Flow Chart & Results by training model in IBM WATSON STUDIO:

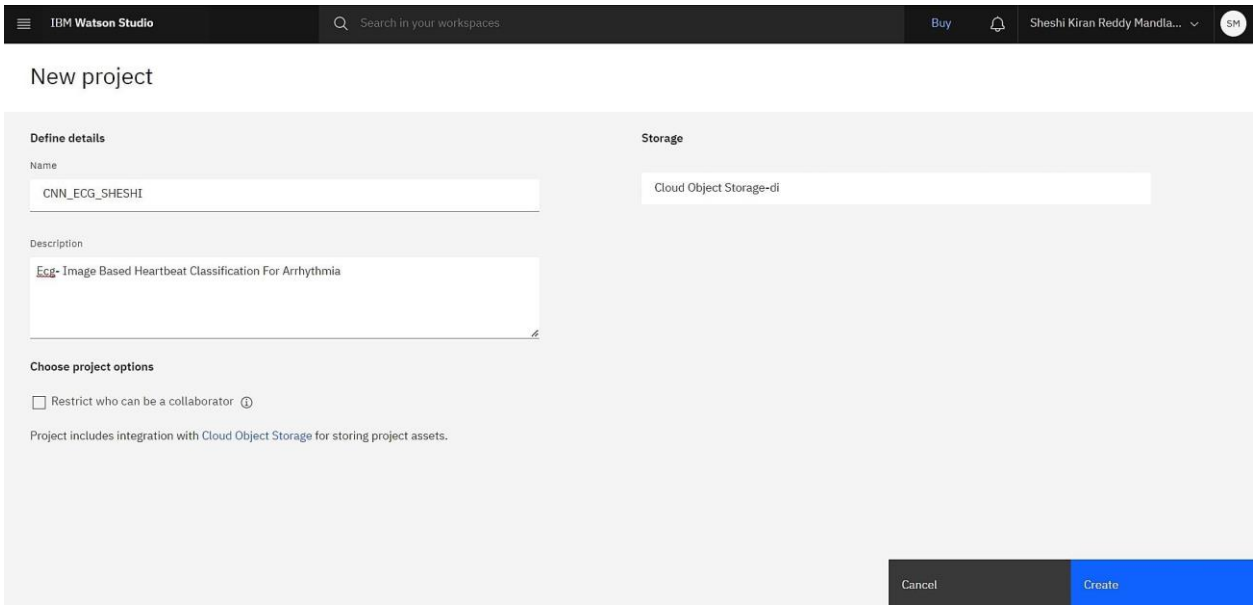
a. Creating IBM cloud account:

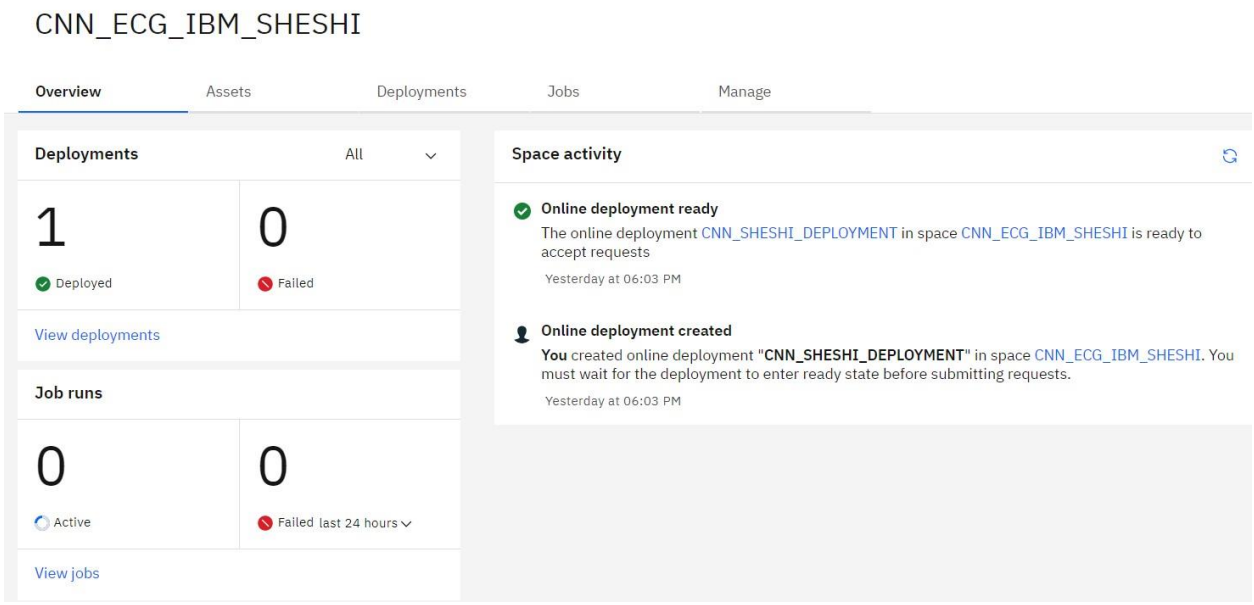
We have to create an IBM Cloud Account and should log in.

