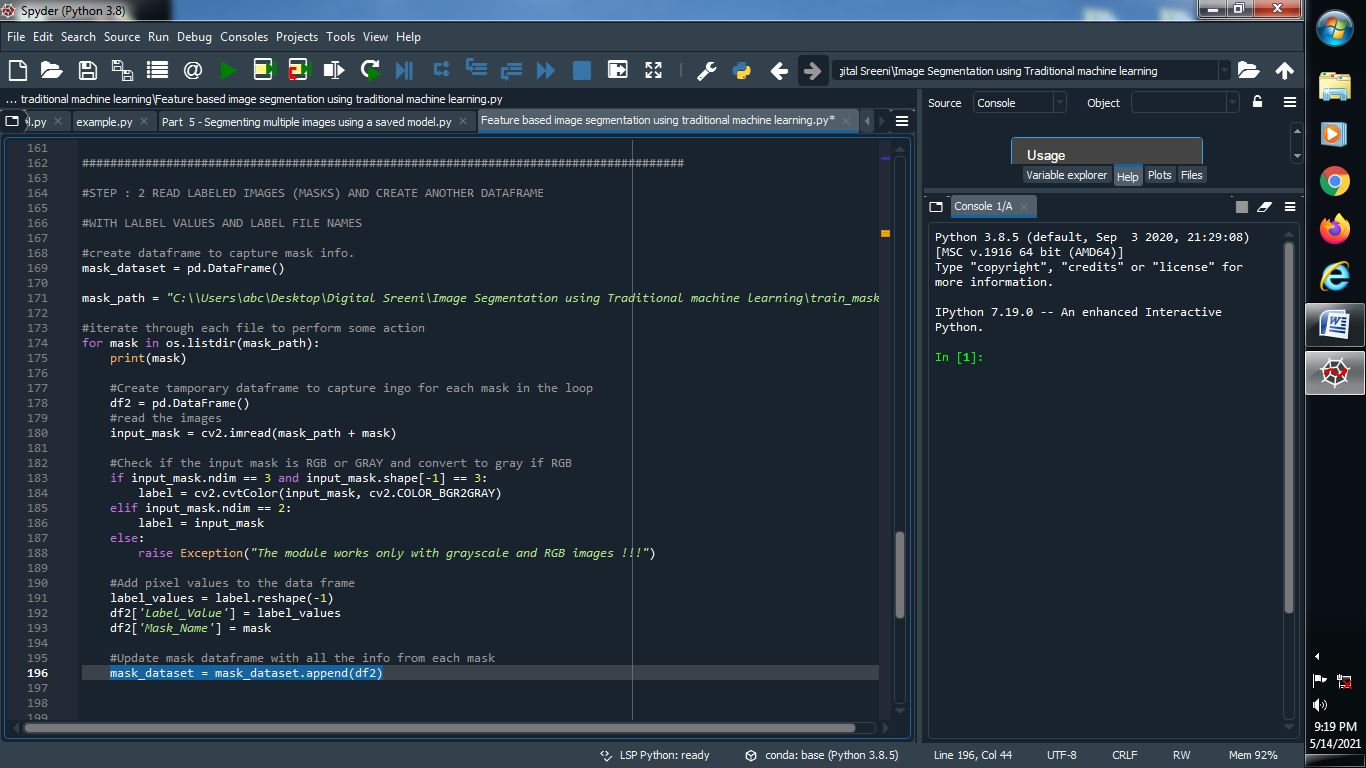
**(4) Check if the mask is RGB or GRAY if RGB than convert it to GRAY :**

****

**(5) Add pixels value to the data frame :**

****

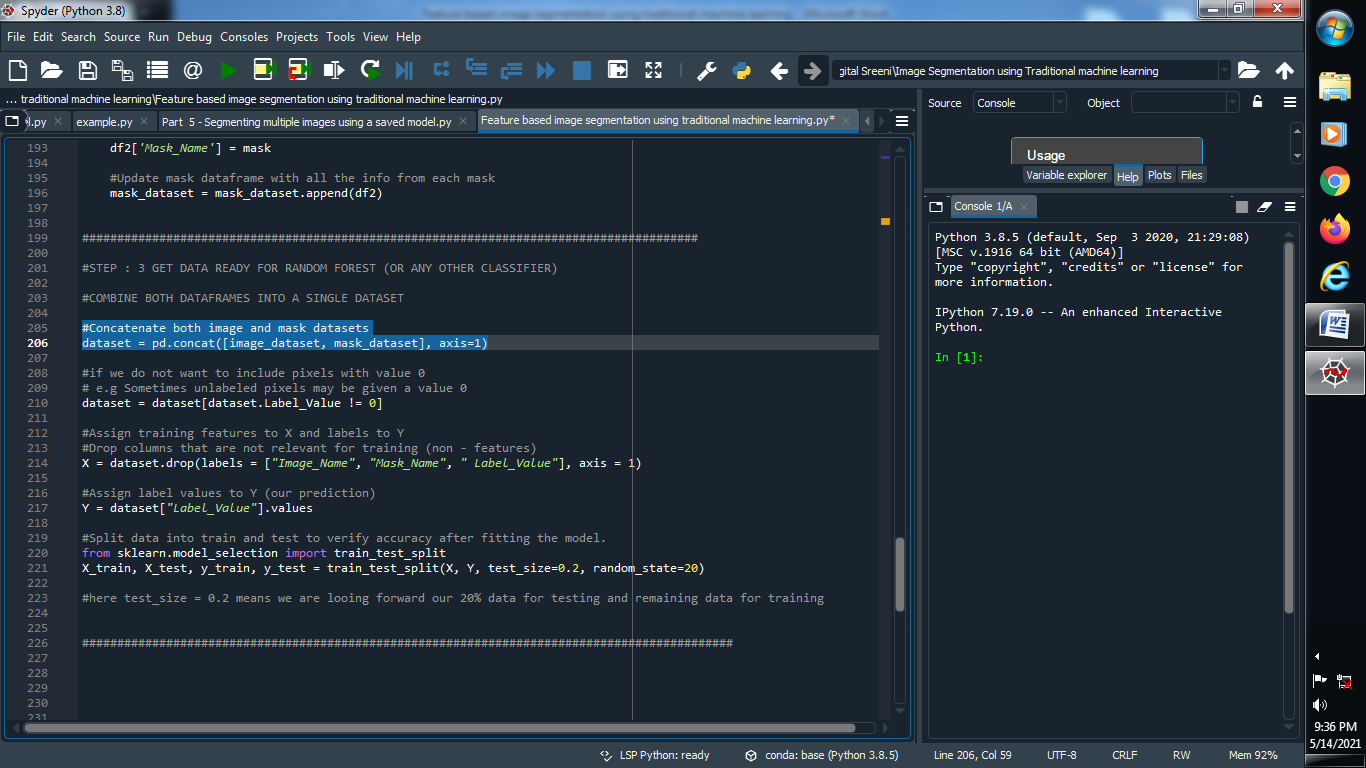
**(6) Update our data frame with all the information with the each mask :**

