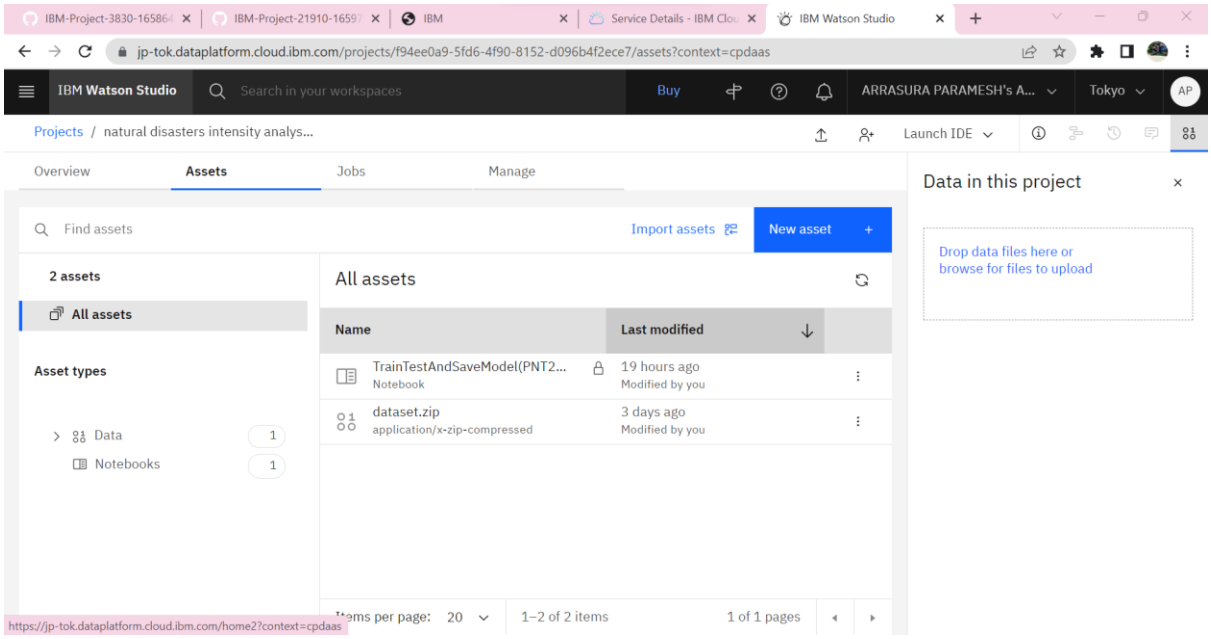
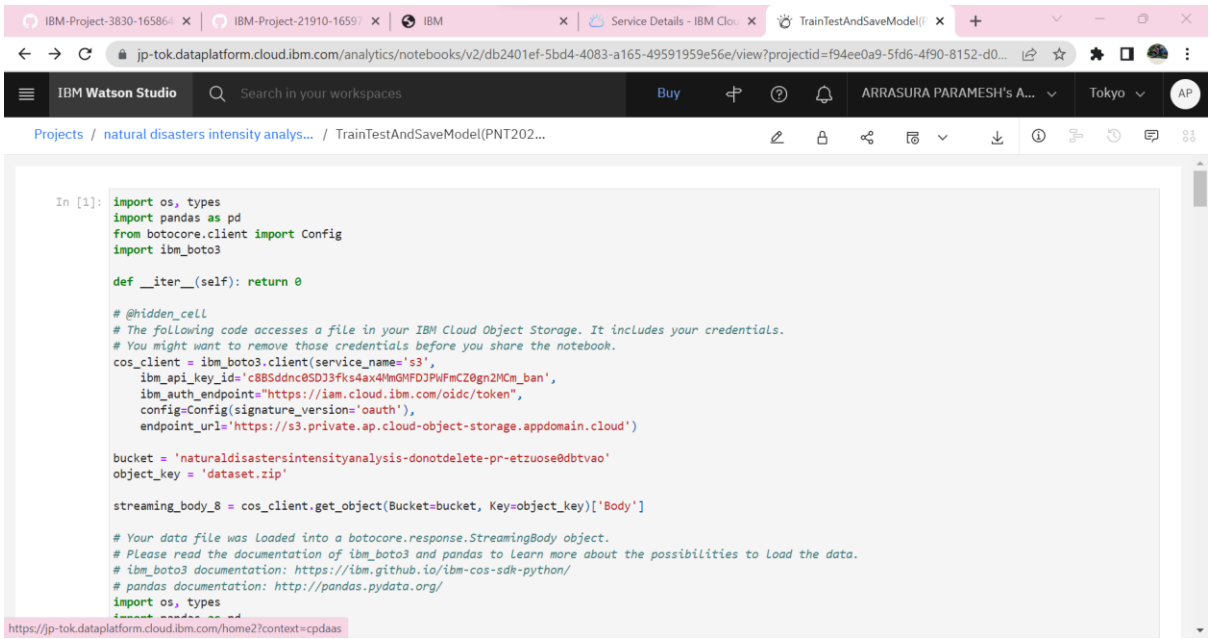
