

## Sprint 2

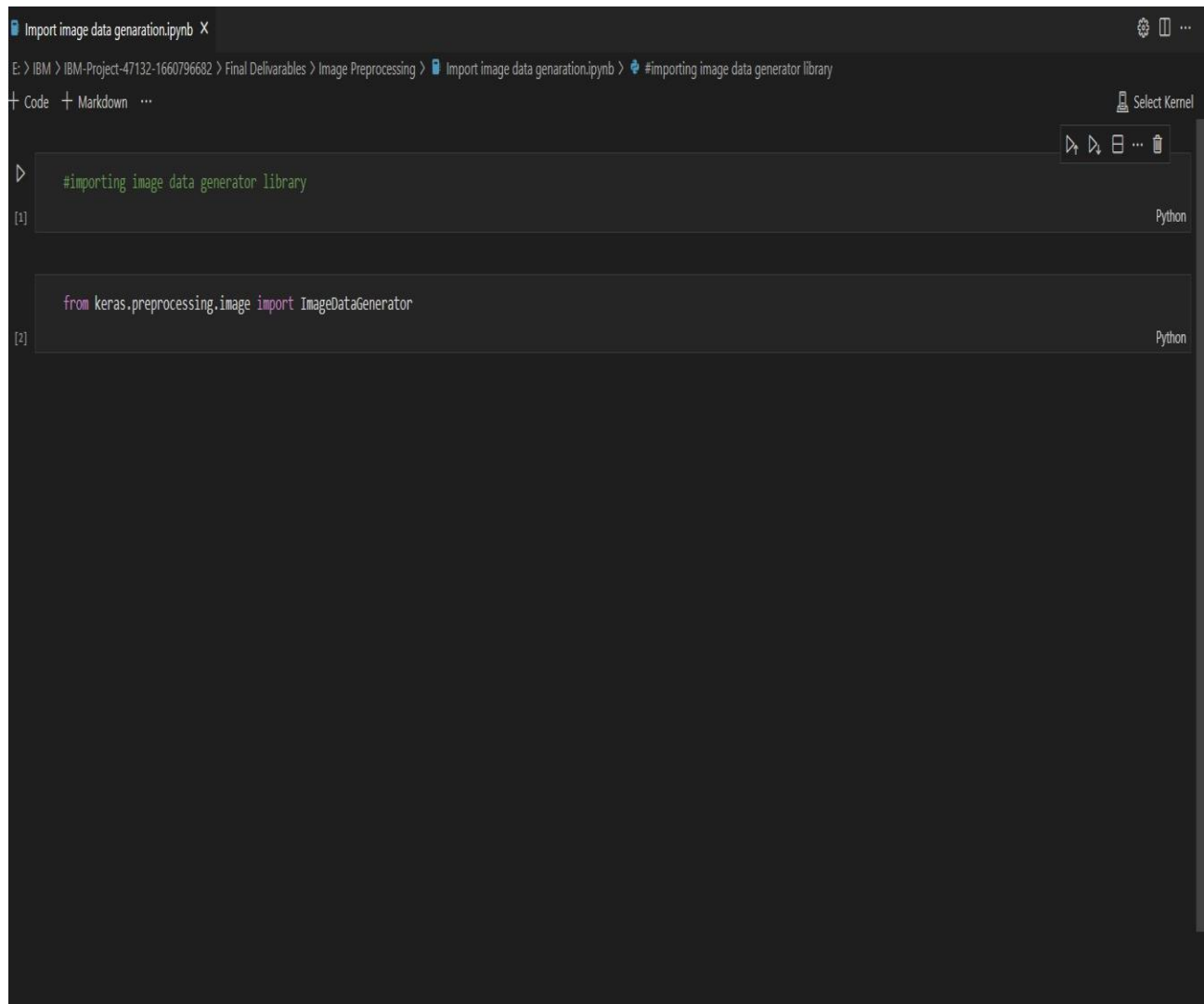
Team ID	PNT2022TMID21298
Project Name	Natural Disasters Intensity Analysis and Classification using Artificial Intelligence
Maximum Marks	20 Marks

### Image Processing:



Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the Image Data Generator class. Let us import the Image Data Generator class from Keras.



```
Import image data generation.ipynb X
E: > IBM > IBM-Project-47132-1660796682 > Final Deliverables > Image Preprocessing > Import image data generation.ipynb > #importing image data generator library
+ Code + Markdown ...
Select Kernel
Python
Python
```