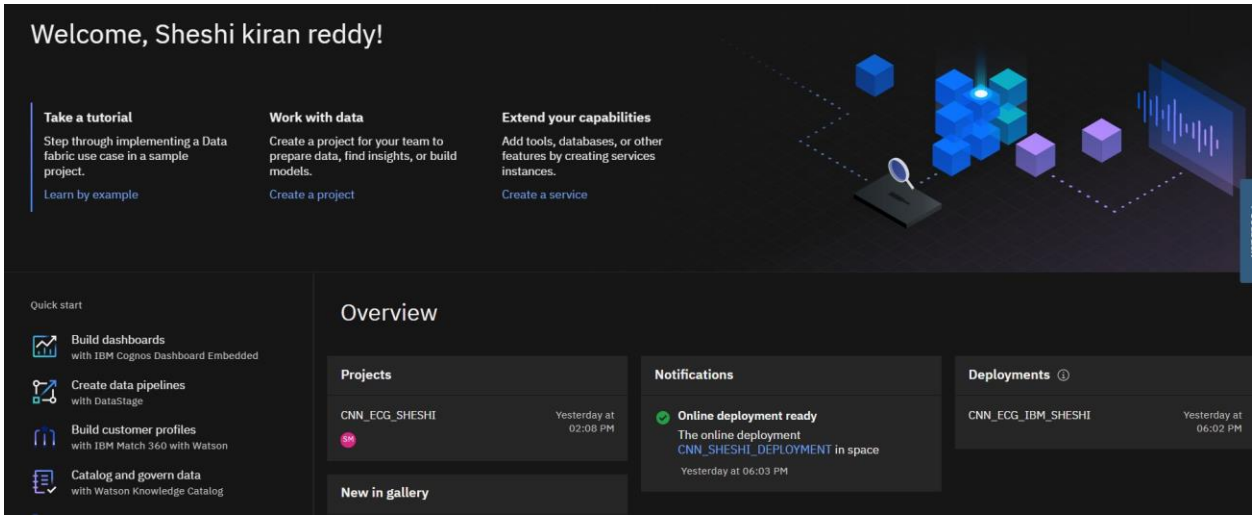
b. Creting Watson Studio Service & Machine Learning Service:

Machine Learning-7n	Default	Dallas	Machine Learning	Active
Watson Studio-yy	Default	Dallas	Watson Studio	Active

