Train The Model On IBM

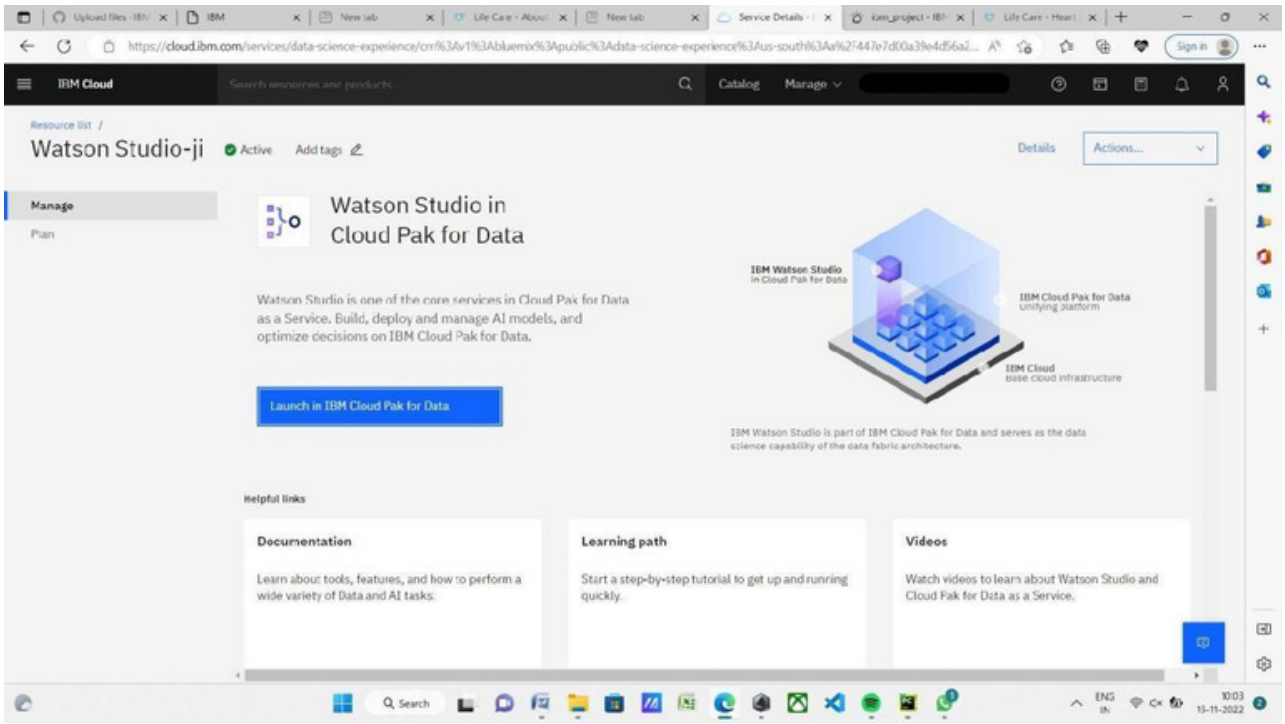
Train The Model on IBM Watson

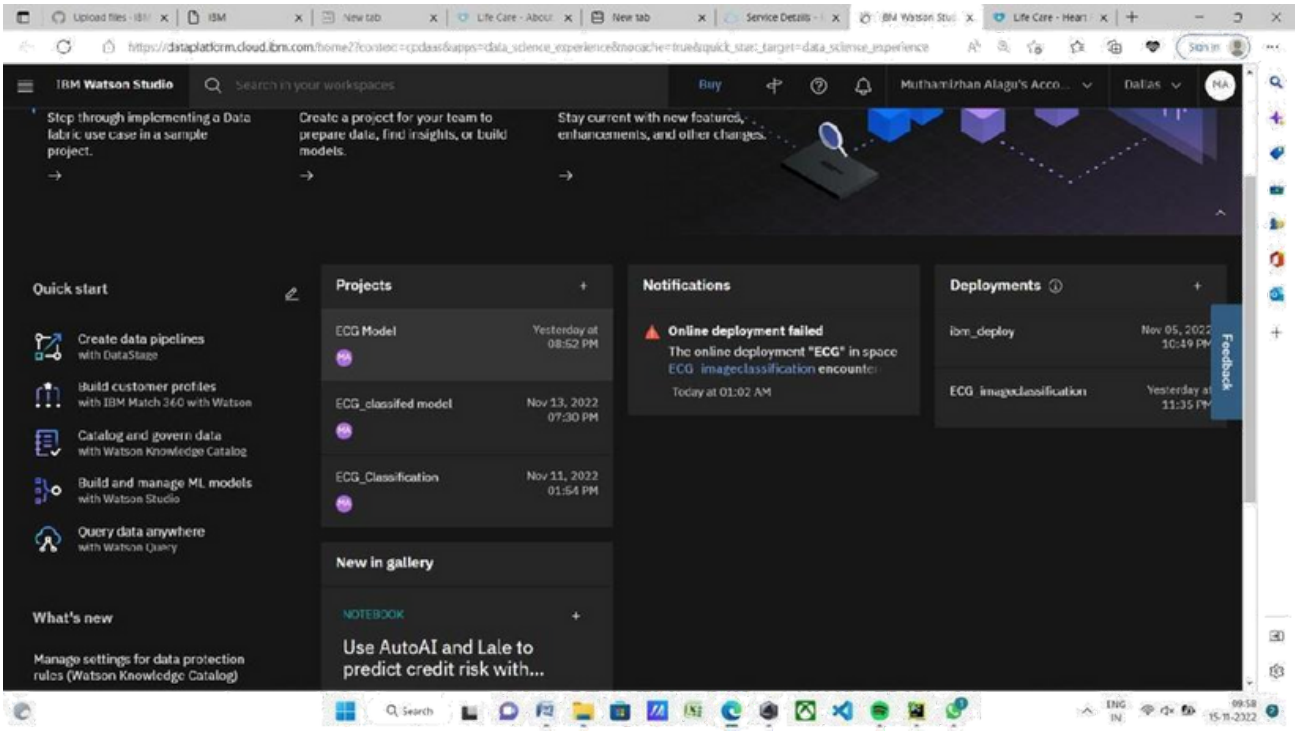
Date	19 Nov 2022
TeamID	PNT2022TMID12347
