

Image Classification Using Scikit-learn

Agenda

**Importing the Libraries Loading the Data**

**02**

**01**

**Data Preprocessing Implementing ML Algorithms**

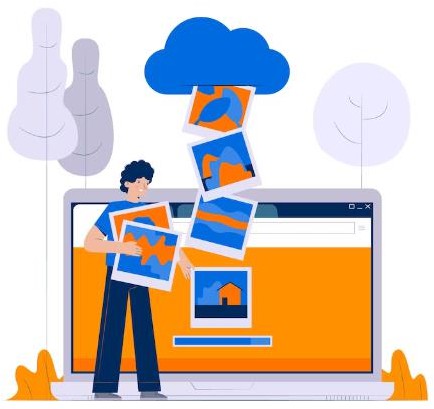
**04**

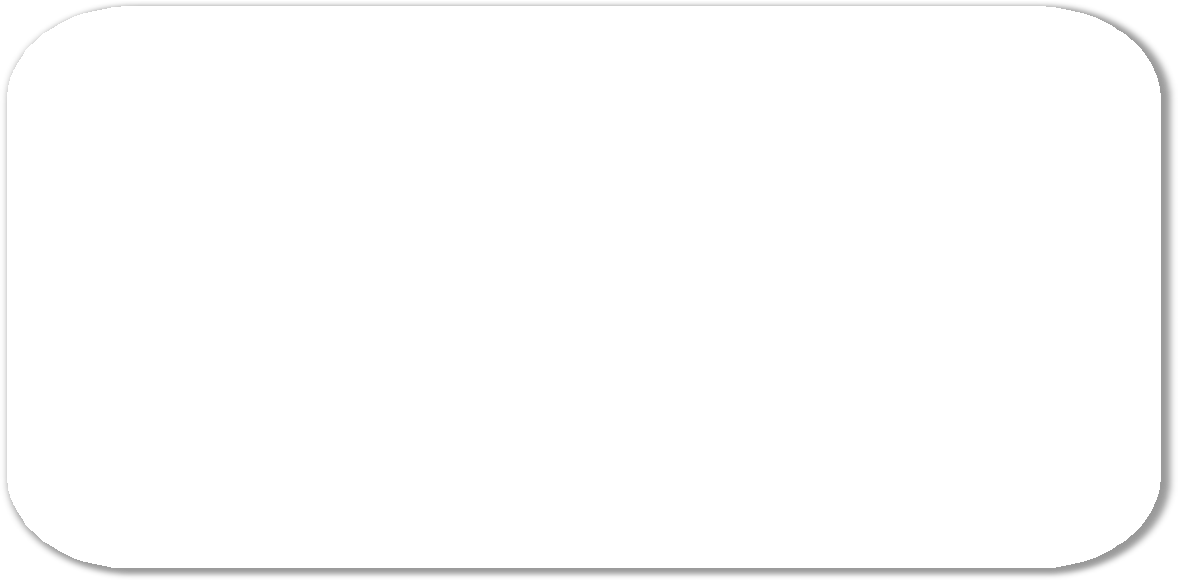
**03**

**Hyperparameter Tuning**

**05**

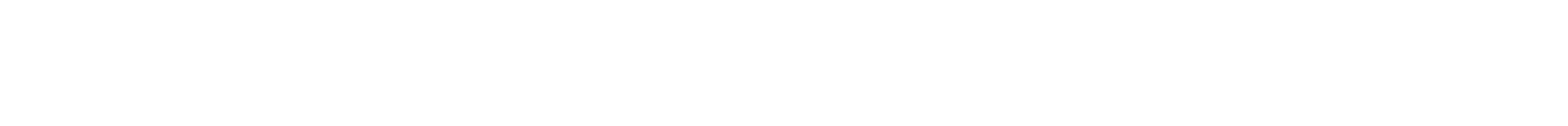
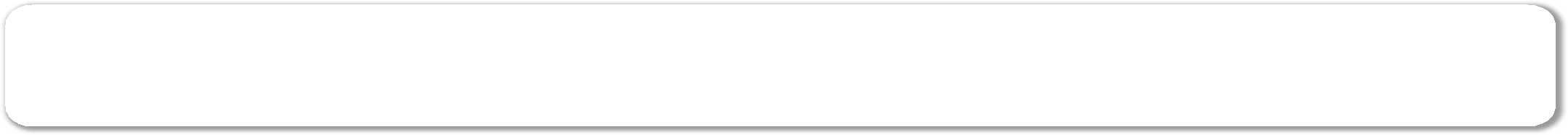
Problem Statement





I have to build an image classification model to classify 2 categories of images namely cat and dogs.