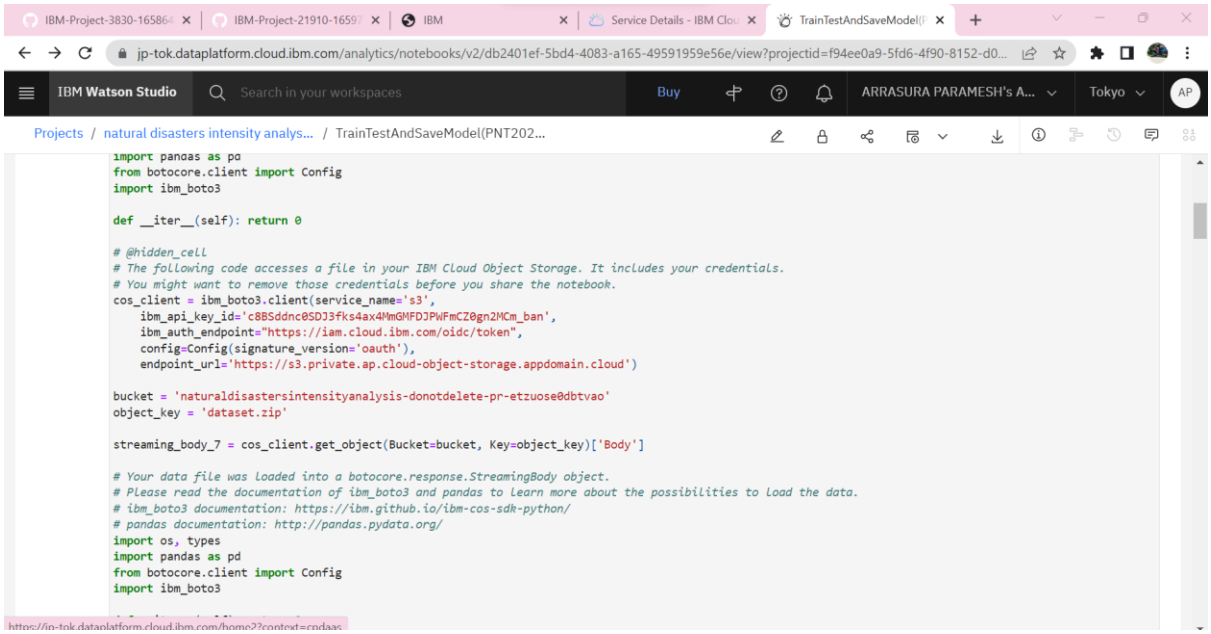
NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