ProjectName	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

TASK:

Train The Model on IBM Watson:

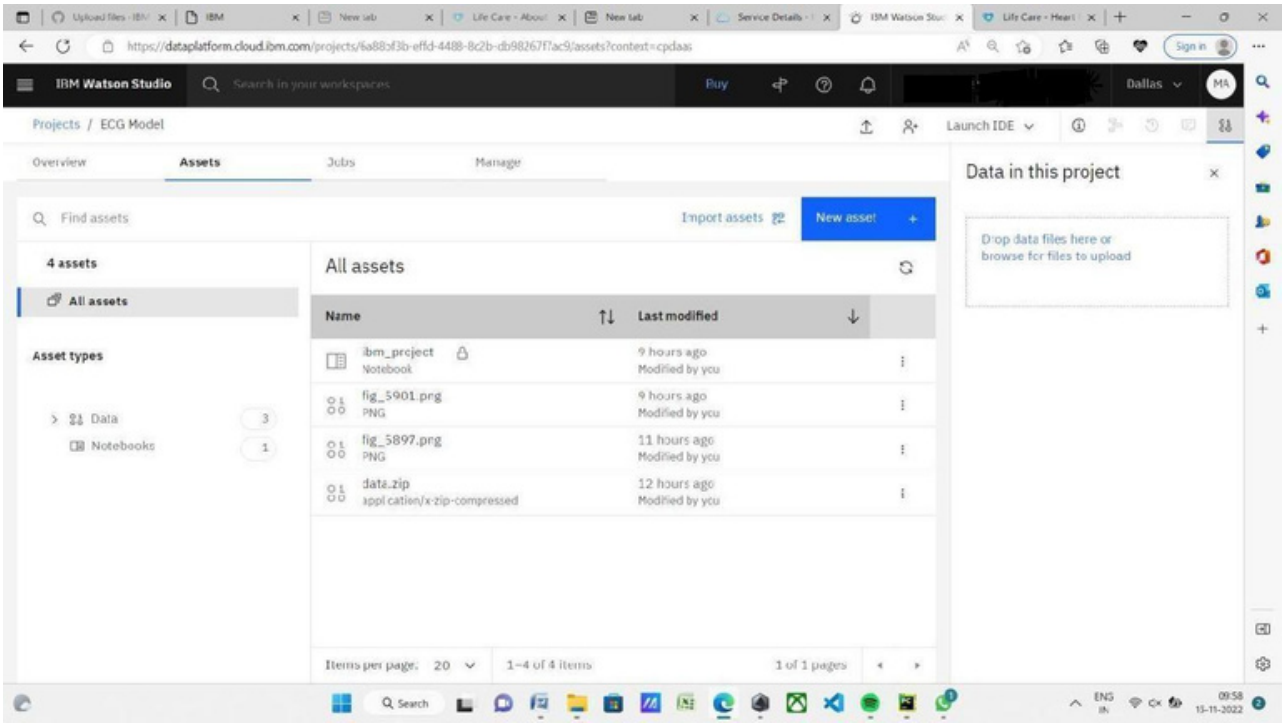
GO ON WATSON SERVICES(SCREEN SHOT):