****

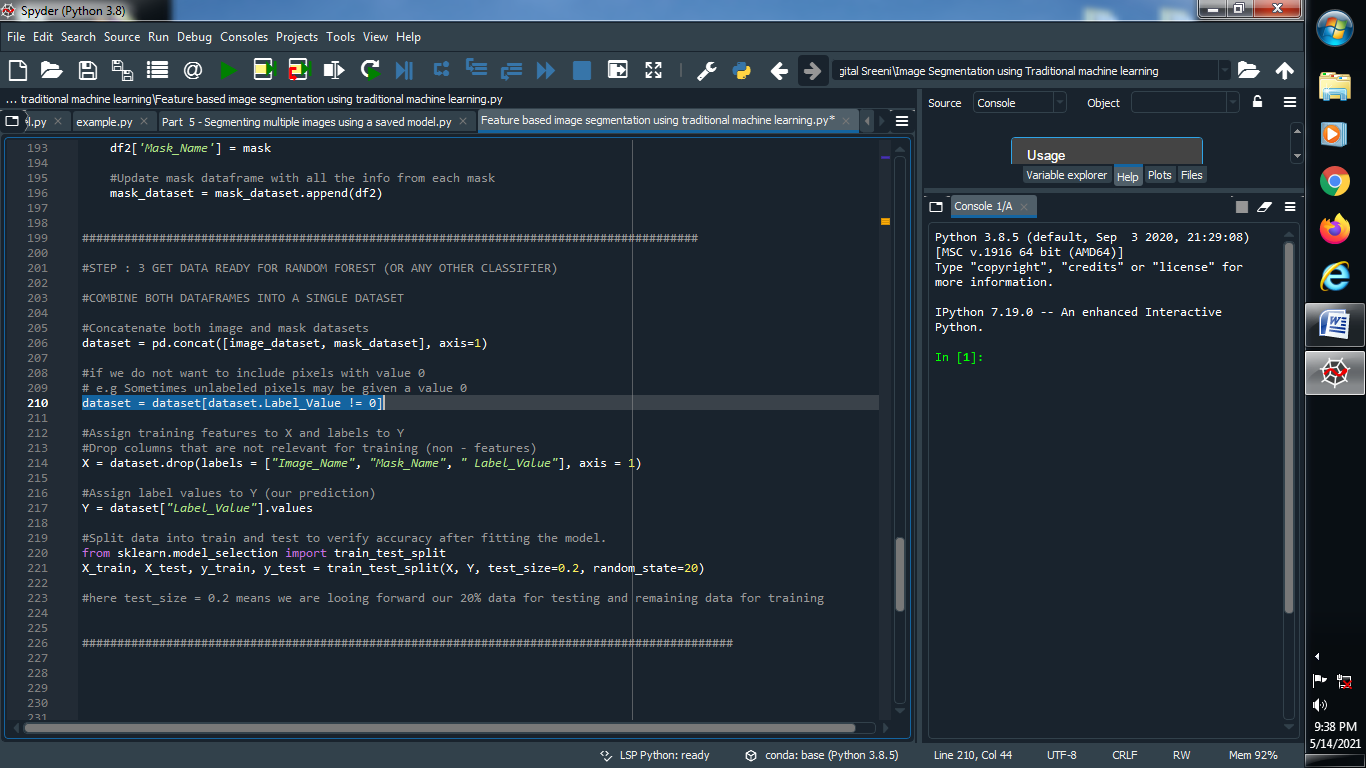
**###################################################################################**

**STEP : 3 GET DATA READY FOR RANDOM FOREST ( OR OTHER CLASSIFIER )**

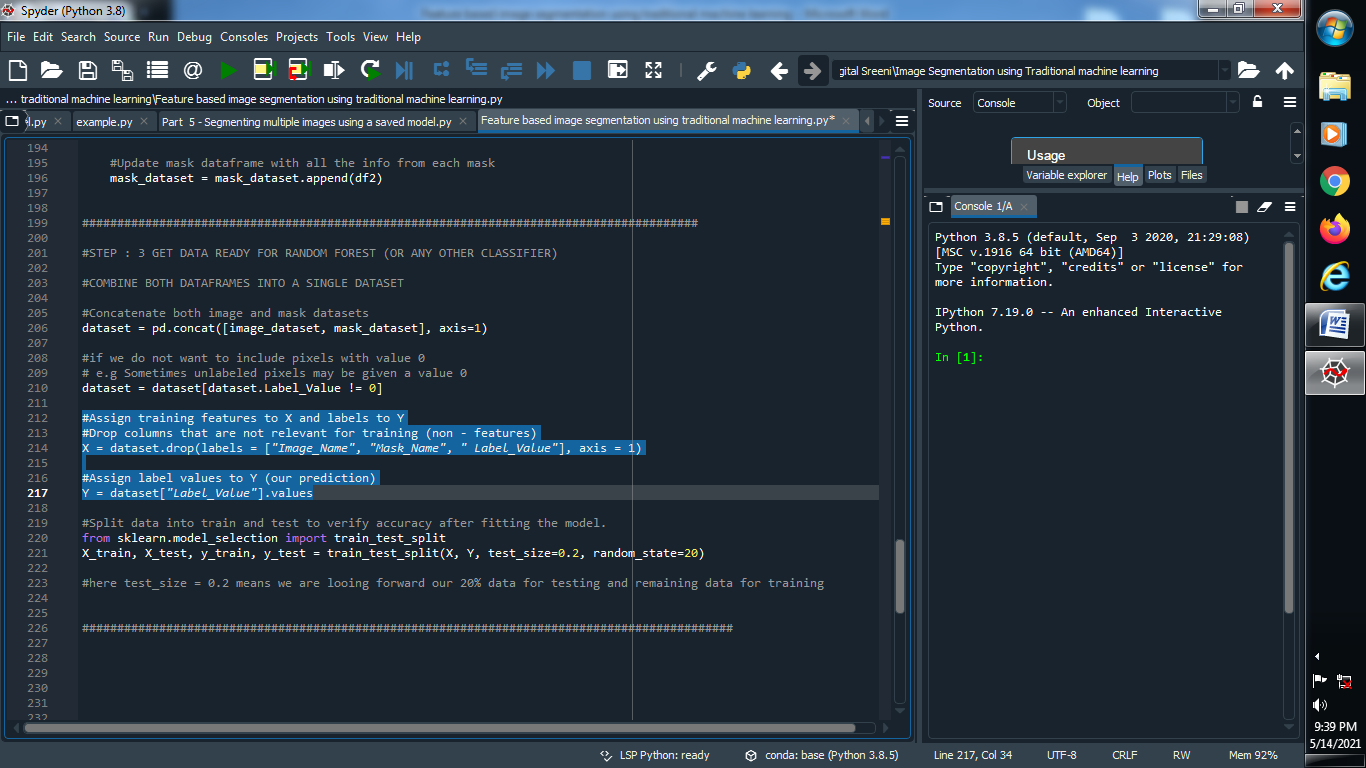
**(1) Combine both dataset into a single dataset using concatenate method :**

****

**(2) Delete that pixel values which is 0 :**

****

**(3) Assign training features to X and Y and drop the columns that are not relevant data for training :**

****

**Note : Always remember that in machine learning our X is allocated for training data and Y is for our Prediction ..**

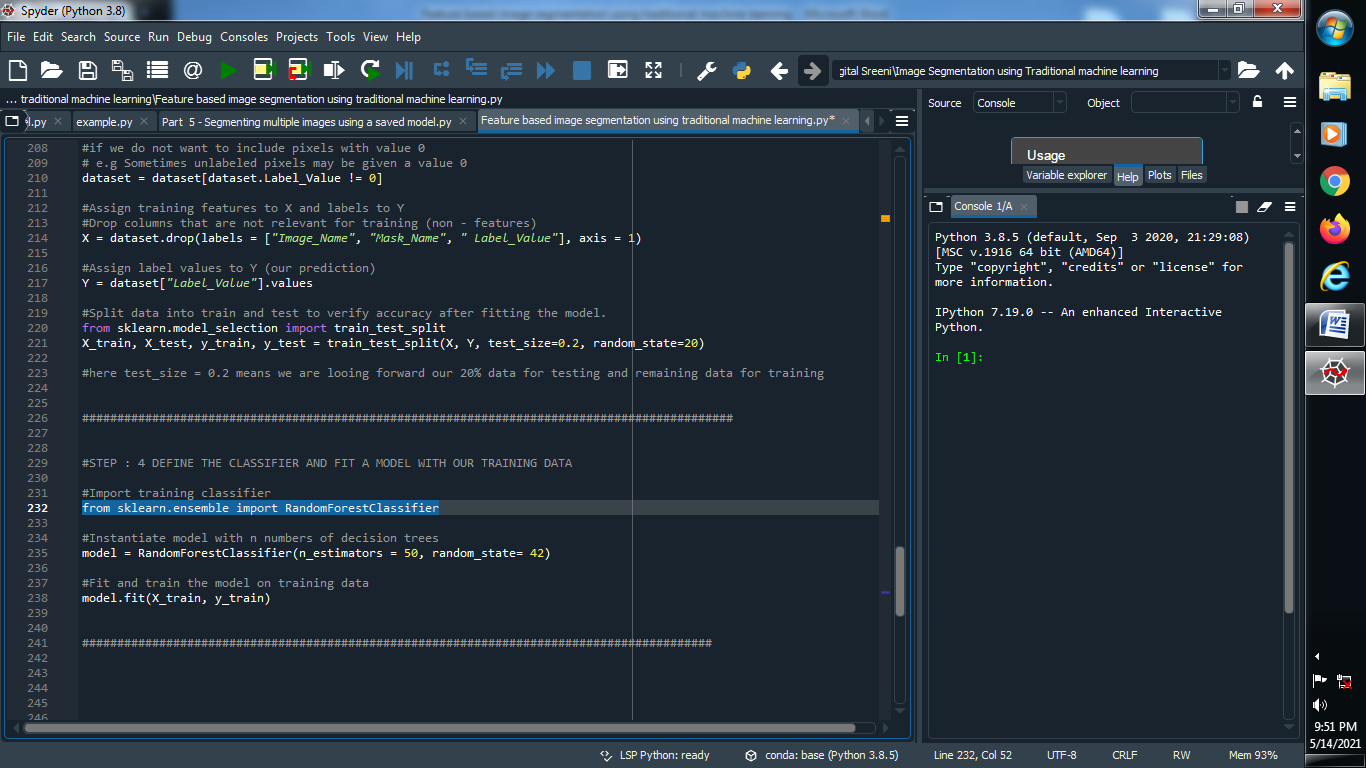
**(4) Split the data into training and testing to verify accuracy after fit the model :**