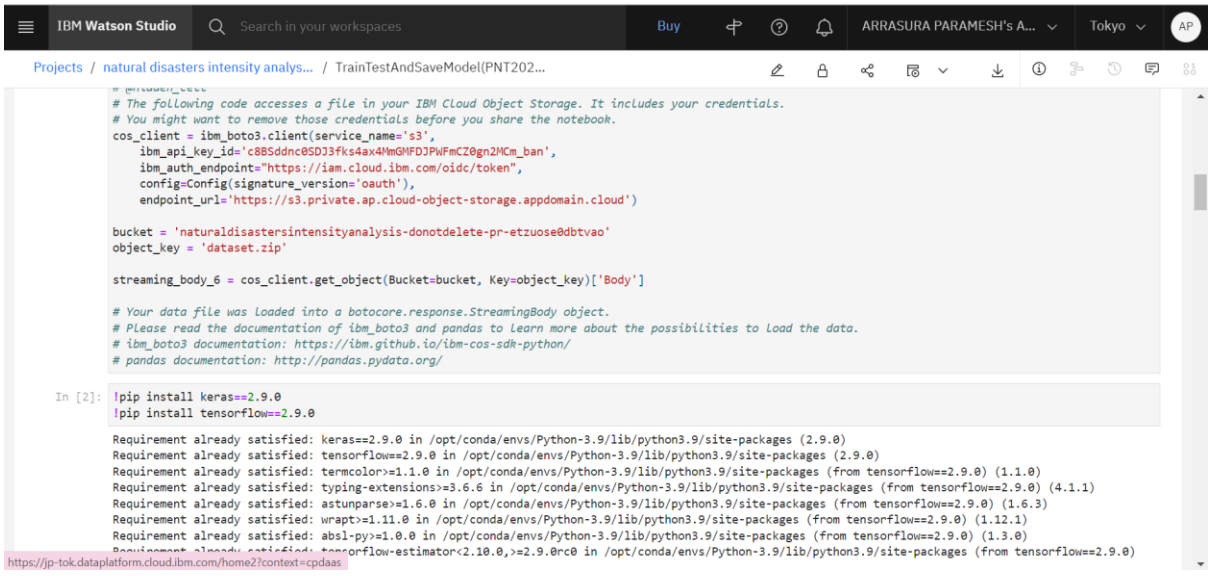
PNT2022TMID06841

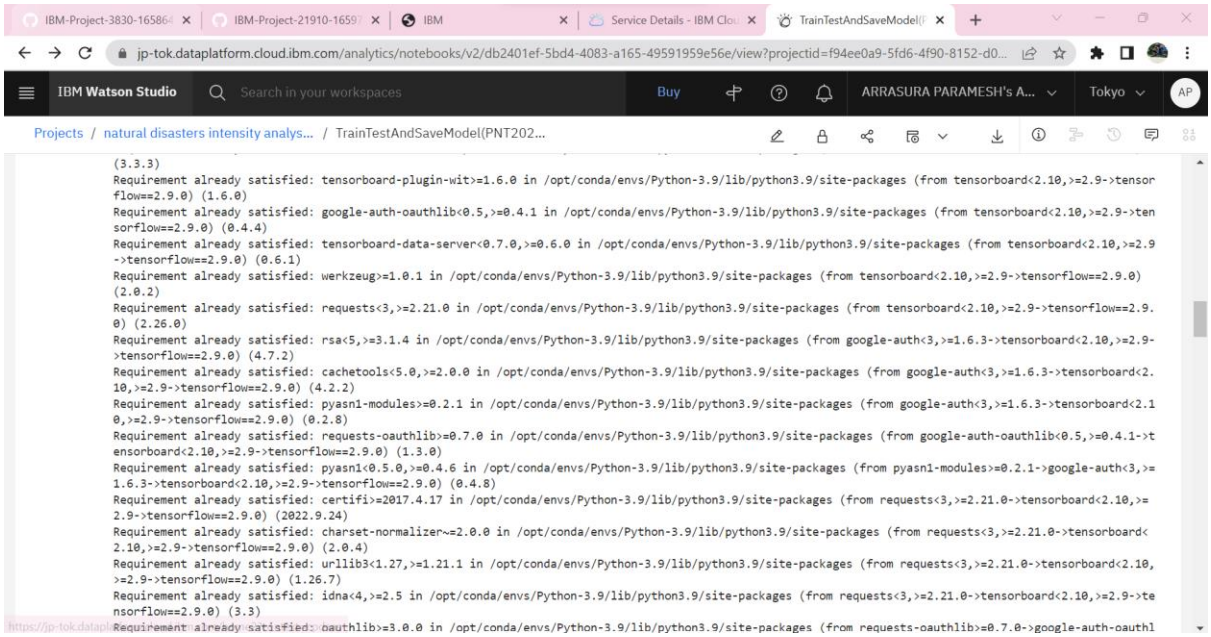
TRAINING THE MODEL ON IBM WATSON STUDIO:

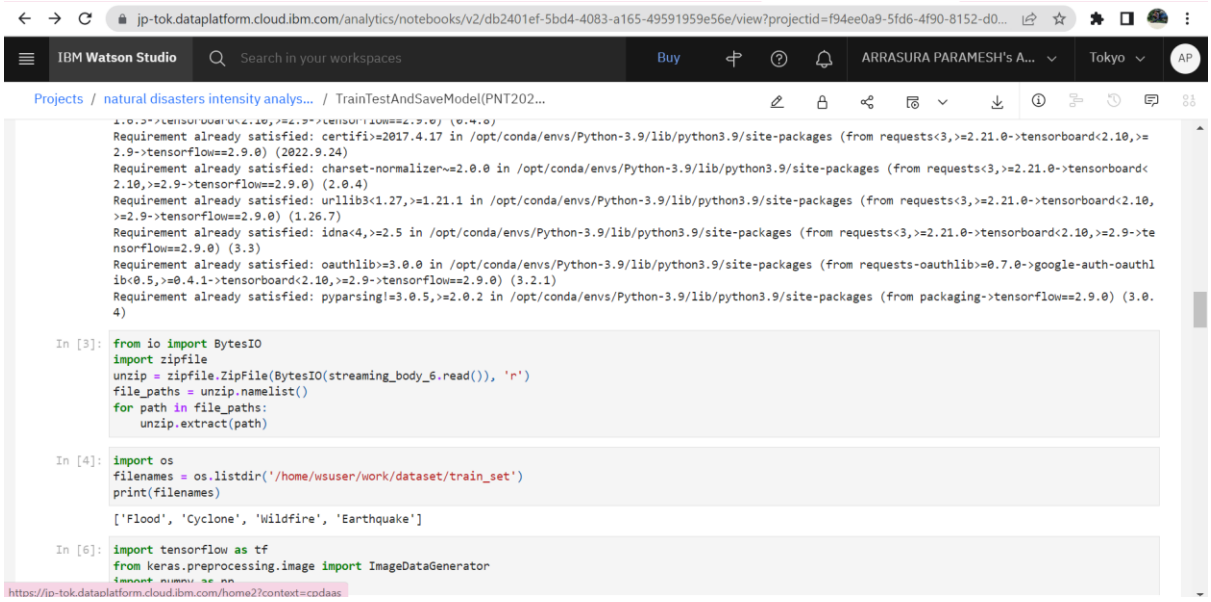


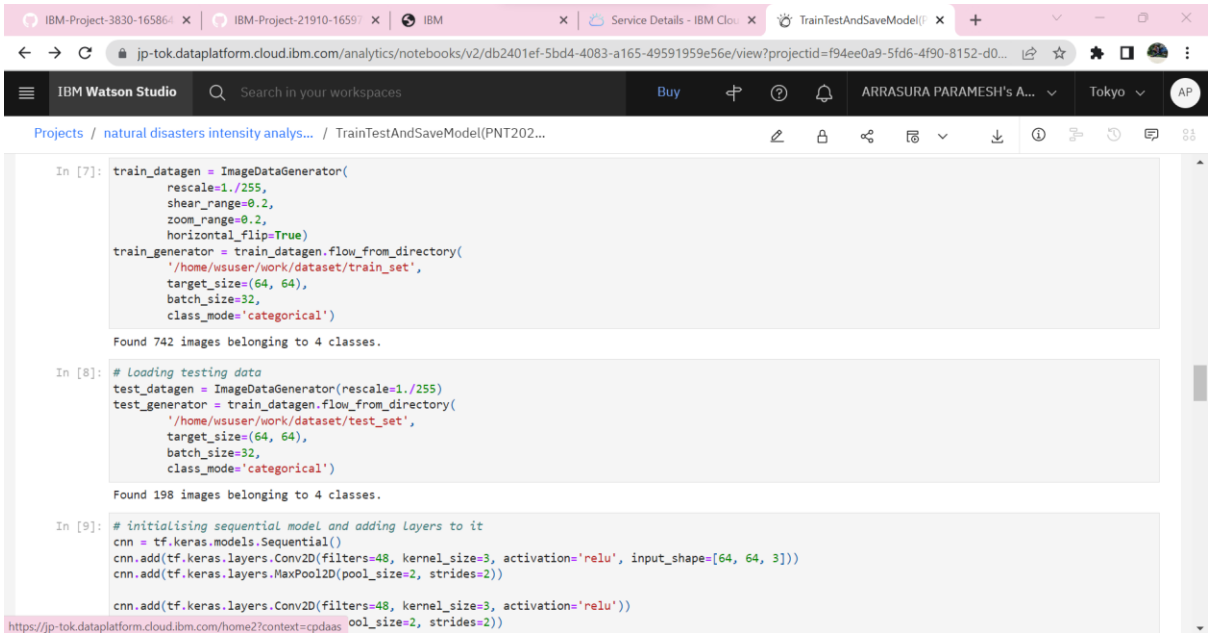


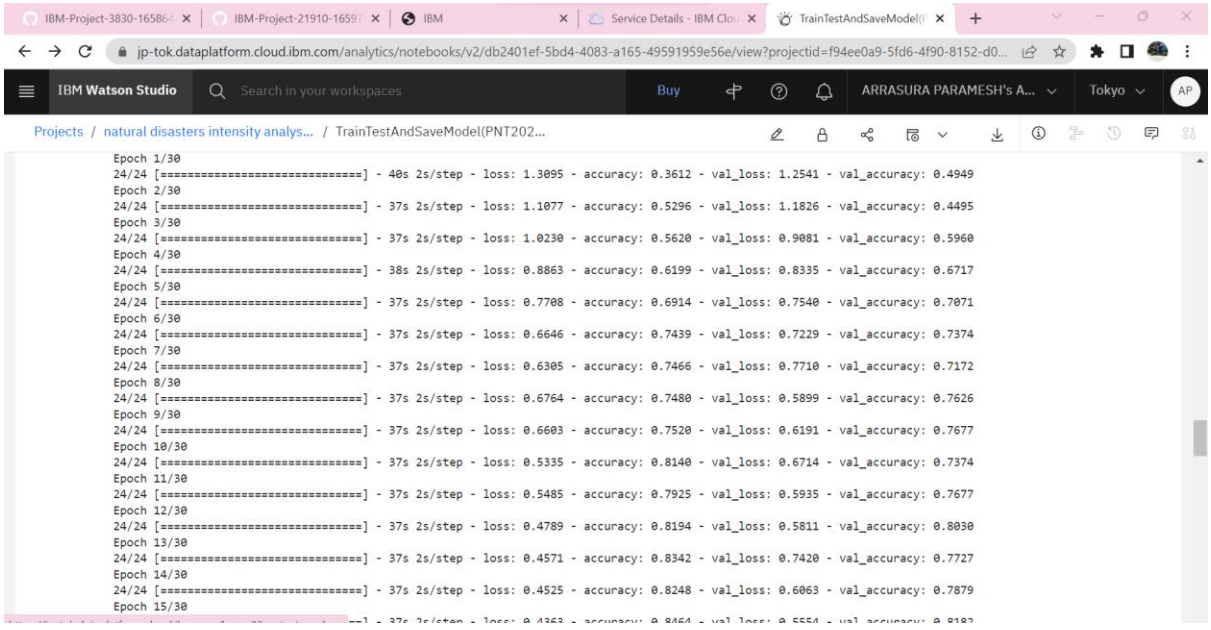

IBM-Project-3830-165864
IBM-Project-21910-16597
IBM
Service Details - IBM Clou
TrainTestAndSaveModel(P
jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/db2401ef-5bd4-4083-a165-49591959e56e/view?projectid=f94ee0a9-5fd6-4f90-8152-d0...
IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
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='c8BSddnc0SDJ3fks4ax4MmGMFDJPWFmCZ0gn2MCm_ban',
ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
config=Config(signature_version='oauth'),
endpoint_url='https://s3.private.ap.cloud-object-storage.appdomain.cloud')
bucket = 'naturaldisastersintensityanalysis-donotdelete-pr-etzuose0dbtvao'
object_key = 'dataset.zip'
streaming_body_7 = 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/
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
https://jp-tok.dataplatform.cloud.ibm.com/home2?context=cpdaas