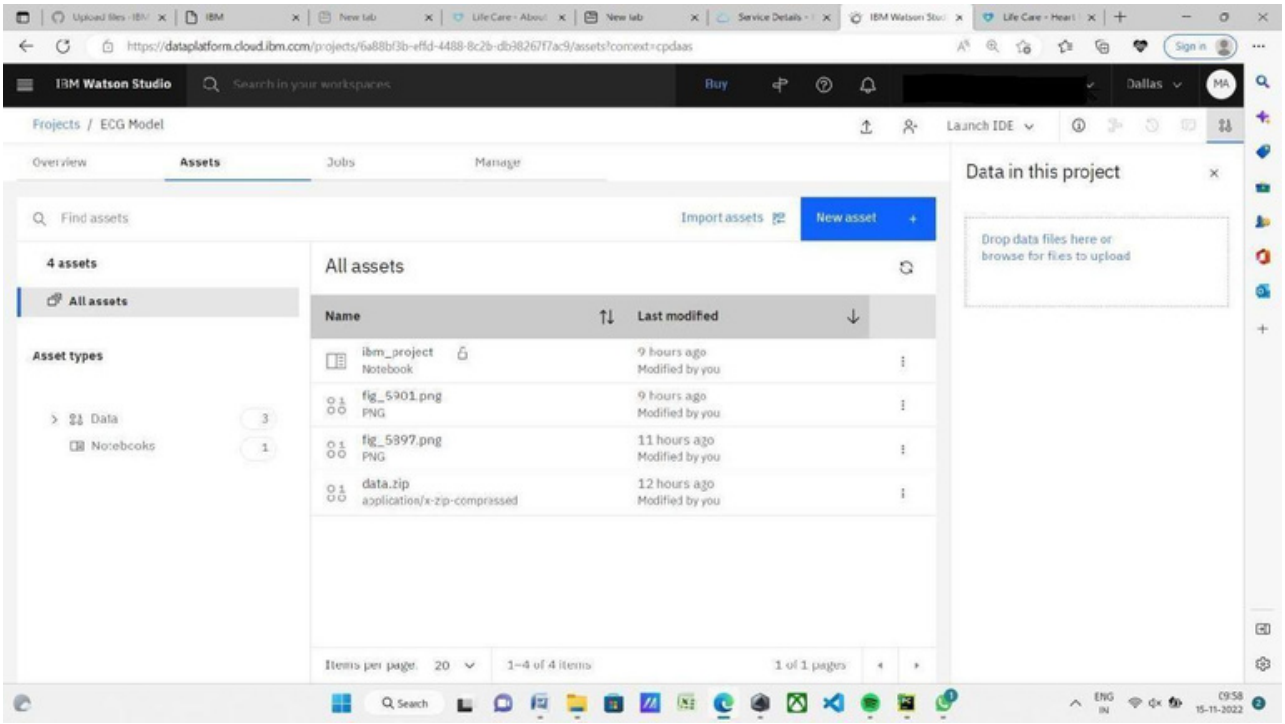




GO ON NEW PROJECT (SCREEN SHOT):