Dataset Information

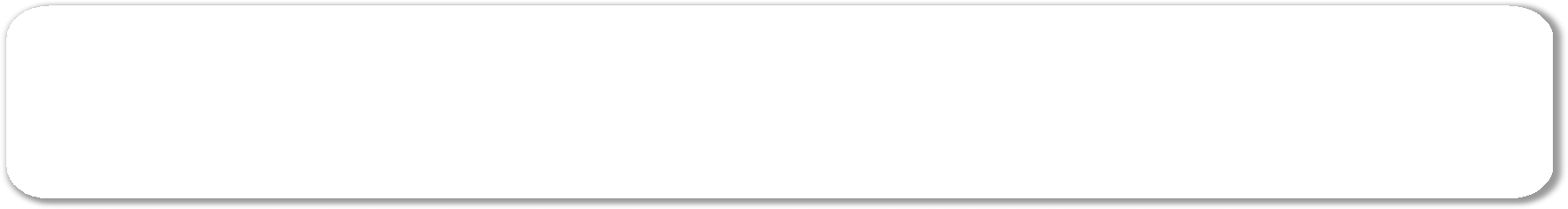


Cats and Dogs dataset contains 1000 images out of which 500 images are of cat

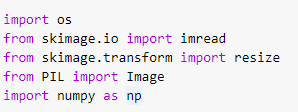
and same for the dog.



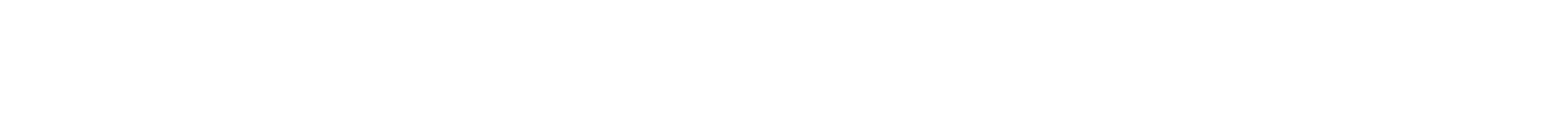
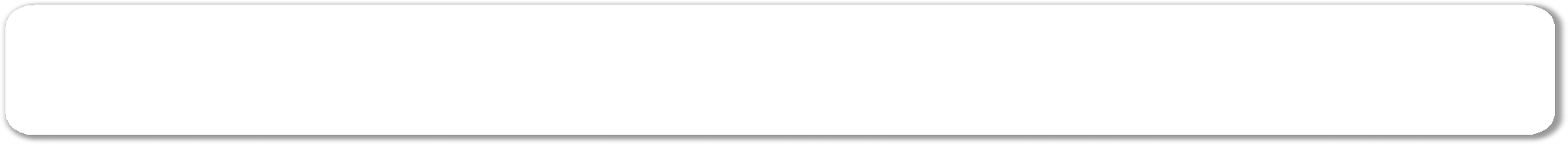
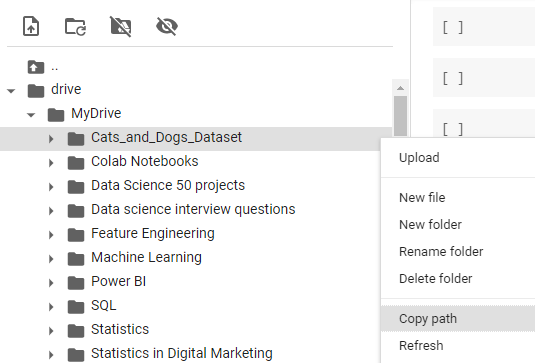
Importing the Libraries



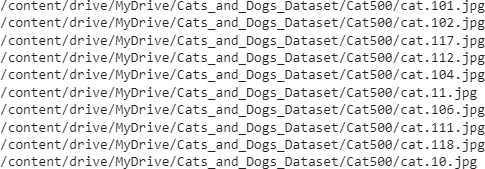
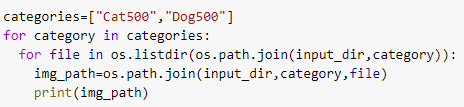
We start off this project by importing all the necessary libraries that will be required for the process.