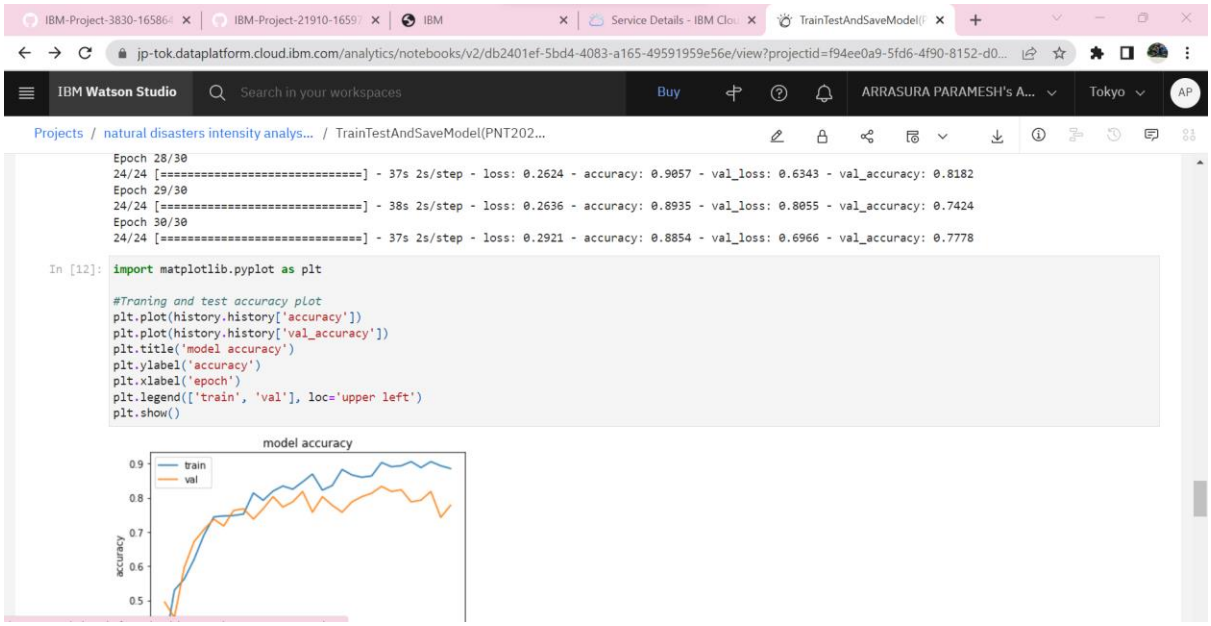

IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
ibm_api_key_id='c8BSddnc0SDJ3fks4ax4MmGMFDJPWFmCZ0gn2MCm_ban',
ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
config=Config(signature_version='oauth'),
endpoint_url='https://s3.private.ap.cloud-object-storage.appdomain.cloud')
bucket = 'naturaldisastersintensityanalysis-donotdelete-pr-etzuose0dbtvao'
object_key = 'dataset.zip'
streaming_body_6 = 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/
In [2]:
!pip install keras==2.9.0
!pip install tensorflow==2.9.0
Requirement already satisfied: keras==2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.9.0)
Requirement already satisfied: tensorflow==2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.9.0)
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.9.0) (1.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.9.0) (4.1.1)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.9.0) (1.6.3)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.9.0) (1.12.1)
Requirement already satisfied: absl-py>=1.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.9.0) (1.3.0)
https://jp-tok.dataplatform.cloud.ibm.com/home2?context=cpdaas

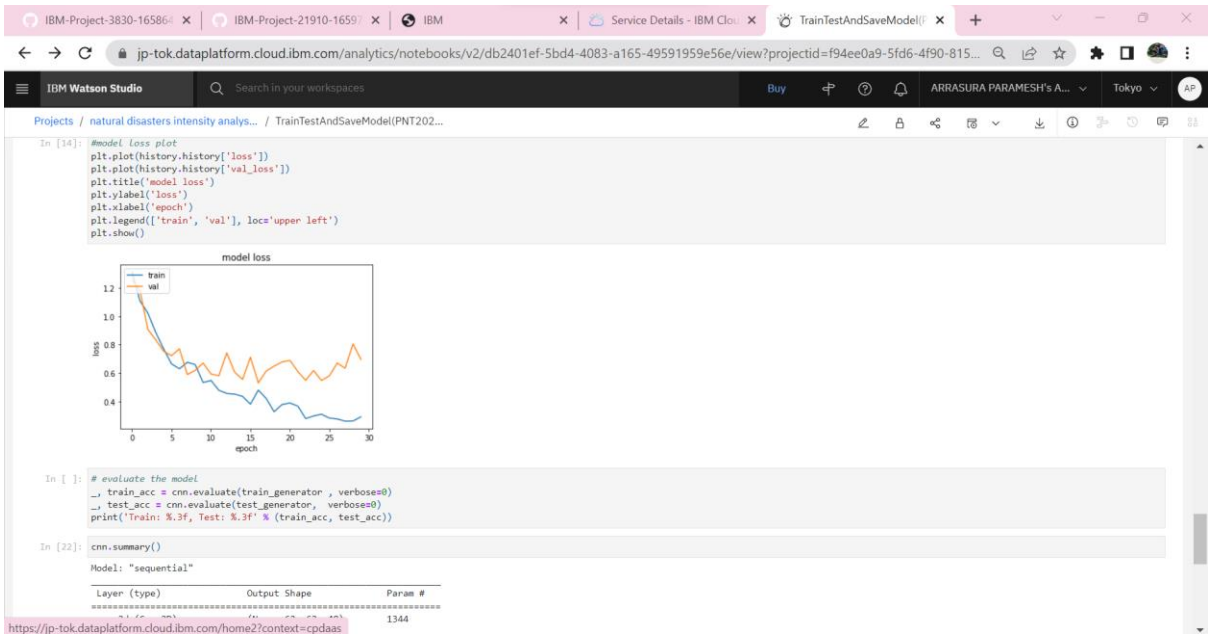

IBM-Project-3830-165864
IBM-Project-21910-1659
IBM
Service Details - IBM Clou
TrainTestAndSaveModel(P
jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/db2401ef-5bd4-4083-a165-49591959e56e/view?projectid=f94ee0a9-5fd6-4f90-8152-d0...
IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
(3.3.3)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.10,>=2.9->tensorflow==2.9.0) (1.6.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.10,>=2.9->tensorflow==2.9.0) (0.4.4)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.10,>=2.9->tensorflow==2.9.0) (0.6.1)
Requirement already satisfied: werkzeug>=1.0.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.10,>=2.9->tensorflow==2.9.0) (2.0.2)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.10,>=2.9->tensorflow==2.9.0) (2.26.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (4.7.2)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (4.2.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (1.3.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (2022.9.24)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (3.3)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthl