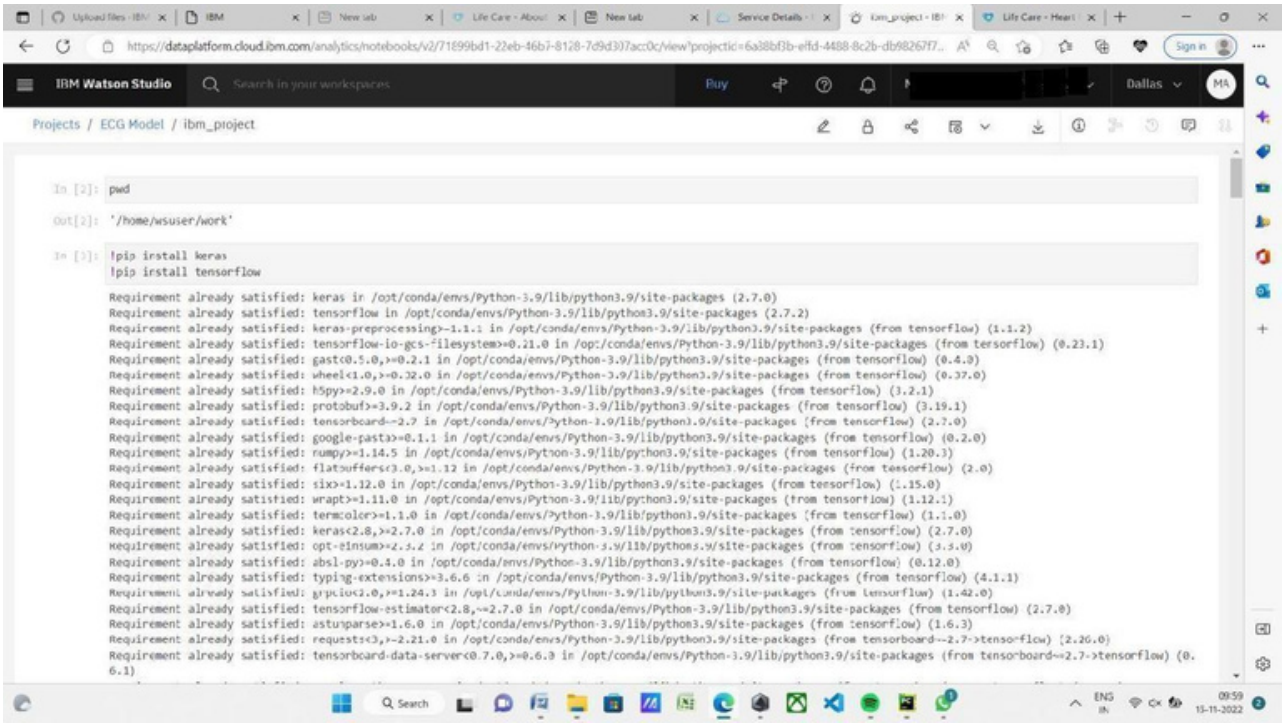
GO ON ASSEST(SCREEN SHOT):

