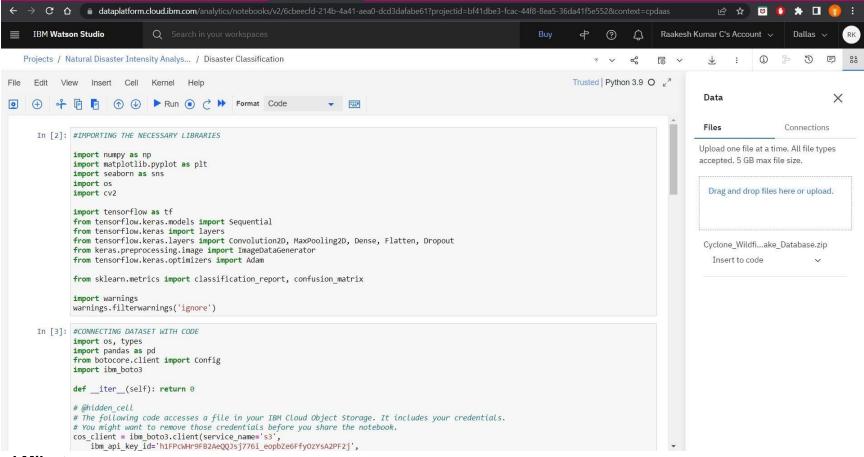
Project Development Phase Sprint - I

Date	29 October 2022
Team ID	PNT2022TMID25575
Project Name	Natural Disasters Intensity Analysis And Classification Using Artificial Intelligence
Maximum Marks	4 Marks

Jupyter Notebook asset in IBM Watson Studio feature of IBM Cloud:



Sprint - I Milestones:

1. Import the Necessary Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import cos
import cv2

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Dense, Flatten, Dropout
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import classification_report, confusion_matrix
import warnings
warnings.filterwarnings('ignore')
```

2. Upload and Connect Dataset with notebook

```
In [3]: #CONNECTING DATASET WITH CODE
        import os, types
        import pandas as pd
        from botocore.client import Config
        import ibm boto3
        def iter (self): return 0
        # @hidden cell
        # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
        # You might want to remove those credentials before you share the notebook.
        cos client = ibm boto3.client(service name='s3',
            ibm api key id='h1FPcWHr9FB2AeQQJsj776i eopbZe6FfyOzYsA2PF2j',
            ibm auth endpoint="https://iam.cloud.ibm.com/oidc/token",
            config=Config(signature_version='oauth'),
            endpoint url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
        bucket = 'naturaldisasterintensityanalysisa-donotdelete-pr-pwiuxy2i5hirv2'
        object key = 'Cyclone Wildfire Flood Earthquake Database.zip'
        streaming body 2 = cos client.get object(Bucket=bucket, Key=object key)['Body']
        # Your data file was loaded into a botocore.response.StreamingBody object.
        # Please read the documentation of ibm boto3 and pandas to learn more about the possibilities to load the data.
        # ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
        # pandas documentation: http://pandas.pydata.org/
```

3. Extracting the Dataset using BytesIO unzip function

4. Listing out the Disaster Classes

```
In [8]: #LISTING OUT THE DISASTER CLASSES
         for i in os.listdir(dir):
             print(i)
            readme.txt
            Earthquake
           Cyclone
            Wildfire
            Flood
In [9]: path=os.path.join(dir, 'readme.txt')
         os.remove(path)
In [10]: for i in os.listdir(dir):
             print(i)
            Earthquake
            Cyclone
            Wildfire
            Flood
```

5. Configuring ImageDataGenerator Class

```
In [18]: dir=r'/home/wsuser/work/Cyclone_Wildfire_Flood_Earthquake_Database'
#CONFIGURING THE ImageDataGenerator CLASS
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

6. Split the dataset into training, testing, and validation data

```
In [11]: pip install split_folders
            Collecting split folders
              Downloading split folders-0.5.1-py3-none-any.whl (8.4 kB)
            Installing collected packages: split-folders
            Successfully installed split-folders-0.5.1
            Note: you may need to restart the kernel to use updated packages.
In [12]: #SPLIT THE DATASET INTO TRAINING, TESTING AND VALIDATION DATA
         import splitfolders
         splitfolders.ratio(dir,output="dataset",seed=42,ratio=(.7,.2,.1),group prefix=None)
            Copying files: 4428 files [00:02, 1678.84 files/s]
In [13]: for i in os.listdir(dir):
             print(i)
            Earthquake
            Cyclone
            Wildfire
            Flood
In [14]: dir1=r'/home/wsuser/work/dataset'
In [15]: for i in os.listdir(dir1):
             print(i)
            val
            train
            test
```

7. Apply ImageDataGenerator Functionality to train set and test/validation set

Found 3097 images belonging to 4 classes. Found 884 images belonging to 4 classes.

70% of the images are for training, 20% of Images are for validation and 10% remaining is for Testing

```
In [26]: #70% of images goes to training, #20% of the total to validation, and #10% remaining goes to testing data
x_test=test_datagen.flow_from_directory(r"/home/wsuser/work/dataset/test")
```

Found 447 images belonging to 4 classes.