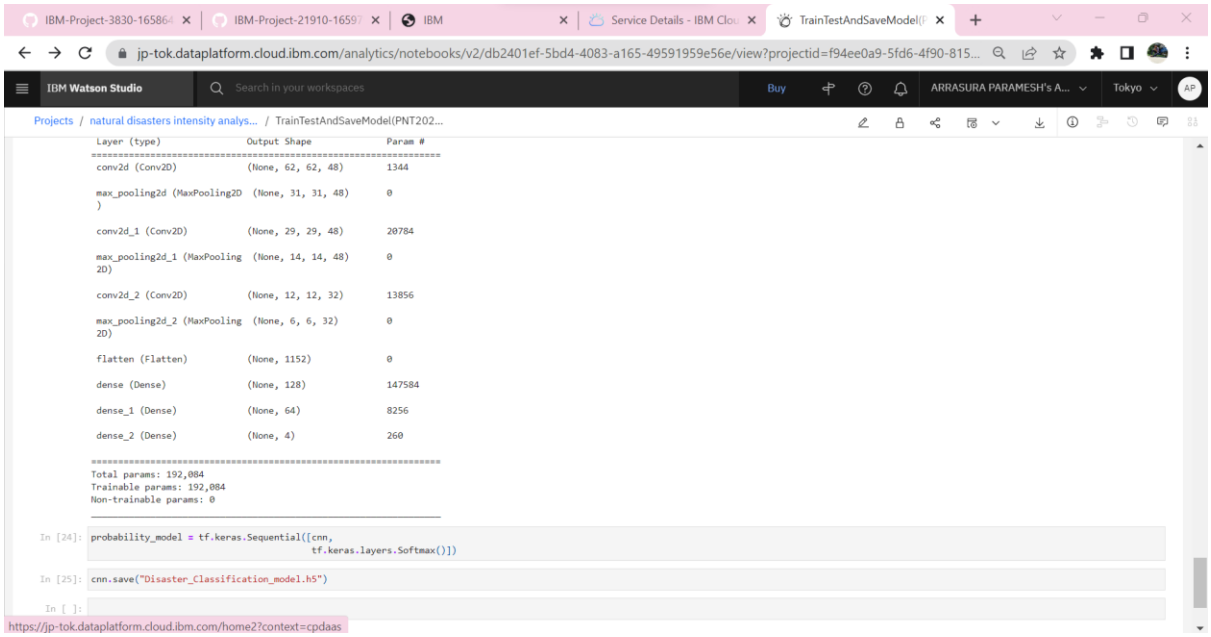

jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/db2401ef-5bd4-4083-a165-49591959e56e/view?projectid=f94ee0a9-5fd6-4f90-8152-d0...
IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (2022.9.24)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (3.3)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow==2.9.0) (3.2.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->tensorflow==2.9.0) (3.0.4)
In [3]:
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_6.read()), 'r')
file_paths = unzip.namelist()
for path in file_paths:
unzip.extract(path)
In [4]:
import os
filenames = os.listdir('/home/wsuser/work/dataset/train_set')
print(filenames)
['Flood', 'Cyclone', 'Wildfire', 'Earthquake']
In [6]:
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
https://jp-tok.dataplatform.cloud.ibm.com/home2?context=cpdaas


IBM-Project-3830-165864
IBM-Project-21910-1659
IBM
Service Details - IBM Clou
TrainTestAndSaveModel(P
jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/db2401ef-5bd4-4083-a165-49591959e56e/view?projectid=f94ee0a9-5fd6-4f90-8152-d0...
IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
In [7]:
train_datagen = ImageDataGenerator(
rescale=1./255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)
train_generator = train_datagen.flow_from_directory(
'/home/wsuser/work/dataset/train_set',
target_size=(64, 64),
batch_size=32,
class_mode='categorical')
Found 742 images belonging to 4 classes.
In [8]:
# Loading testing data
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = train_datagen.flow_from_directory(
'/home/wsuser/work/dataset/test_set',
target_size=(64, 64),
batch_size=32,
class_mode='categorical')
Found 198 images belonging to 4 classes.
In [9]:
# initialising sequential model and adding layers to it
cnn = tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=48, kernel_size=3, activation='relu', input_shape=[64, 64, 3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=48, kernel_size=3, activation='relu'))
ool_size=2, strides=2))
https://jp-tok.dataplatform.cloud.ibm.com/home2?context=cpdaas