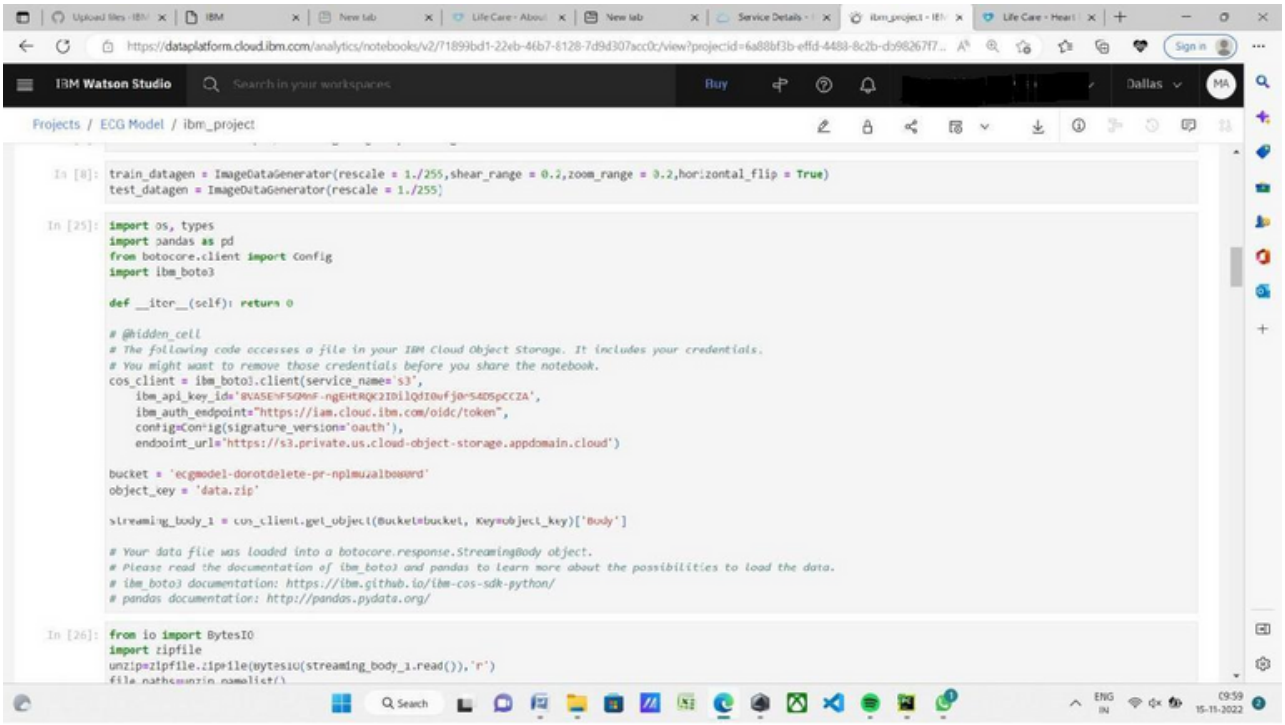


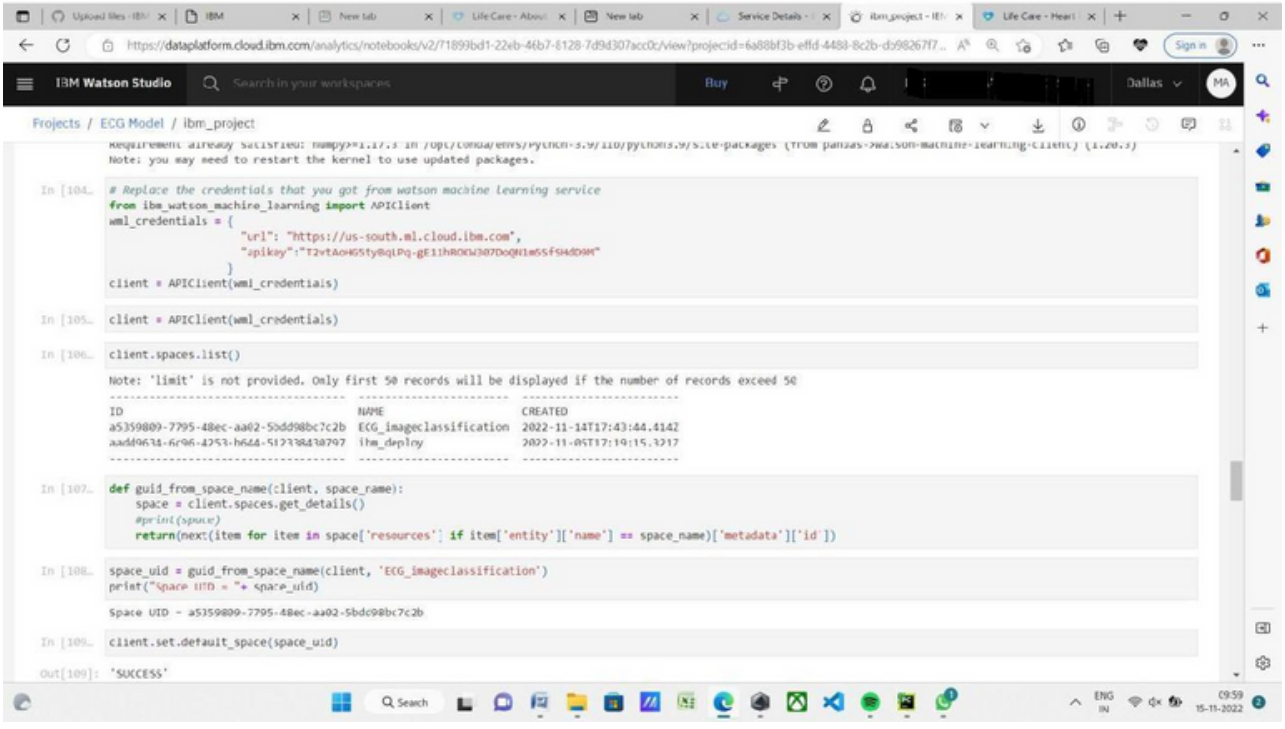


GO ON IBM_PROJECT(SCREEN SHOT):

JUPYTER NOTEBOOK(SCREEN SHOT):


IBM Watson Studio
Projects / ECG Model / ibm_project
In [2]: pwd
Out[2]: '/home/wsuser/work'
In [3]: !pip install keras
!pip install tensorflow
Requirement already satisfied: keras in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.0)
Requirement already satisfied: tensorflow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.2)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.23.1)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: wheel<1.0,>=0.32.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.37.0)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.2.1)
Requirement already satisfied: protobuf>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: tensorboard~=2.7 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.20.3)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.0)
Requirement already satisfied: six>=1.12.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.15.0)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: keras<2.8,>=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: absl-py>=0.4.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.12.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (4.1.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (2.26.0)
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~=2.7->tensorflow) (0.6.1)


IBM Watson Studio
Projects / ECG Model / ibm_project
In [8]:
train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