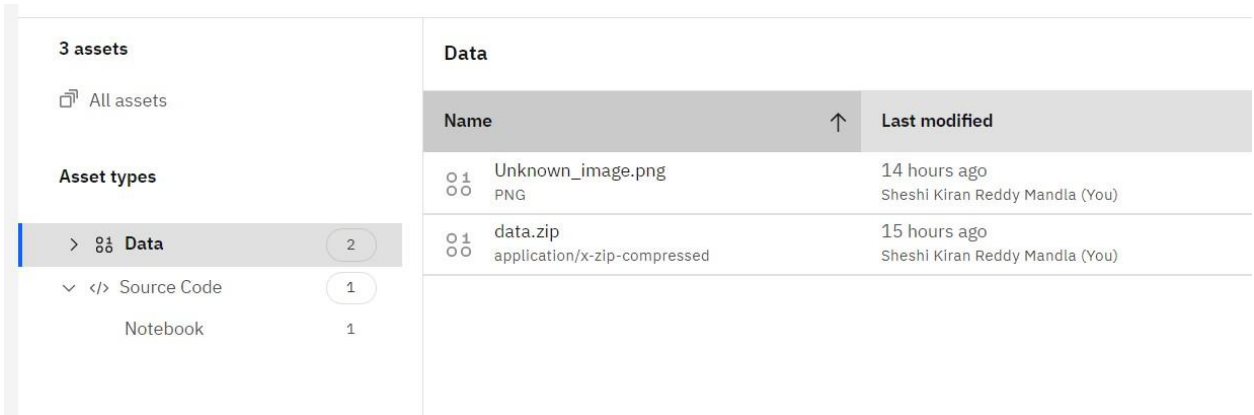
c. Create a Project & Deployment space in the watson studio:

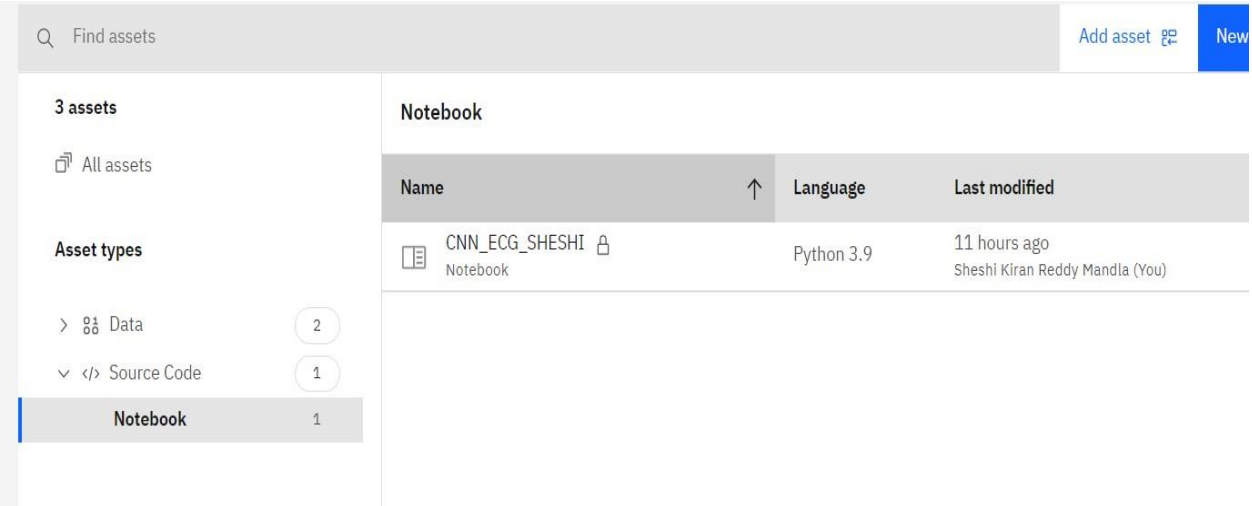






d. Upload The dataset and create a jupyter source file in the created project:





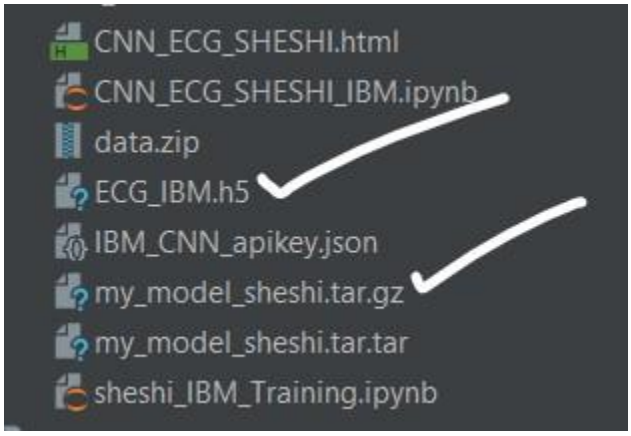
e. Apply CNN algorithm and save the model and deploy it using API key generated:

```
In [77]: model.save('ECG_IBM.h5')

In [109]: !tar -zcvf ECG-arrhythmia-classification-model_new.tgz ECG_IBM.h5
ECG_IBM.h5

In [110]: ls -1
data/
ECG-arrhythmia-classification-model_new.tgz
ECG-classification.tgz
ECG_IBM.h5
```

```
In [112]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"EfnNlIAqu-_QXB0QsQqQQlnwkes_B9ssggk8ipjZQH67"
}
client=APIClient(wml_credentials)

In [121]: 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
fc75767f-659b-4dc3-a238-cbb53d201cf2  CNN_ECG_IBM_SHESHI  2022-07-05T10:52:44.075Z
------------------------------------  ------------------  -------------------------

In [122]: space_uid="fc75767f-659b-4dc3-a238-cbb53d201cf2"

In [123]: client.set.default_space(space_uid)

Out[123]: 'SUCCESS'

In [124]: client.set.default_space(space_uid)

Out[124]: 'SUCCESS'

In [126]: software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid

Out[126]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

In [143]: model_details = client.repository.store_model(model='ECG-arrhythmia-classification-model_new.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"CNN_SHESHI",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})
model_id=client.repository.get_model_uid(model_details)

This method is deprecated, please use get_model_id()

In [144]: model_id

Out[144]: '70742fe8-7ac8-4855-994c-b572931ef787'

In [147]: client.repository.download(model_id,'my_model.tar1.gz')

Successfully saved model content to file: 'my_model.tar1.gz'

Out[147]: '/home/wsuser/work/my_model.tar1.gz'
```

f. For downloading the model we have to run the last part of the above code in the local jupyter notebook:

```
In [ ]: 1 client.repository.download('70742fe8-7ac8-4855-994c-b572931ef787','my_model_sheshi.tar.gz')
```



g. **Now we will extract the .h5 model file and will do the app deployment using flask as done in the previous training:**

```
import os
import numpy as np  # used for numerical analysis
from flask import Flask, request, render_template
# Flask-It is our framework which we are going to use to run/serve our appl
# request-for accessing file which was uploaded by the user on our applicat
# render_template- used for rendering the html pages
from tensorflow.keras.models import load_model  # to load our trained model
from tensorflow.keras.preprocessing import image

app = Flask(__name__)  # our flask app
model = load_model('ECG_IBM.h5')  # loading the model


@app.route("/")  # default route
def about():
    return render_template("home.html")  # rendering html page
```

Hence we trained the model using IBM Watson.

Advantages & Disadvantages:

6.1 Advantages:

- The proposed model predicts Arrhythmia in images with a high accuracy rate of nearly 96%
- The early detection of Arrhythmia gives better understanding of disease causes, initiates therapeutic interventions and enables developing appropriate treatments.

6.2 Disadvantages:

- Not useful for identifying the different stages of Arrhythmia disease.
- Not useful in monitoring motor symptoms

Applications :

- It is useful for identifying the arrhythmia disease at an early stage. ● It is useful in detecting cardiovascular disorders

.

Conclusion:

- Cardiovascular disease is a major health problem in today's world. The early diagnosis of cardiac arrhythmia highly relies on the ECG.
- Unfortunately, the expert level of medical resources is rare, visually identify the ECG signal is challenging and time-consuming.
- The advantages of the proposed CNN network have been put to evidence.
- It is endowed with an ability to effectively process the non-filtered dataset with its potential anti-noise features. Besides that, ten-fold cross-validation is implemented in this work to further demonstrate the robustness of the network.

Future Scope:

For future work, it would be interesting to explore the use of optimization techniques to find a feasible design and solution. The limitation of our study is that we have yet to apply any optimization techniques to optimize the model parameters and we believe that with the implementation of the optimization, it will be able to further elevate the performance of the proposed solution to the next level.

THE END