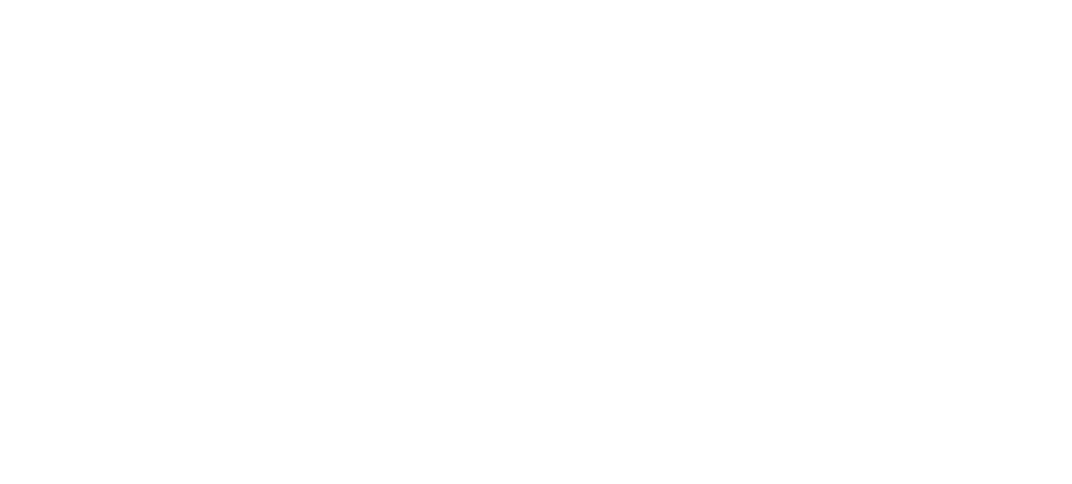
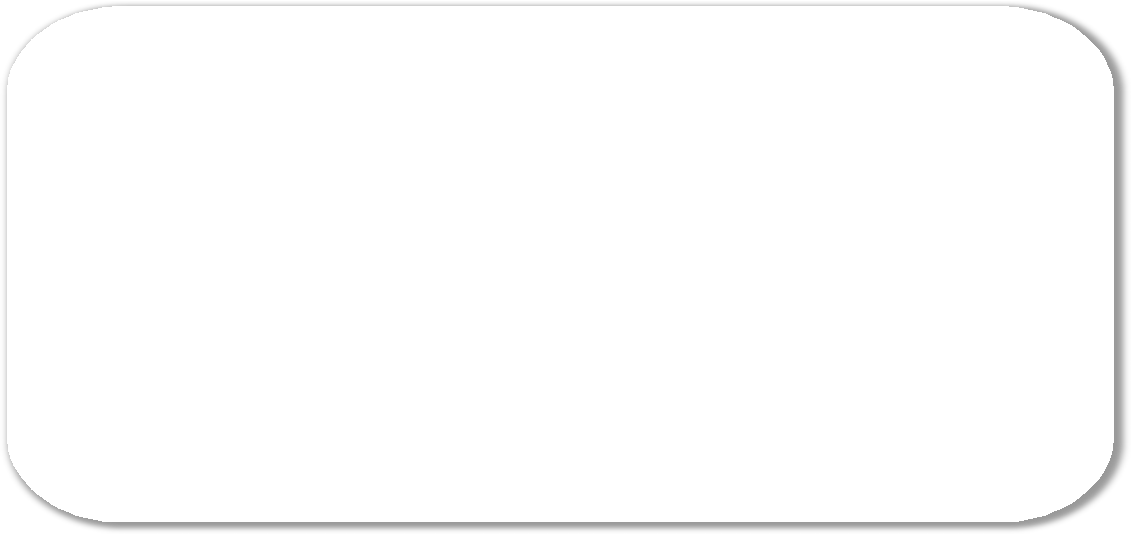


Loading the Image Data



Loading the image data by mounting the drive and providing the path of a folder stored in drive.

Displaying the Image



To display an image we need a path of each

image. Here we are using two functions

to handle path .