In [25]:
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='8VA5EhF5GMmF-ngEHtRQK2IDilQdI0ufjb-54D5pCCZA',
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 = 'ecgmodel-donotdelete-pr-nplmuzalbowwrd'
object_key = 'data.zip'
streaming_body_1 = 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 [26]:
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')


IBM Watson Studio
Projects / ECG Model / ibm_project
Note: you may need to restart the kernel to use updated packages.
In [104...
# Replace the credentials that you got from watson machine learning service
from ibm_watson_machine_learning import APIClient
wml_credentials = {
"url": "https://us-south.ml.cloud.ibm.com",
"apikey":"T2vtAo4G5tyBqLPq-gE1ihROKW307DoQN1m5SfS4dDWM"
}
client = APIClient(wml_credentials)
In [105...
client = APIClient(wml_credentials)
In [106...
client.spaces.list()
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
ID NAME CREATED
a5359809-7795-48ec-aa02-5bdd98bc7c2b ECG_imageclassification 2022-11-14T17:43:44.414Z
aadd9634-6c96-4253-b664-513788438797 ibm_deploy 2022-11-05T17:19:15.321Z
In [107...
def guid_from_space_name(client, space_name):
space = client.spaces.get_details()
#print(space)
return(next(item for item in space['resources'] if item['entity']['name'] == space_name)['metadata']['id'])
In [108...
space_uid = guid_from_space_name(client, 'ECG_imageclassification')
print("Space UID = "+ space_uid)
Space UID = a5359809-7795-48ec-aa02-5bdc98bc7c2b
In [109...
client.set.default_space(space_uid)
Out[109]:
'SUCCESS'