The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar at the top reads "Import image data generation.ipynb X". Below it, the file path is displayed: "E: > IBM > IBM-Project-47132-1660796682 > Final Deliverables > Image Preprocessing > Import image data generation.ipynb > #importing image data generator library". On the left, there are tabs for "+ Code" and "+ Markdown". On the right, there is a "Select Kernel" button and a toolbar with icons for running, saving, and other actions. The notebook contains two code cells. The first cell, labeled "[1]", contains the comment "#importing image data generator library" and is marked as "Python". The second cell, labeled "[2]", contains the code "from keras.preprocessing.image import ImageDataGenerator" and is also marked as "Python".

- **Configure image Data Generator class**

Image Data Generator class is instantiated and the configuration for the types of data augmentation

There are five main types of data augmentation techniques for image data; specifically:

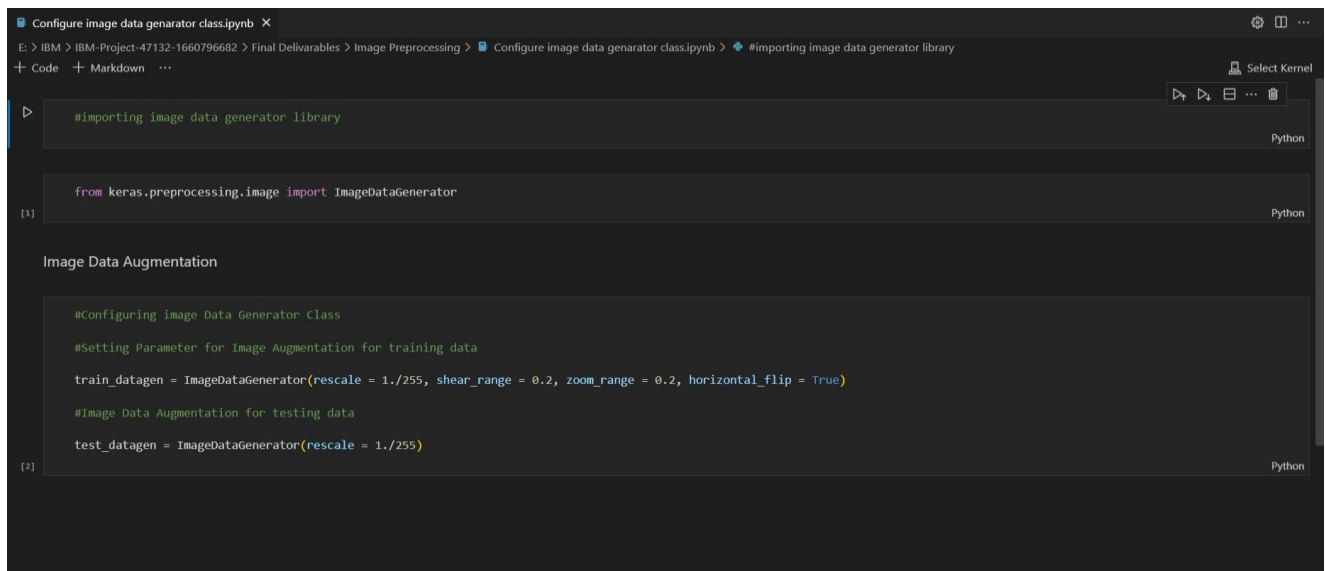
Image shifts via the width shift range and height shift range arguments.

The image flips via the horizontal flip and vertical flip arguments.

Image rotations via the rotation range argument

Image brightness via the brightness range argument.

An instance of the Image Data Generator class can be constructed for train and test.



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads "Configure image data generator class.ipynb". The breadcrumb navigation shows the path: "E > IBM > IBM-Project-47132-1660796682 > Final Deliverables > Image Preprocessing > Configure image data generator class.ipynb". The notebook contains two code cells. The first cell, labeled "[1]", imports the ImageDataGenerator class from keras.preprocessing.image. The second cell, labeled "[2]", is titled "Image Data Augmentation" and contains comments and code for configuring the training and testing data generators. The training generator is configured with rescale=1./255, shear\_range=0.2, zoom\_range=0.2, and horizontal\_flip=True. The testing generator is configured with rescale=1./255.

```
#importing image data generator library

from keras.preprocessing.image import ImageDataGenerator

Image Data Augmentation

#Configuring image Data Generator Class
#Setting Parameter for Image Augmentation for training data
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
#Image Data Augmentation for testing data
test_datagen = ImageDataGenerator(rescale = 1./255)
```

- **Apply Image Data Generator Functionality to Trainset and Test set**

Let us apply Image Data Generator functionality to Trainset and Test set by using the following code

For Training set using flow from directory function.