****

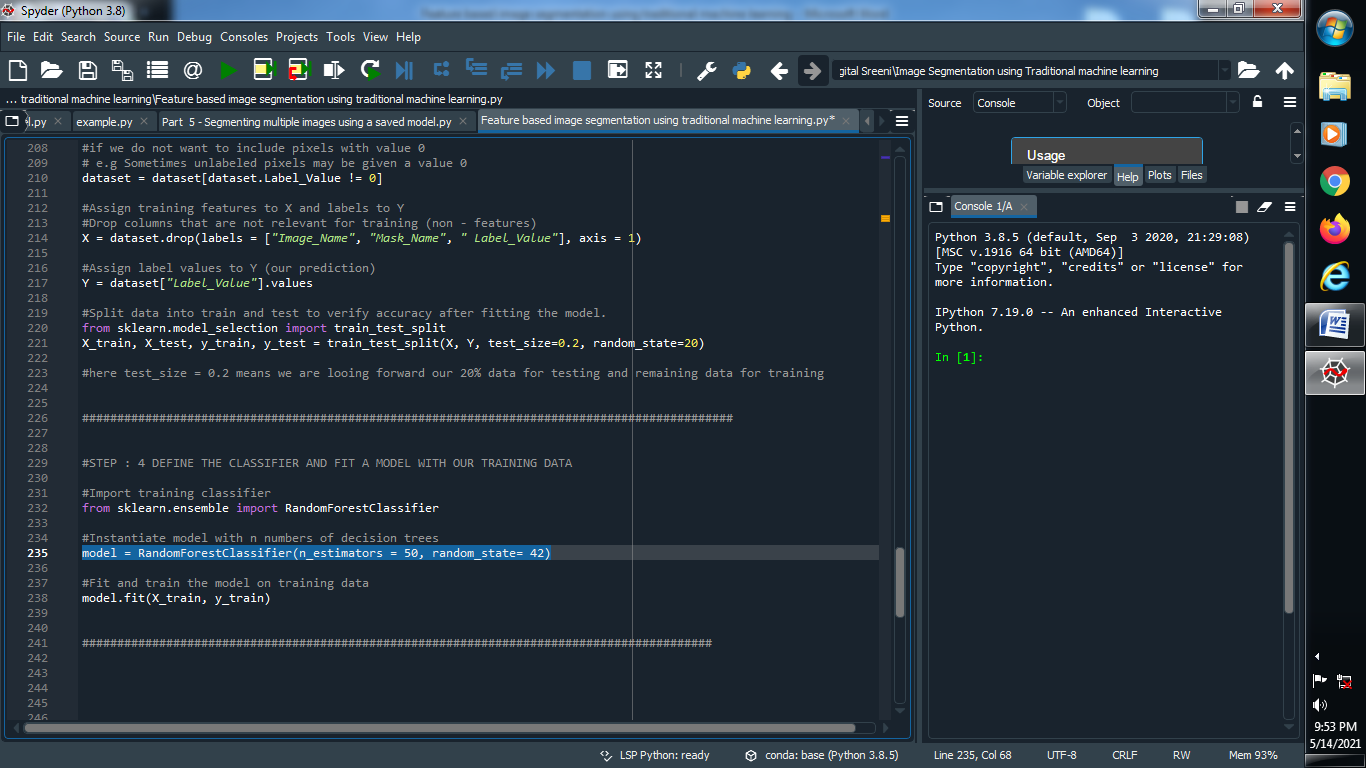
**####################################################################################**

