

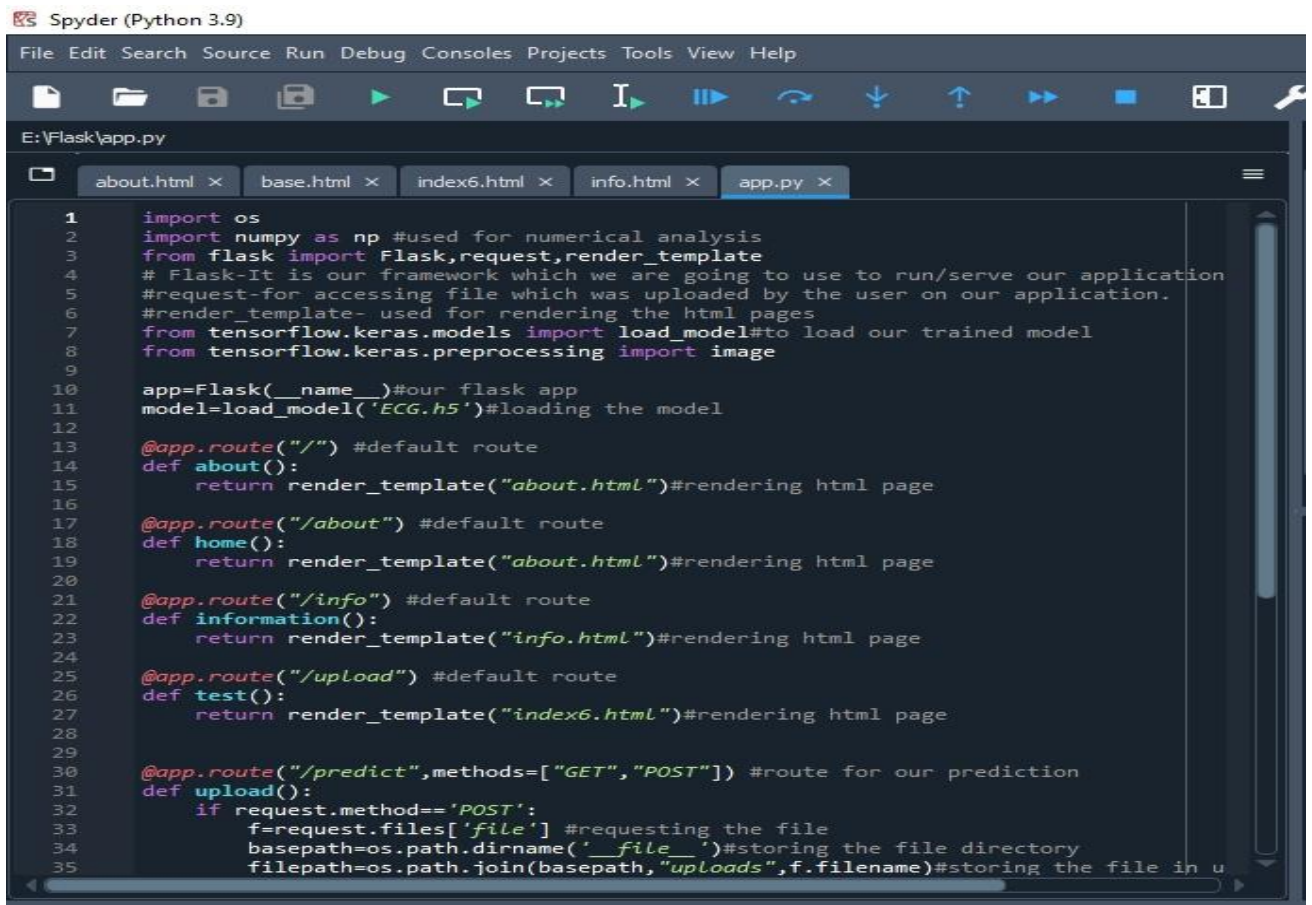
Sprint-3

Classification of Arrhythmia by Using Deep Learning With 2-DECG Spectral Image Representation

Build python code:

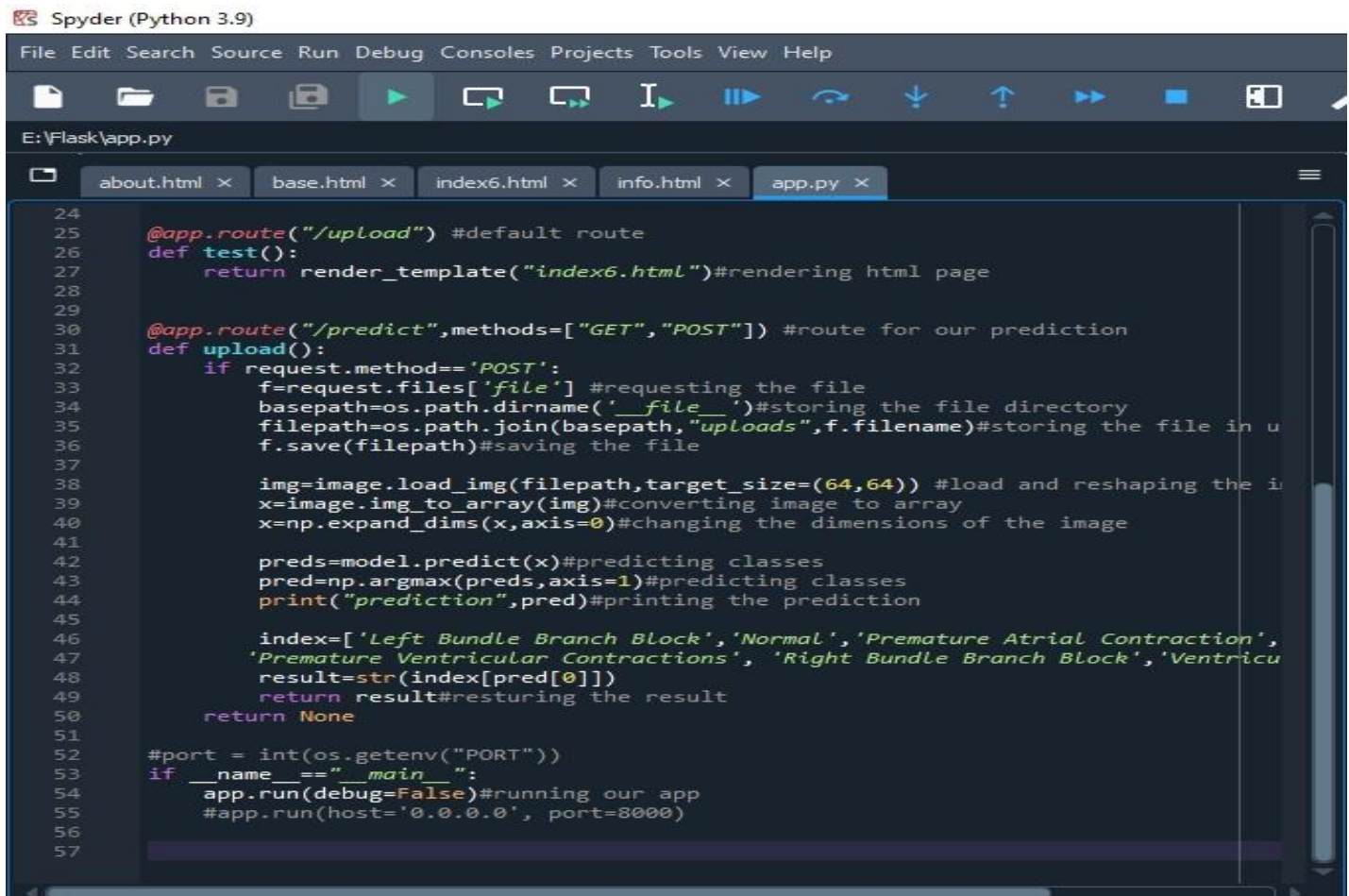
- Let us build the flask file 'app.py' which is a web framework written in python for server-side scripting. Let's see step by step procedure for building the backend application.
- The app starts running when the "__name__" constructor is called in main.
- render_template is used to return HTML file.
- "GET" method is used to take input from the user.
- "POST" method is used to display the output to the user.

Screenshot:



```
1 import os
2 import numpy as np #used for numerical analysis
3 from flask import Flask,request,render_template
4 # Flask-It is our framework which we are going to use to run/serve our application
5 #request-for accessing file which was uploaded by the user on our application.
6 #render_template- used for rendering the html pages
7 from tensorflow.keras.models import load_model#to load our trained model
8 from tensorflow.keras.preprocessing import image
9
10 app=Flask(__name__)#our flask app
11 model=load_model('ECG.h5')#loading the model
12
13 @app.route("/") #default route
14 def about():
15     return render_template("about.html")#rendering html page
16
17 @app.route("/about") #default route
18 def home():
19     return render_template("about.html")#rendering html page
20
21 @app.route("/info") #default route
22 def information():
23     return render_template("info.html")#rendering html page
24
25 @app.route("/upload") #default route
26 def test():
27     return render_template("index6.html")#rendering html page
28
29
30 @app.route("/predict",methods=["GET","POST"]) #route for our prediction
31 def upload():
32     if request.method=='POST':
33         f=request.files['file'] #requesting the file
34         basepath=os.path.dirname('__file__')#storing the file directory
35         filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in u
```

Routing to the HTML page:



```
24
25 @app.route("/upload") #default route
26 def test():
27     return render_template("index6.html")#rendering html page
28
29
30 @app.route("/predict",methods=["GET","POST"]) #route for our prediction
31 def upload():
32     if request.method=='POST':
33         f=request.files['file'] #requesting the file
34         basepath=os.path.dirname('__file__')#storing the file directory
35         filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in u
36         f.save(filepath)#saving the file
37
38         img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the i
39         x=image.img_to_array(img)#converting image to array
40         x=np.expand_dims(x,axis=0)#changing the dimensions of the image
41
42         preds=model.predict(x)#predicting classes
43         pred=np.argmax(preds,axis=1)#predicting classes
44         print("prediction",pred)#printing the prediction
45
46         index=['Left Bundle Branch Block','Normal','Premature Atrial Contraction',
47               'Premature Ventricular Contractions', 'Right Bundle Branch Block','Ventricu
48               result=str(index[pred[0]])
49         return result#resturing the result
50     return None
51
52 #port = int(os.getenv("PORT"))
53 if __name__=="__main__":
54     app.run(debug=False)#running our app
55     #app.run(host='0.0.0.0', port=8000)
56
57
```

Showcasing prediction on UI

When the image is uploaded, it predicts the category of uploaded the image is either 'Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricular Contractions', 'Right Bundle Branch Block', 'Ventricular Fibrillation'. If the image predicts value as 0, then it is displayed as “Left Bundle Branch”. Similarly, if the predicted value is 1, it displays “Normal” as output and so on.