IBM-Project-3830-165864
IBM-Project-21910-1659
IBM
Service Details - IBM Clou
TrainTestAndSaveModel(P
jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/db2401ef-5bd4-4083-a165-49591959e56e/view?projectid=f94ee0a9-5fd6-4f90-8152-d0...
IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
Epoch 1/30
24/24 [==============================] - 40s 2s/step - loss: 1.3095 - accuracy: 0.3612 - val_loss: 1.2541 - val_accuracy: 0.4949
Epoch 2/30
24/24 [==============================] - 37s 2s/step - loss: 1.1077 - accuracy: 0.5296 - val_loss: 1.1826 - val_accuracy: 0.4495
Epoch 3/30
24/24 [==============================] - 37s 2s/step - loss: 1.0230 - accuracy: 0.5620 - val_loss: 0.9081 - val_accuracy: 0.5960
Epoch 4/30
24/24 [==============================] - 38s 2s/step - loss: 0.8863 - accuracy: 0.6199 - val_loss: 0.8335 - val_accuracy: 0.6717
Epoch 5/30
24/24 [==============================] - 37s 2s/step - loss: 0.7708 - accuracy: 0.6914 - val_loss: 0.7540 - val_accuracy: 0.7071
Epoch 6/30
24/24 [==============================] - 37s 2s/step - loss: 0.6646 - accuracy: 0.7439 - val_loss: 0.7229 - val_accuracy: 0.7374
Epoch 7/30
24/24 [==============================] - 37s 2s/step - loss: 0.6305 - accuracy: 0.7466 - val_loss: 0.7710 - val_accuracy: 0.7172
Epoch 8/30
24/24 [==============================] - 37s 2s/step - loss: 0.6764 - accuracy: 0.7480 - val_loss: 0.5899 - val_accuracy: 0.7626
Epoch 9/30
24/24 [==============================] - 37s 2s/step - loss: 0.6603 - accuracy: 0.7520 - val_loss: 0.6191 - val_accuracy: 0.7677
Epoch 10/30
24/24 [==============================] - 37s 2s/step - loss: 0.5335 - accuracy: 0.8140 - val_loss: 0.6714 - val_accuracy: 0.7374
Epoch 11/30
24/24 [==============================] - 37s 2s/step - loss: 0.5485 - accuracy: 0.7925 - val_loss: 0.5935 - val_accuracy: 0.7677
Epoch 12/30
24/24 [==============================] - 37s 2s/step - loss: 0.4789 - accuracy: 0.8194 - val_loss: 0.5811 - val_accuracy: 0.8030
Epoch 13/30
24/24 [==============================] - 37s 2s/step - loss: 0.4571 - accuracy: 0.8342 - val_loss: 0.7420 - val_accuracy: 0.7727
Epoch 14/30
24/24 [==============================] - 37s 2s/step - loss: 0.4525 - accuracy: 0.8248 - val_loss: 0.6063 - val_accuracy: 0.7879
Epoch 15/30


IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
Epoch 28/30
24/24 [==============================] - 37s 2s/step - loss: 0.2624 - accuracy: 0.9057 - val_loss: 0.6343 - val_accuracy: 0.8182
Epoch 29/30
24/24 [==============================] - 38s 2s/step - loss: 0.2636 - accuracy: 0.8935 - val_loss: 0.8055 - val_accuracy: 0.7424
Epoch 30/30
24/24 [==============================] - 37s 2s/step - loss: 0.2921 - accuracy: 0.8854 - val_loss: 0.6966 - val_accuracy: 0.7778
In [12]:
import matplotlib.pyplot as plt
#Traning and test accuracy plot
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
model accuracy
train
val
accuracy
epoch


IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
In [14]:
#model loss plot
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
model loss
train
val
loss
epoch
In [ ]:
evaluate the model
_, train_acc = cnn.evaluate(train_generator , verbose=0)
_, test_acc = cnn.evaluate(test_generator, verbose=0)
print('Train: %.3f, Test: %.3f' % (train_acc, test_acc))
In [22]:
cnn.summary()
Model: "sequential"
Layer (type)
Output Shape
Param #
1344
https://jp-tok.dataplatform.cloud.ibm.com/home2?context=cpdaas


IBM-Project-3830-165864
IBM-Project-21910-16597
IBM
Service Details - IBM Clou
TrainTestAndSaveModel(P
jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/db2401ef-5bd4-4083-a165-49591959e56e/view?projectid=f94ee0a9-5fd6-4f90-815...
IBM Watson Studio
Search in your workspaces
Buy
ARRASURA PARAMESH's A...
Tokyo
AP
Projects / natural disasters intensity analys... / TrainTestAndSaveModel(PNT202...
Layer (type) Output Shape Param #
conv2d (Conv2D) (None, 62, 62, 48) 1344
max_pooling2d (MaxPooling2D (None, 31, 31, 48) 0
)
conv2d_1 (Conv2D) (None, 29, 29, 48) 20784
max_pooling2d_1 (MaxPooling (None, 14, 14, 48) 0
2D)
conv2d_2 (Conv2D) (None, 12, 12, 32) 13856
max_pooling2d_2 (MaxPooling (None, 6, 6, 32) 0
2D)
flatten (Flatten) (None, 1152) 0
dense (Dense) (None, 128) 147584
dense_1 (Dense) (None, 64) 8256
dense_2 (Dense) (None, 4) 260
Total params: 192,084
Trainable params: 192,084
Non-trainable params: 0
In [24]: probability_model = tf.keras.Sequential([cnn,
tf.keras.layers.Softmax()])
In [25]: cnn.save("Disaster_Classification_model.h5")
In [ ]:
https://jp-tok.dataplatform.cloud.ibm.com/home2?context=cpdaas