**STEP : 4 DEFINE A CLASSIFIER AND FIT A MODEL WITH OUR TRAINING DATA**

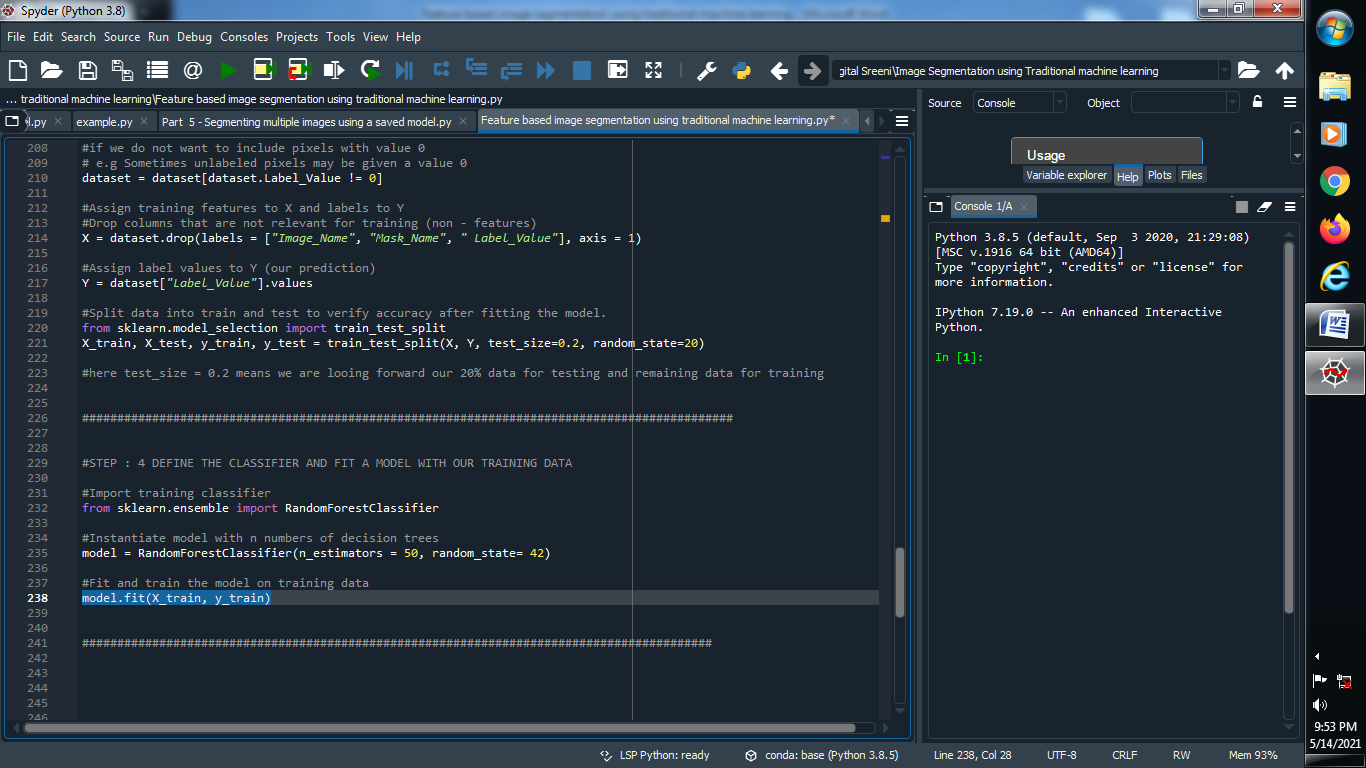
**(1) Import training classifier :**

****

**(2) Create a model and instantiate model with n number of decision trees :**

****

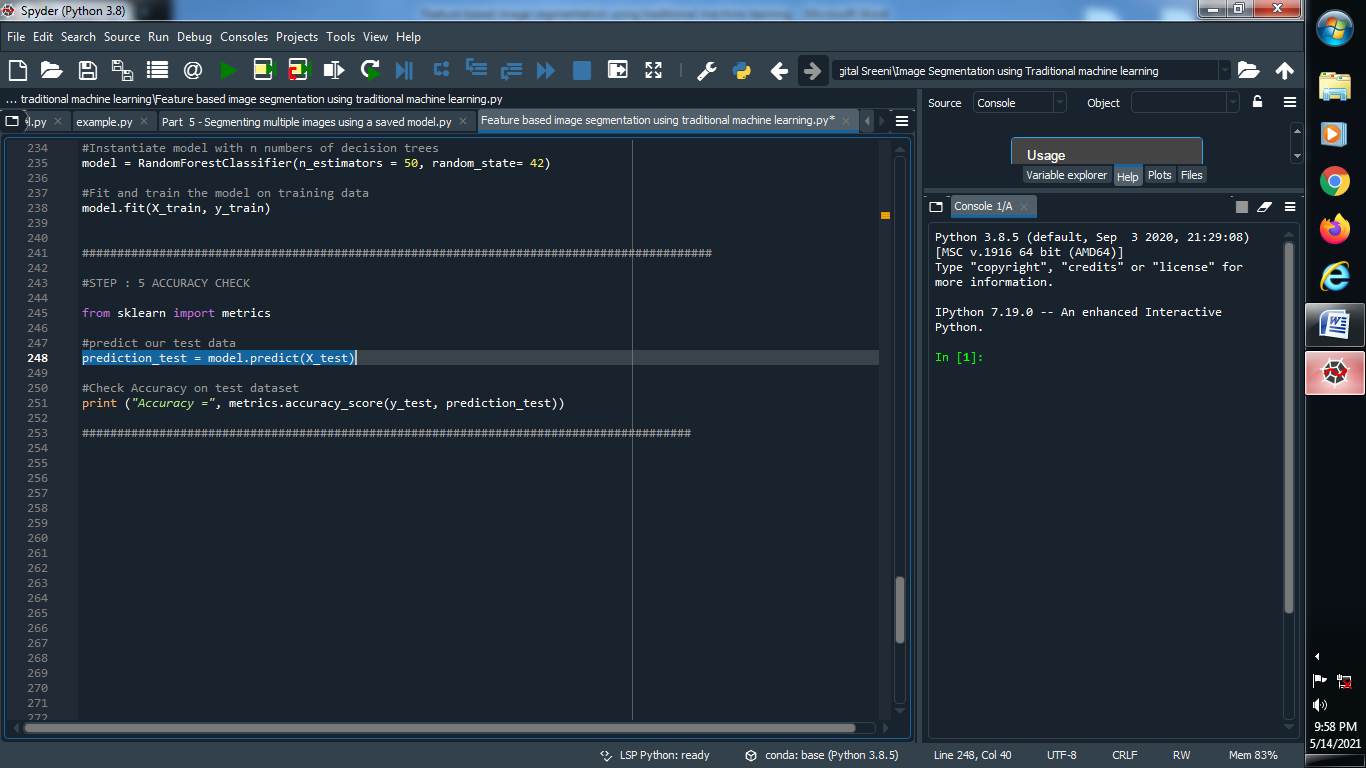
**(3) Fit and train the model on training data :**

****

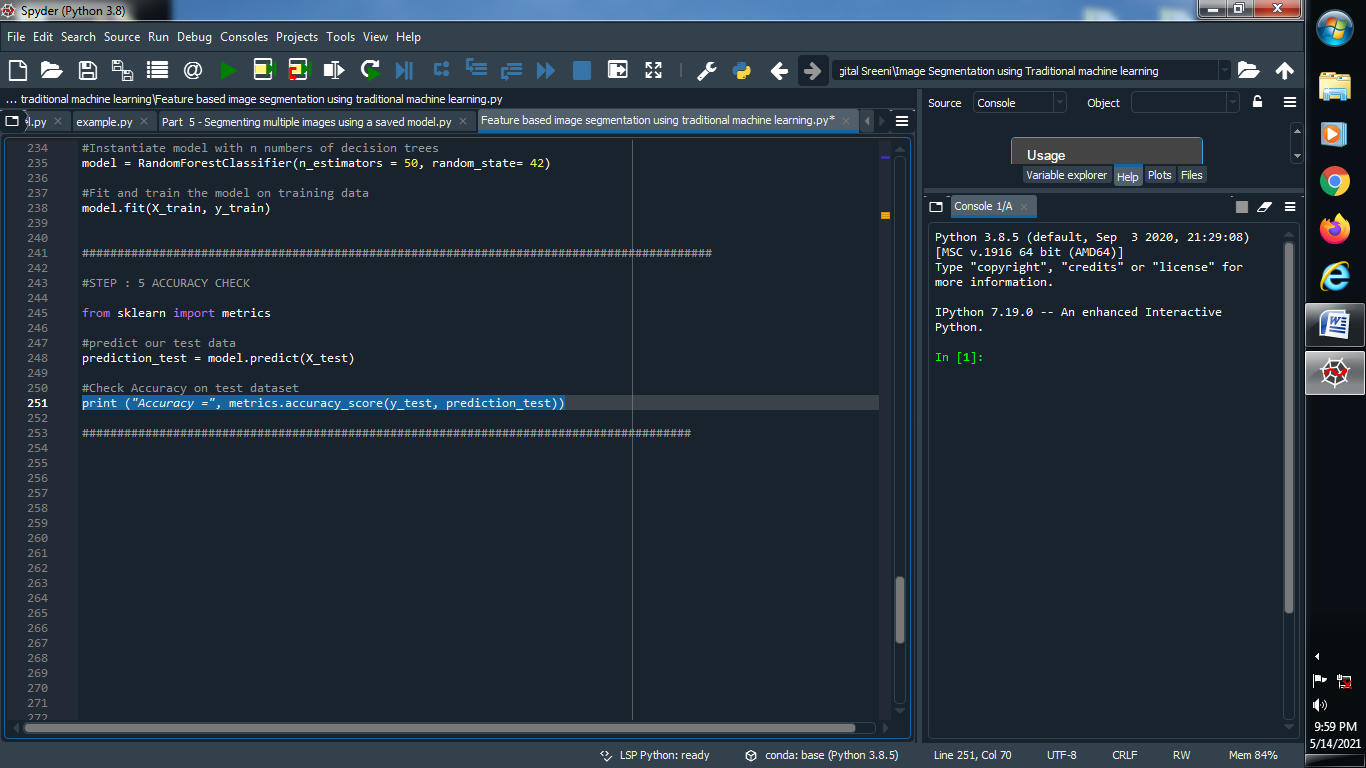
**###############################################################################################**