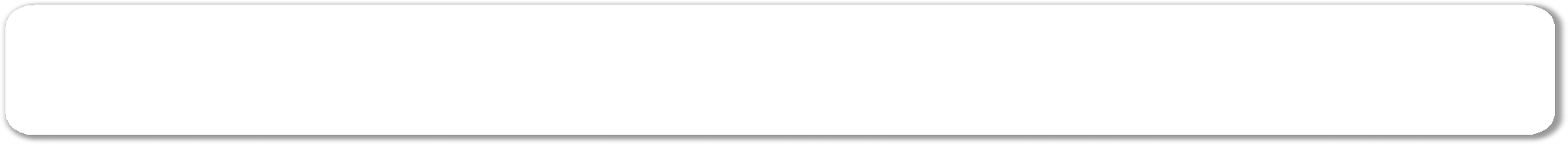
**os.listdir** - print a list of names of all the files

present in the specified path.

**os.path.join** - return a combined path by

merging arguments.

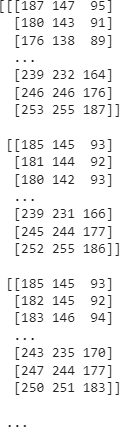
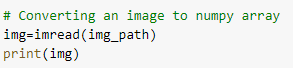
Displaying the Image

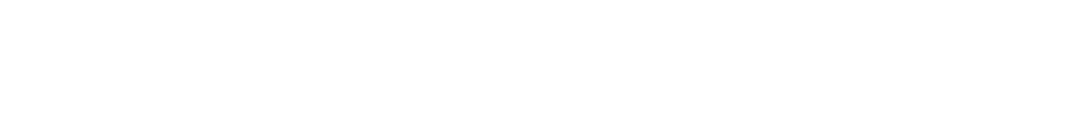
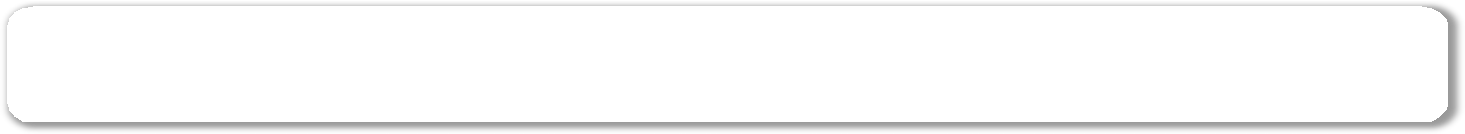


Taking any random path of the previous output to display the image.

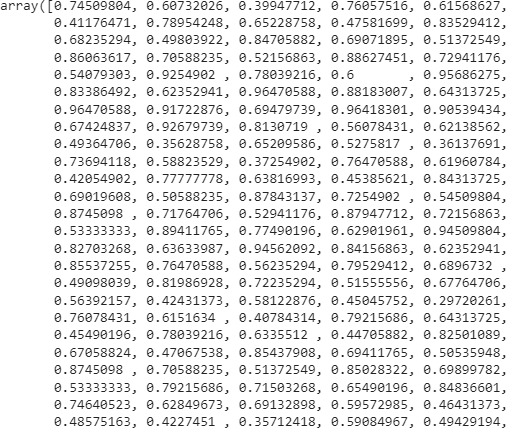


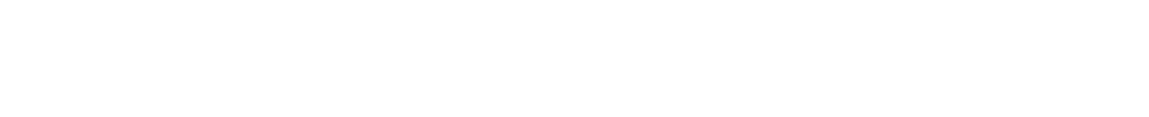
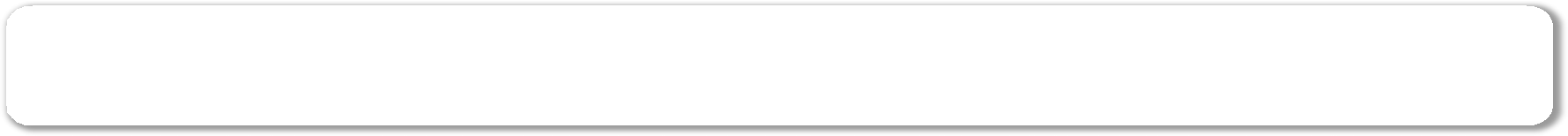