This function will return batches of images from the subdirectories Cyclone, Earthquake, Flood, Wildfire together with labels 0 to 3{Cyclone: 0, Earthquake: 1, Flood: 2, Wildfire: 3}

```
In [15]: #extract zip file
         !unzip '/content/drive/MyDrive/dataset.zip'

Archive: /content/drive/MyDrive/dataset.zip
  inflating: dataset/readme.txt
   creating: dataset/test_set/
   creating: dataset/test_set/Cyclone/
  inflating: dataset/test_set/Cyclone/867.jpg
  inflating: dataset/test_set/Cyclone/868.jpg
  inflating: dataset/test_set/Cyclone/869.jpg
  inflating: dataset/test_set/Cyclone/870.jpg
  inflating: dataset/test_set/Cyclone/871.jpg
  inflating: dataset/test_set/Cyclone/872.jpg
  inflating: dataset/test_set/Cyclone/873.jpg
  inflating: dataset/test_set/Cyclone/874.jpg
  inflating: dataset/test_set/Cyclone/875.jpg
  inflating: dataset/test_set/Cyclone/876.jpg
  inflating: dataset/test_set/Cyclone/877.jpg
  inflating: dataset/test_set/Cyclone/878.jpg
  inflating: dataset/test_set/Cyclone/879.jpg
  inflating: dataset/test_set/Cyclone/880.jpg
  inflating: dataset/test_set/Cyclone/881.jpg
  inflating: dataset/test_set/Cyclone/882.jpg
  inflating: dataset/test_set/Cyclone/883.jpg
  inflating: dataset/test_set/Cyclone/884.jpg
  inflating: dataset/test_set/Cyclone/885.jpg
  inflating: dataset/test_set/Cyclone/886.jpg
  inflating: dataset/test_set/Cyclone/887.jpg
  inflating: dataset/test_set/Cyclone/888.jpg
  inflating: dataset/test_set/Cyclone/889.jpg
  inflating: dataset/test_set/Cyclone/890.jpg
  inflating: dataset/test_set/Cyclone/891.jpg
  inflating: dataset/test_set/Cyclone/892.jpg
  inflating: dataset/test_set/Cyclone/893.jpg
  inflating: dataset/test_set/Cyclone/894.jpg
  inflating: dataset/test_set/Cyclone/895.jpg
  inflating: dataset/test_set/Cyclone/896.jpg
  inflating: dataset/test_set/Cyclone/897.jpg
  inflating: dataset/test_set/Cyclone/898.jpg
  inflating: dataset/test_set/Cyclone/899.jpg
  inflating: dataset/test_set/Cyclone/900.jpg
```



```
inflating: dataset/train_set/Wildfire/91.jpg
inflating: dataset/train_set/Wildfire/98.jpg
inflating: dataset/train_set/Wildfire/99.jpg
```

```
In [16]: #importing image data generator library
```

```
In [17]: from keras.preprocessing.image import ImageDataGenerator
```

### Image Data Augmentation

```
In [18]: #Configuring image Data Generator Class

#Setting Parameter for Image Augmentation for training data

train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)

#Image Data Augmentation for testing data

test_datagen = ImageDataGenerator(rescale = 1./255)
```

## Apply ImageDataGenerator Functionality To Trainset And Testset

```
In [19]: #Performing data augmentation to train data

x_train = train_datagen.flow_from_directory('/content/drive/MyDrive/dataset/dataset/train_set', target_size = (64,64), batch_size = 5, color_mode = 'r

#performing data augmentation to test data

x_test = test_datagen.flow_from_directory('/content/drive/MyDrive/dataset/dataset/test_set', target_size = (64,64), batch_size = 5, color_mode = 'rgb'
```

```
Found 742 images belonging to 4 classes.
Found 198 images belonging to 4 classes.
```