**STEP : 5 ACCURACY CHECK**

**(1) Predict our test model :**

****

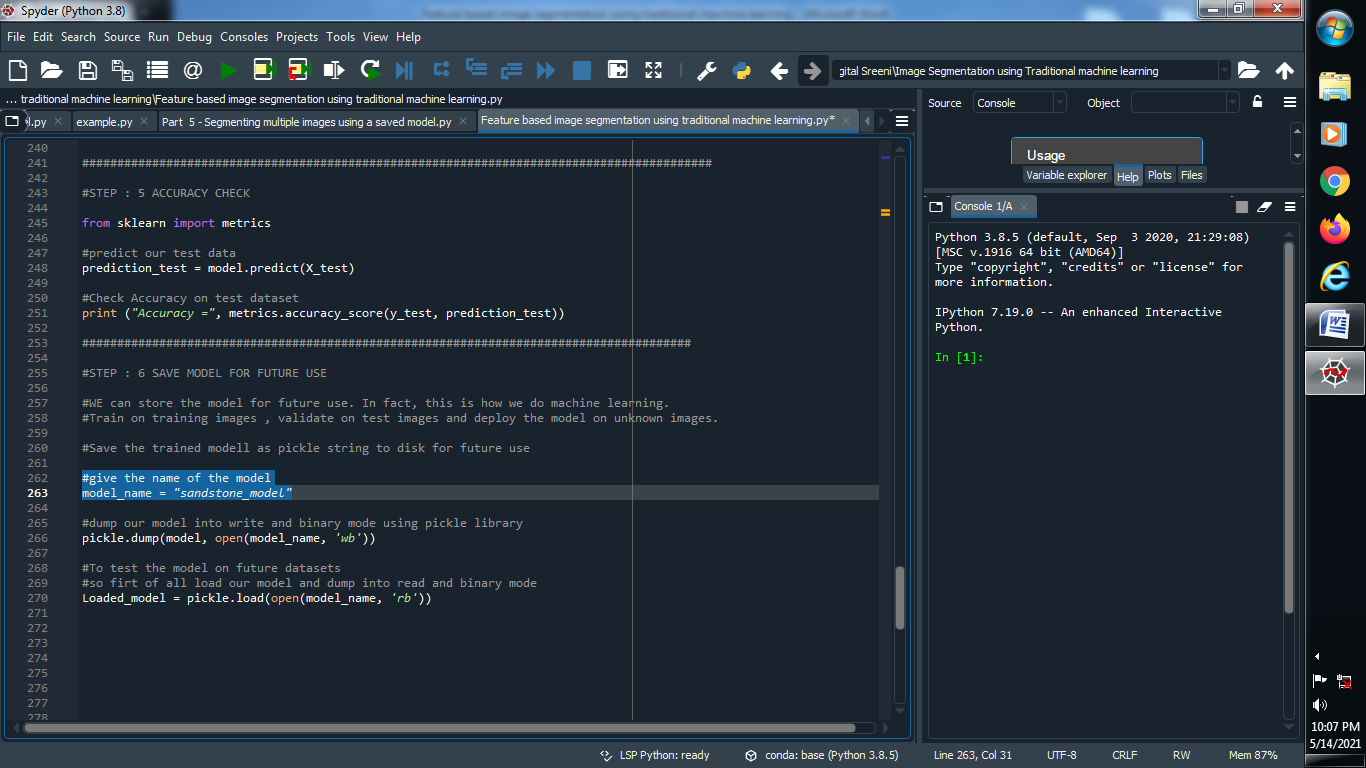
**(2) Check accuracy on test dataset :**

****

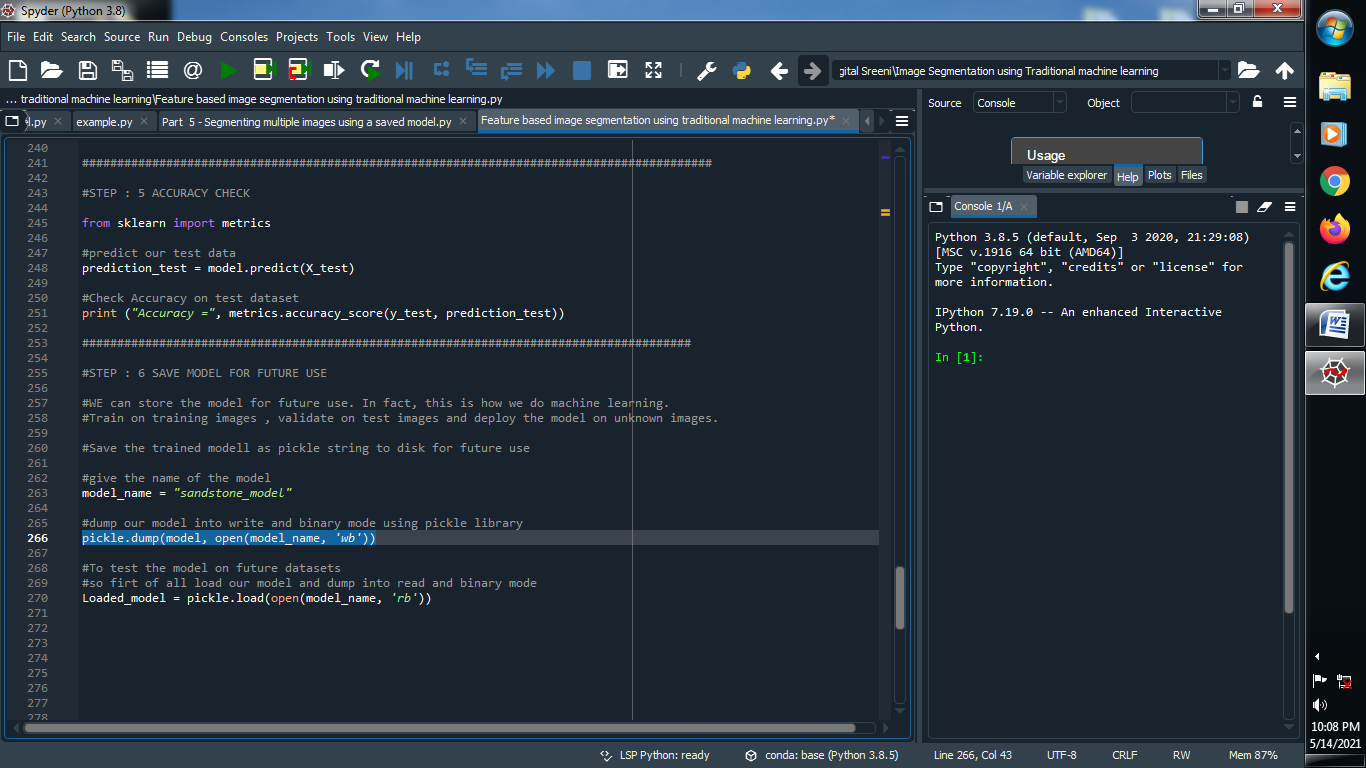
**####################################################################################################**