Image conversion to array

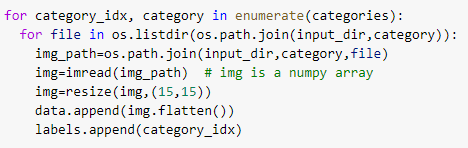
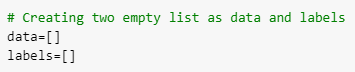


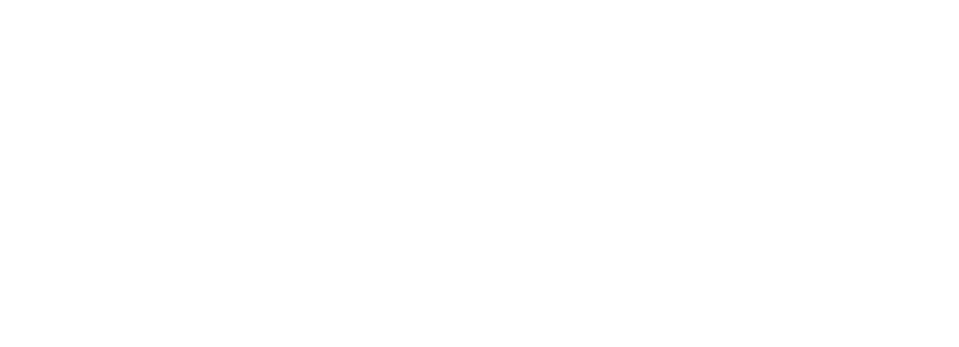
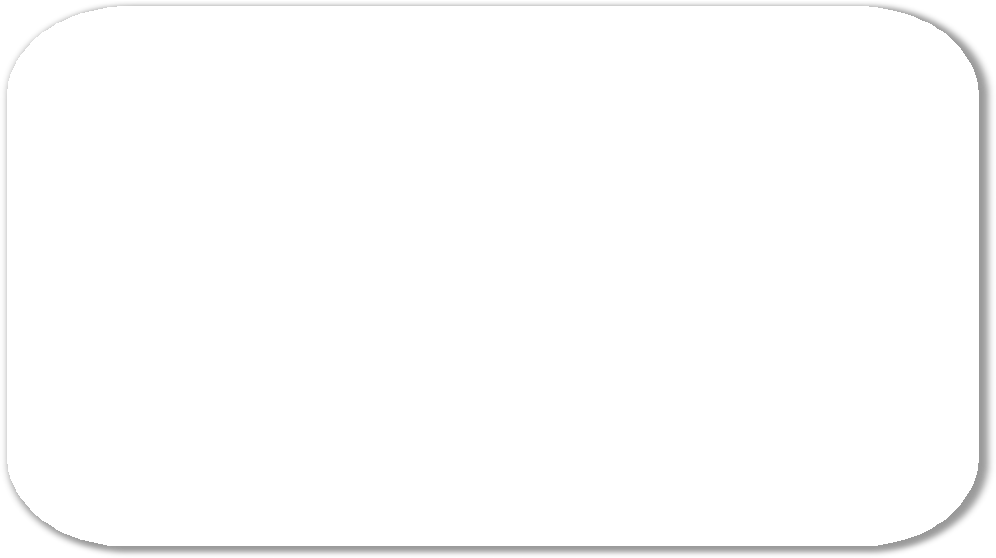
Converting the image to numpy array

Image conversion to array



Resizing and flattening the numpy array.

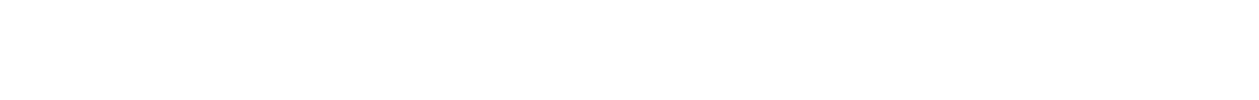
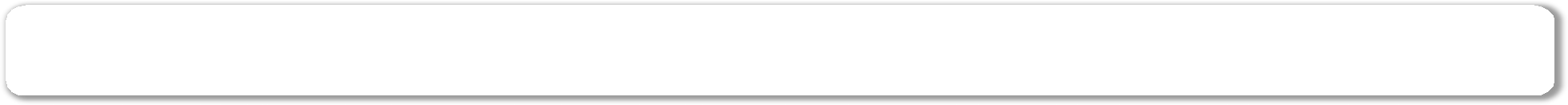
Image conversion to array



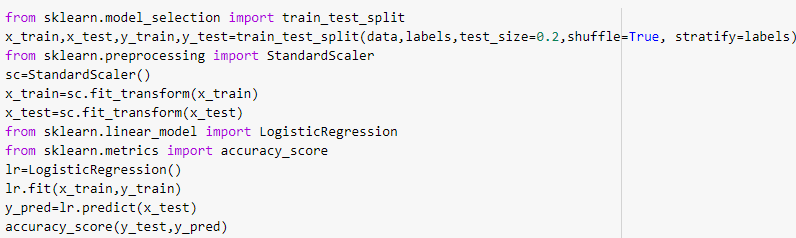
Converting all the images into a 1-D array and storing it into data variable. Here Label contains two categories i.e 0 (Cat) and

1 (Dog)

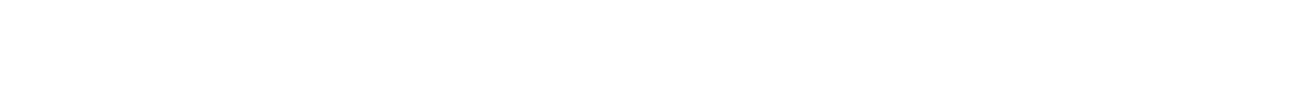
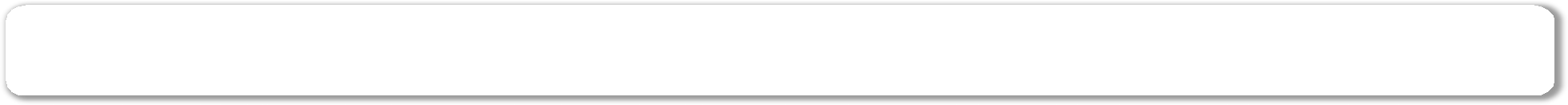
Logistic Regression



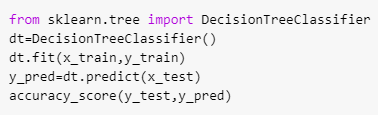
Accuracy obtained by logistic regression is 54 percent.



Decision Tree

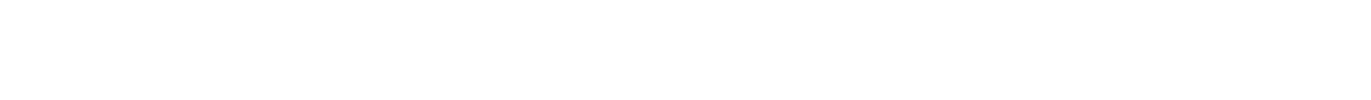
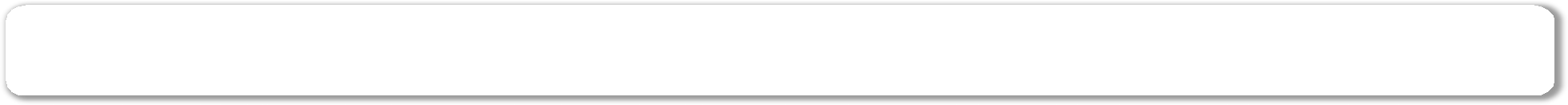


Accuracy obtained by decision tree model is 56 percent.

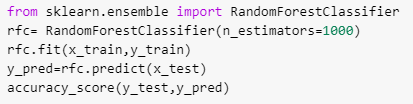




Random Forest

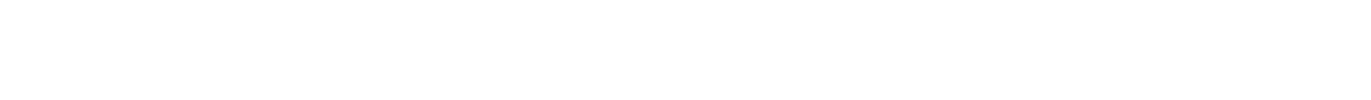
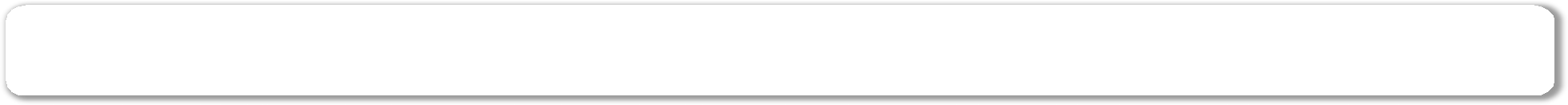


Accuracy obtained by random forest model is 64.5 percent.