**STEP : 6 SAVE MODEL FOR FUTURE USE**

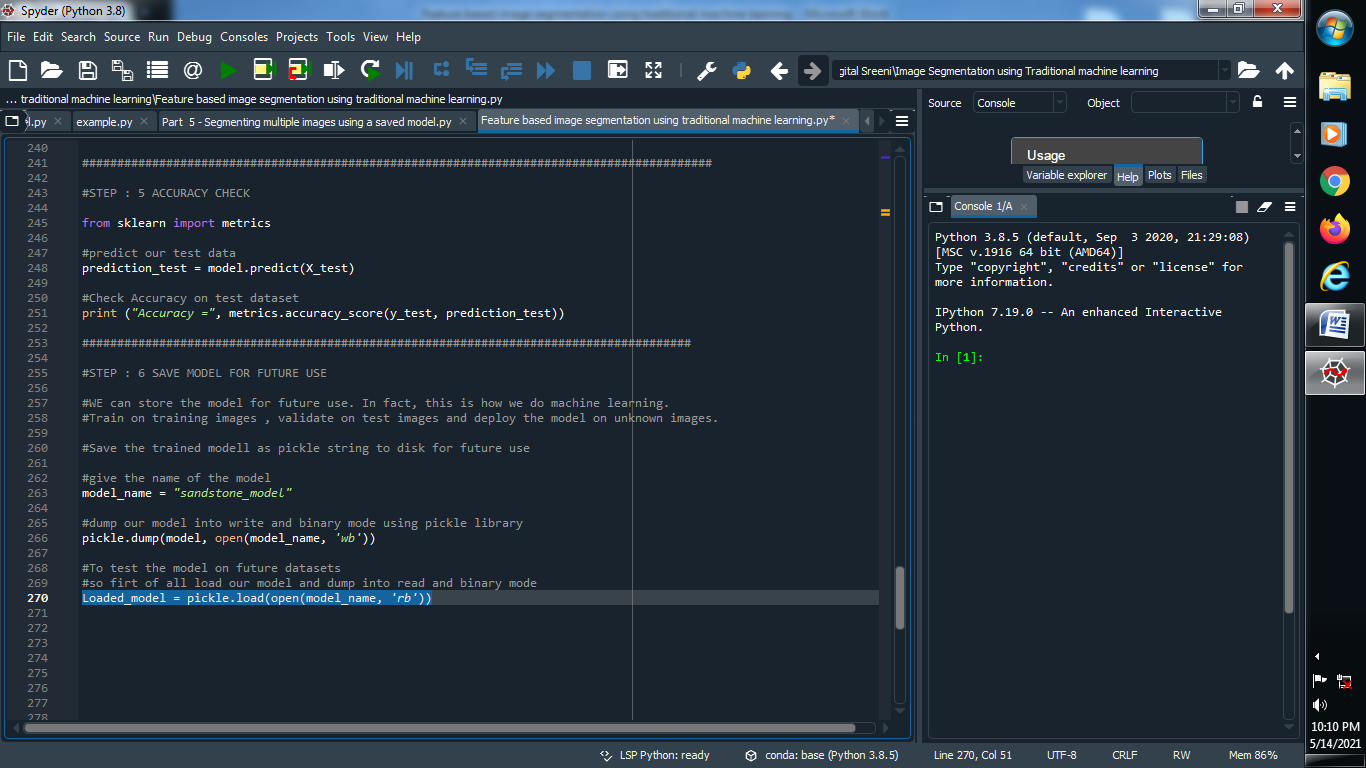
**(1) Give the model name :**

****

**(2) dump our model into write and binary mode using pickle library :**

****

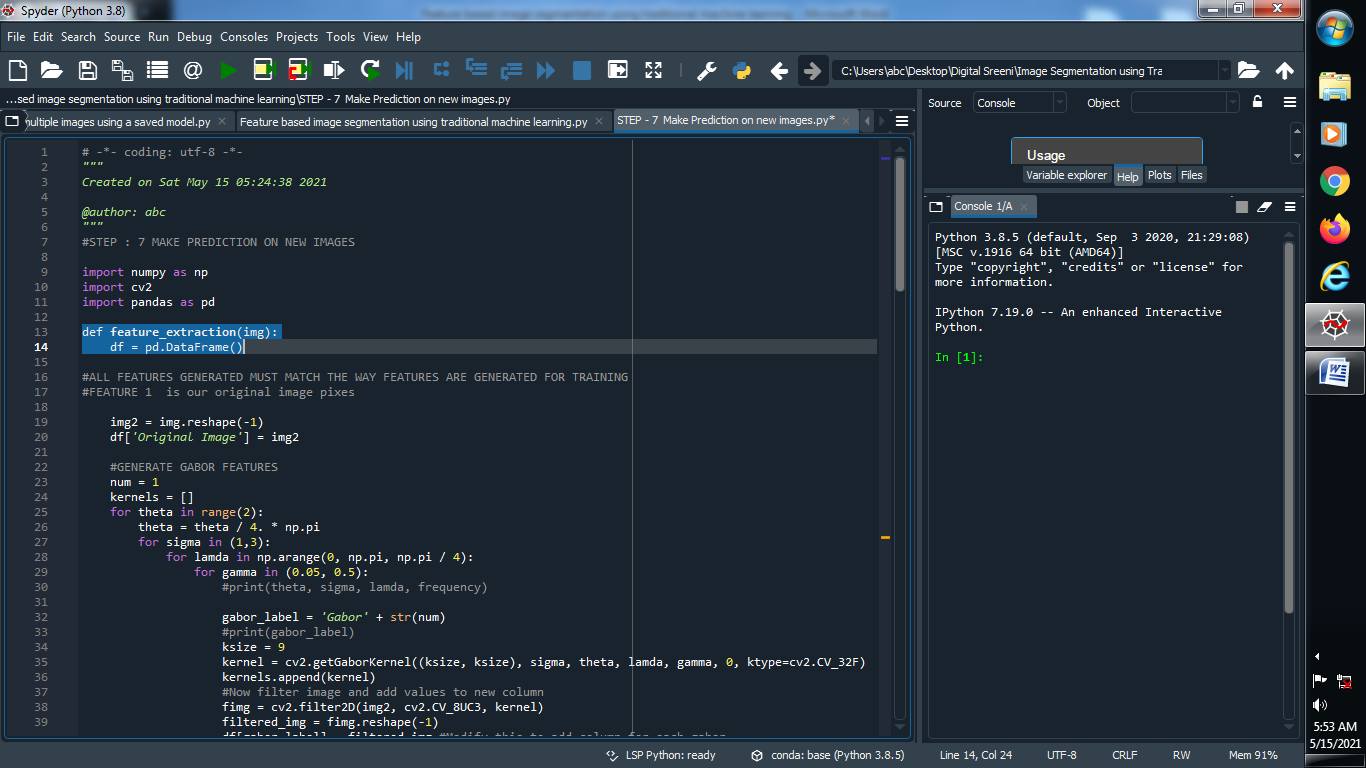
**(3) To test our model on future use so that we load our model**

****

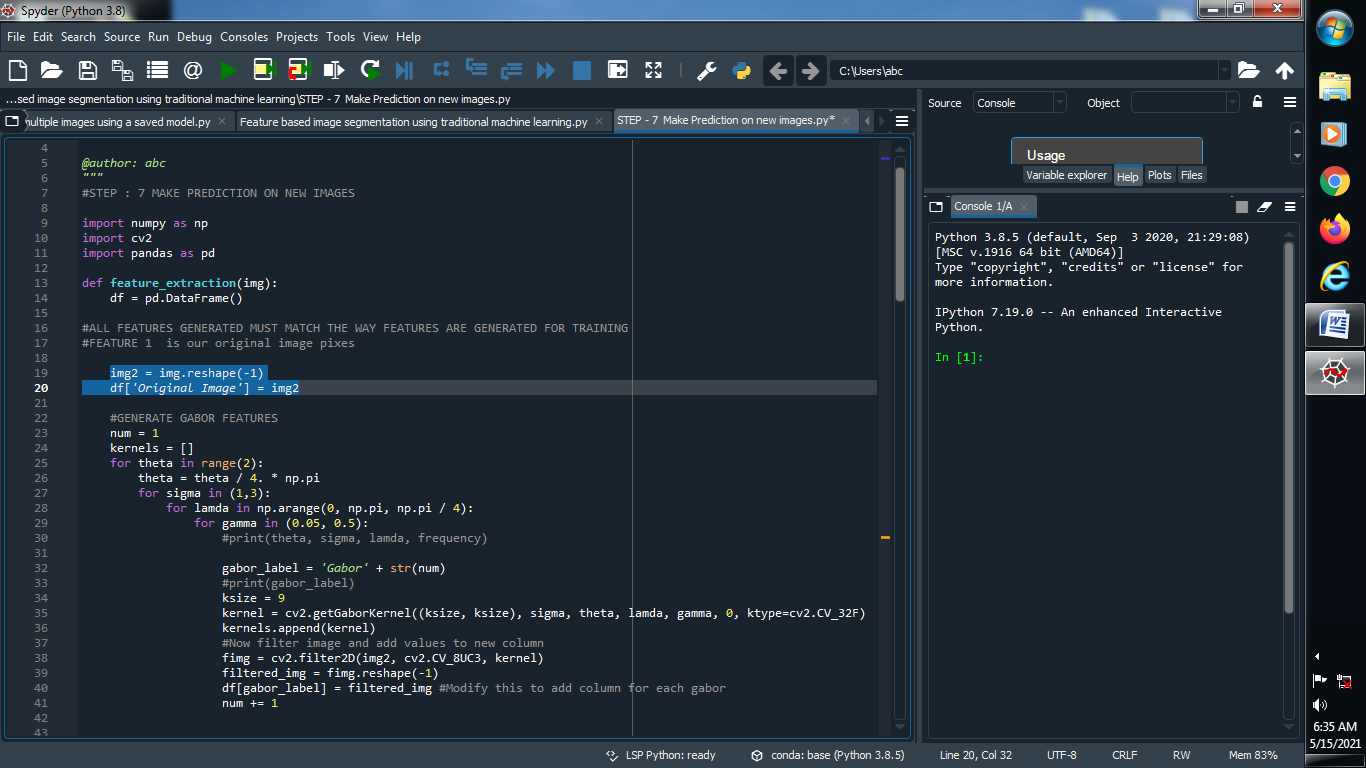
**########################################################################################**