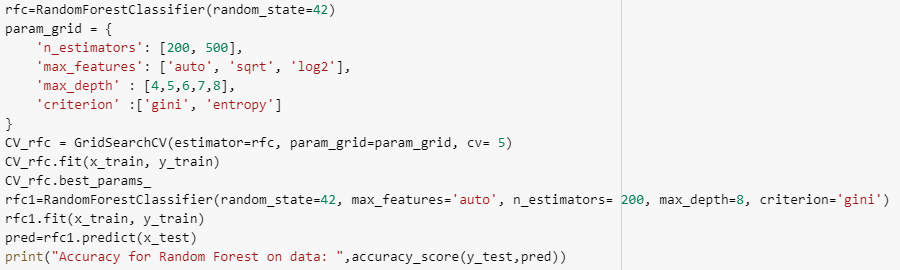




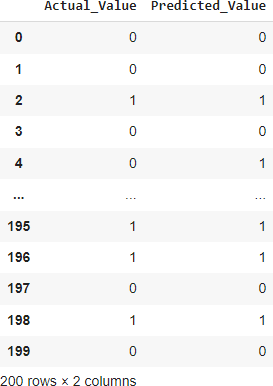
Hyperparameter Tuning - Random Forest

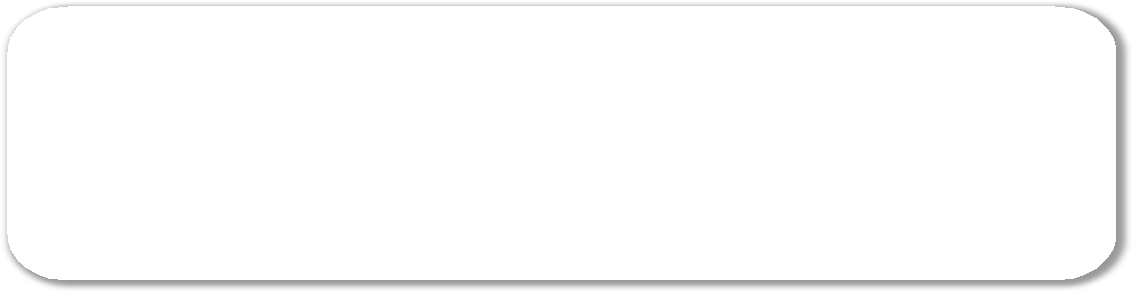


Accuracy obtained by random forest model is 65.5 percent.

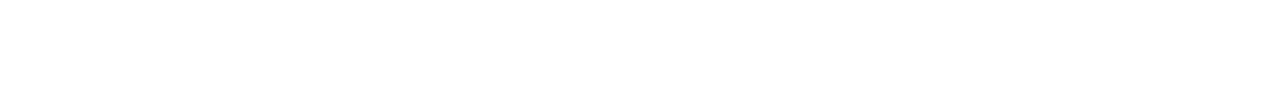
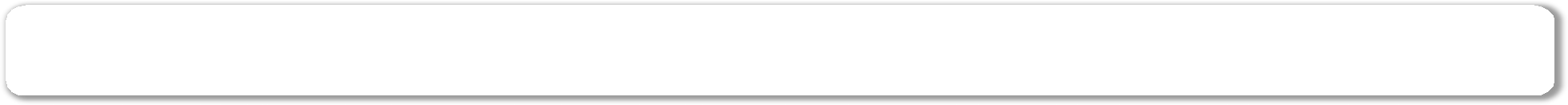




Hyperparameter Tuning - Random Forest



**Note -** I have trained 1000 images of cats and dogs. Increase in sample image will lead to increase in accuracy.



Creating the dataframe for actual and predicted value.