**STEP : 7 MAKE PREDICTION ON NEW IMAGES**

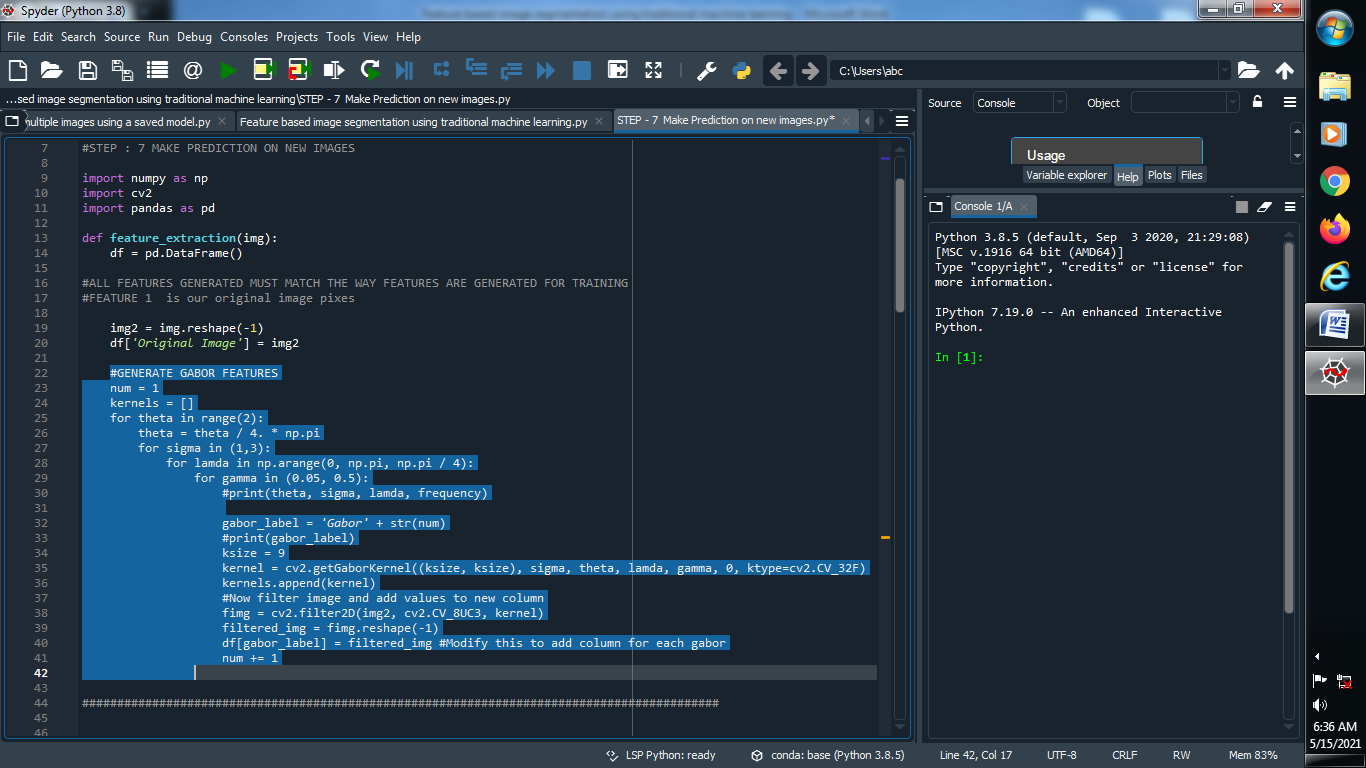
**(1) Define Function and Create data frame**

****

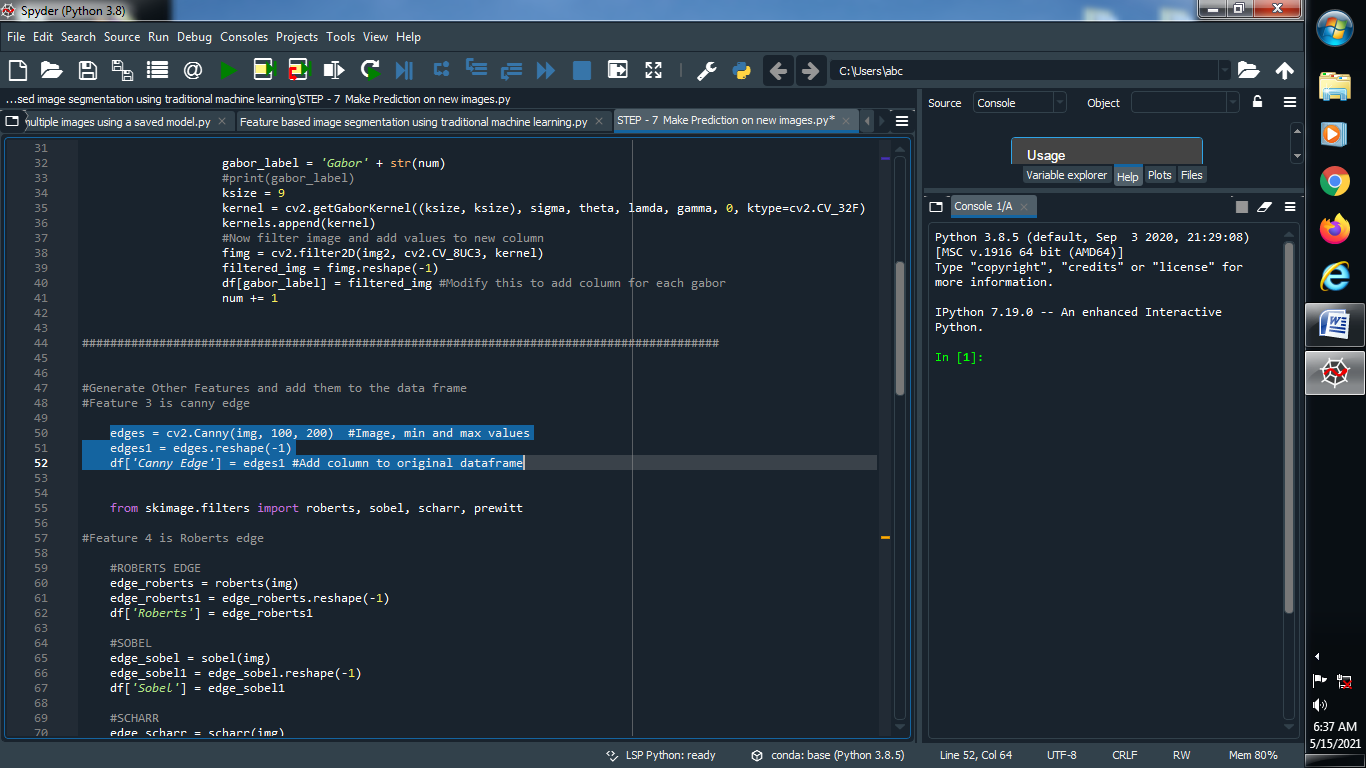
**(2) Reshape our image and add column name to our data frame**

****

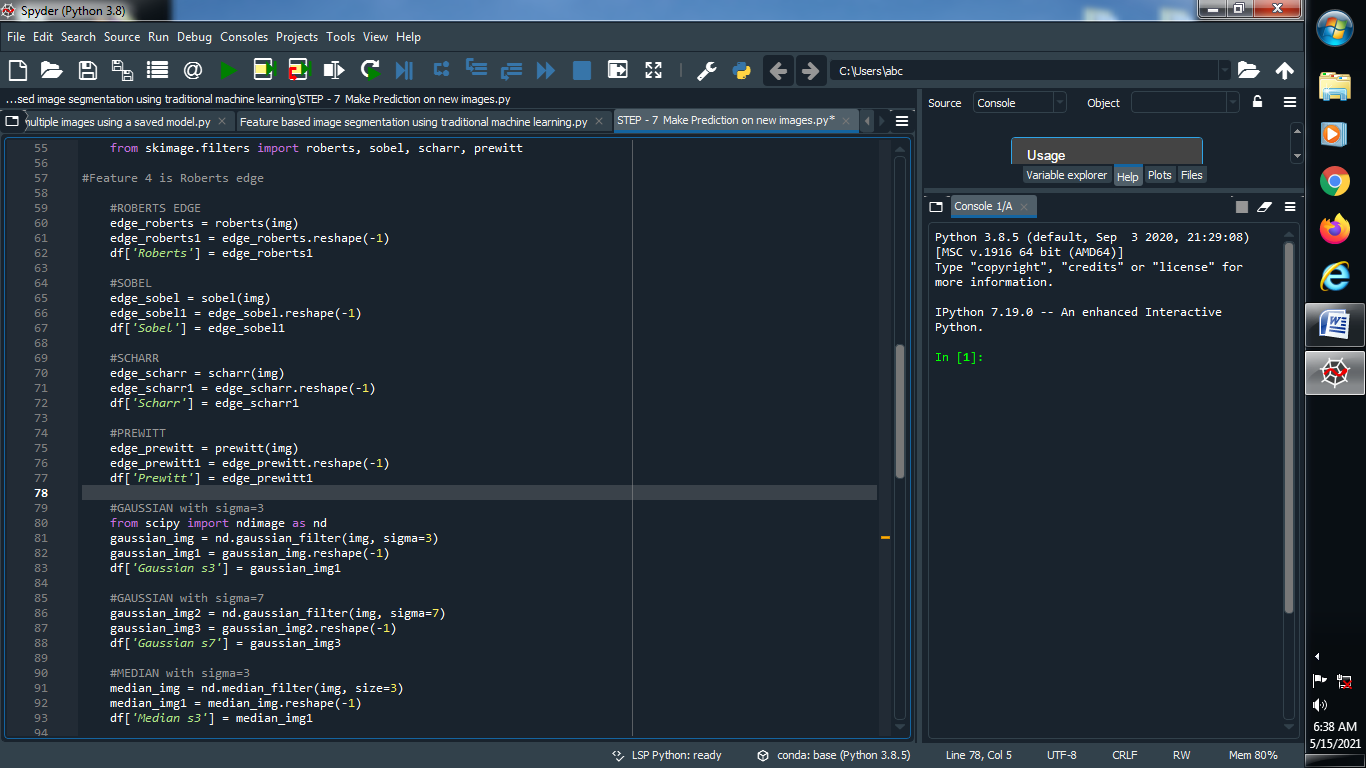
**(3) Generate gabor feature :**

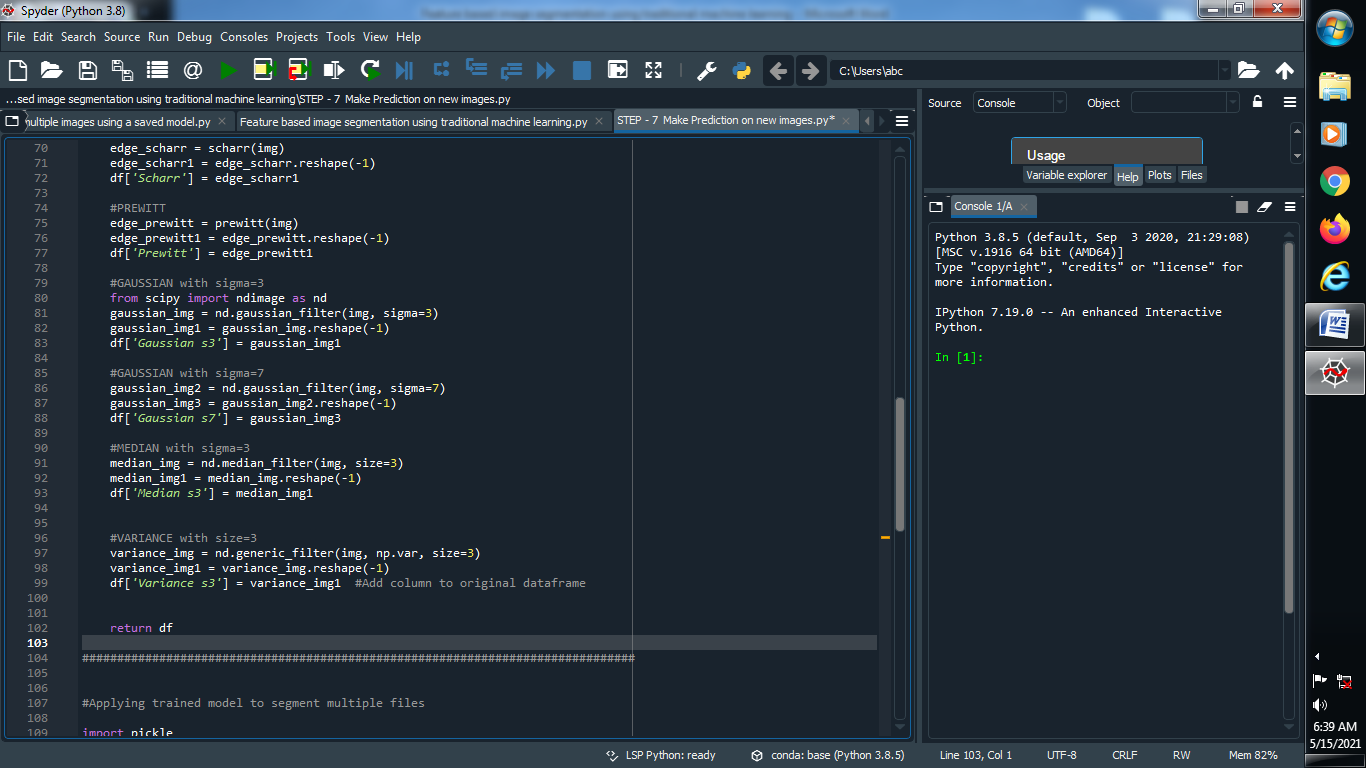
****

**(4) Generate Canny edge feature :**

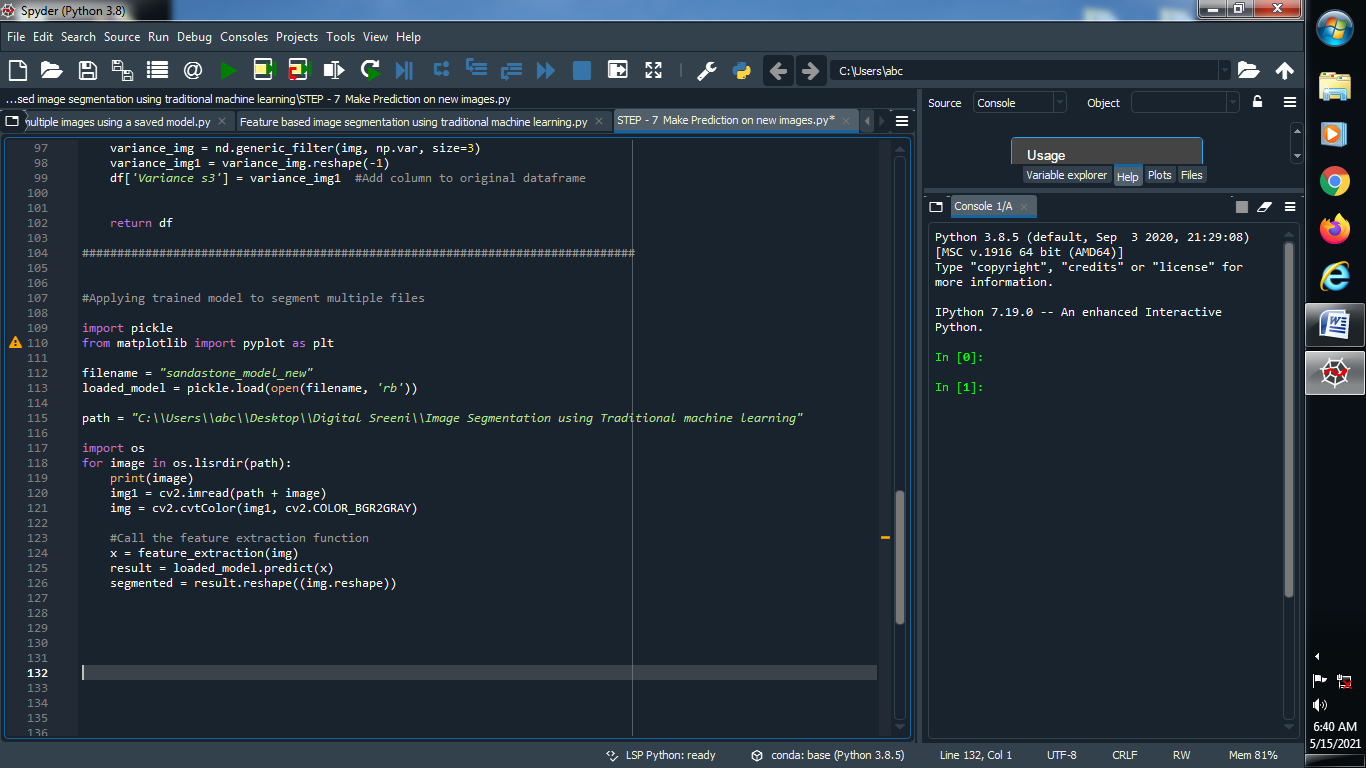
****

**(5) Add more features :**

****

****

**(6) Applying trained model to segment multiple files :**

****