Project Development Phase

Sprint4

Team ID	PNT2022TMID13501
Project Name	Project - Crude Oil Price Prediction

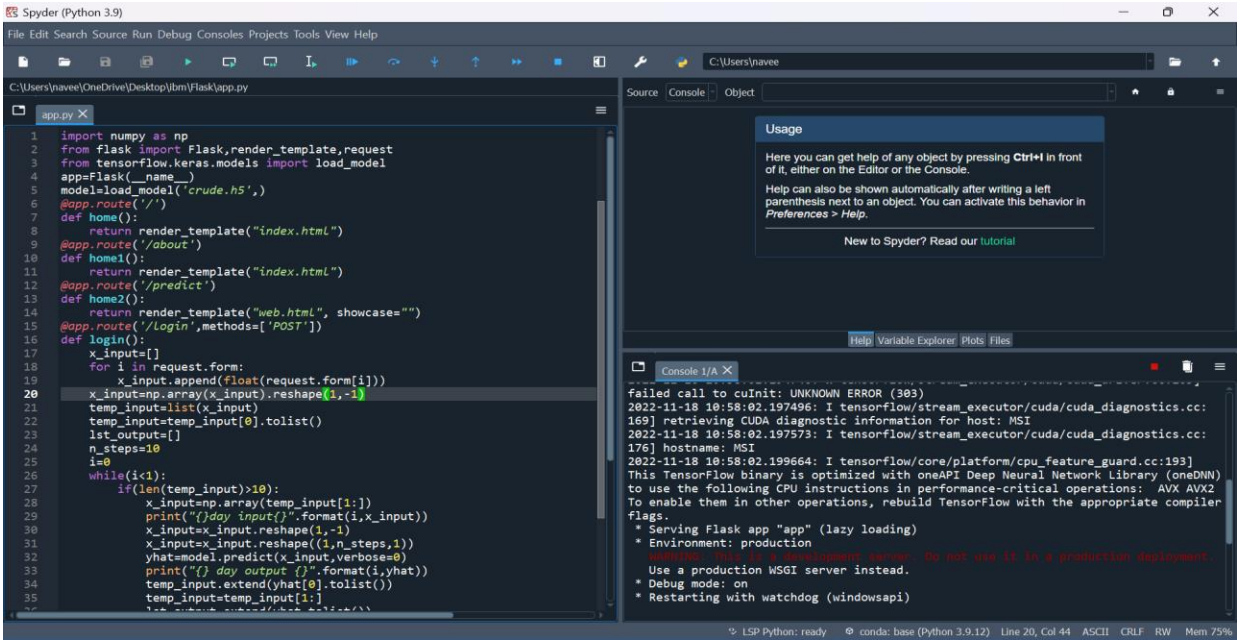
Python:

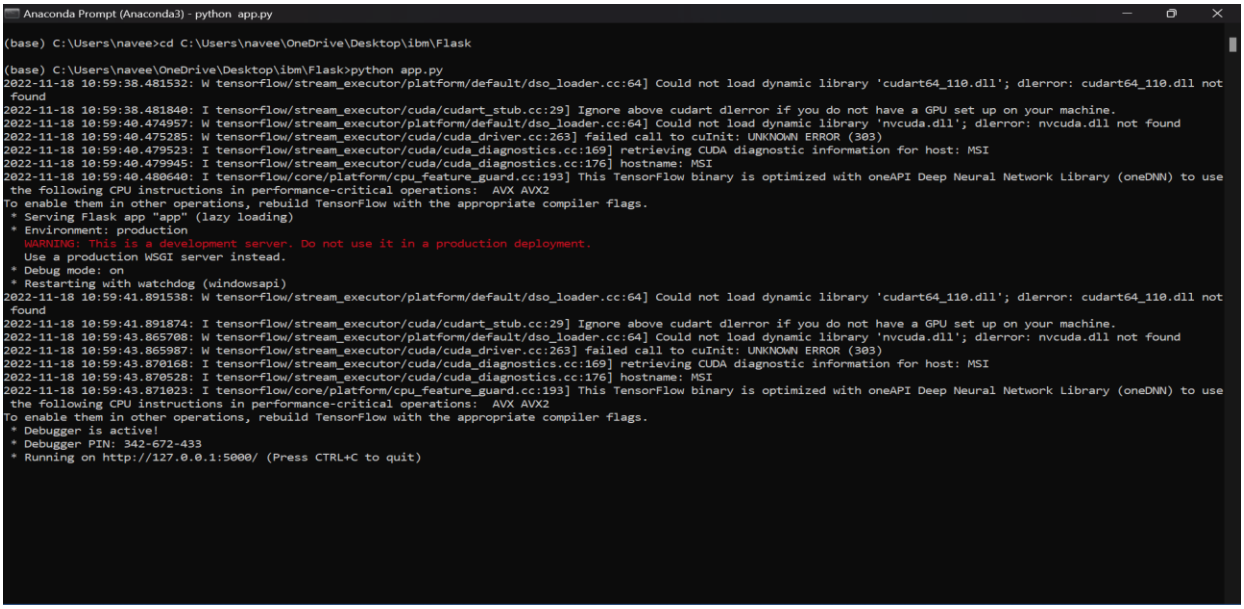
```
import numpy as np
from flask import Flask,render_template,request
from tensorflow.keras.models import load_model
app=Flask(_name_)
model=load_model('crude.h5',)
@app.route('/')
def home():
   return render_template("index.html")
@app.route('/about')
def home1():
   return render_template("index.html")
@app.route('/predict')
def home2():
   return render_template("web.html", showcase="")
@app.route('/login',methods=['POST'])
def login():
   x_input=[]
   for i in request.form:
      x_input.append(float(request.form[i]))
   x_input=np.array(x_input).reshape(1,-1)
   temp_input=list(x_input)
   temp_input=temp_input[0].tolist()
```

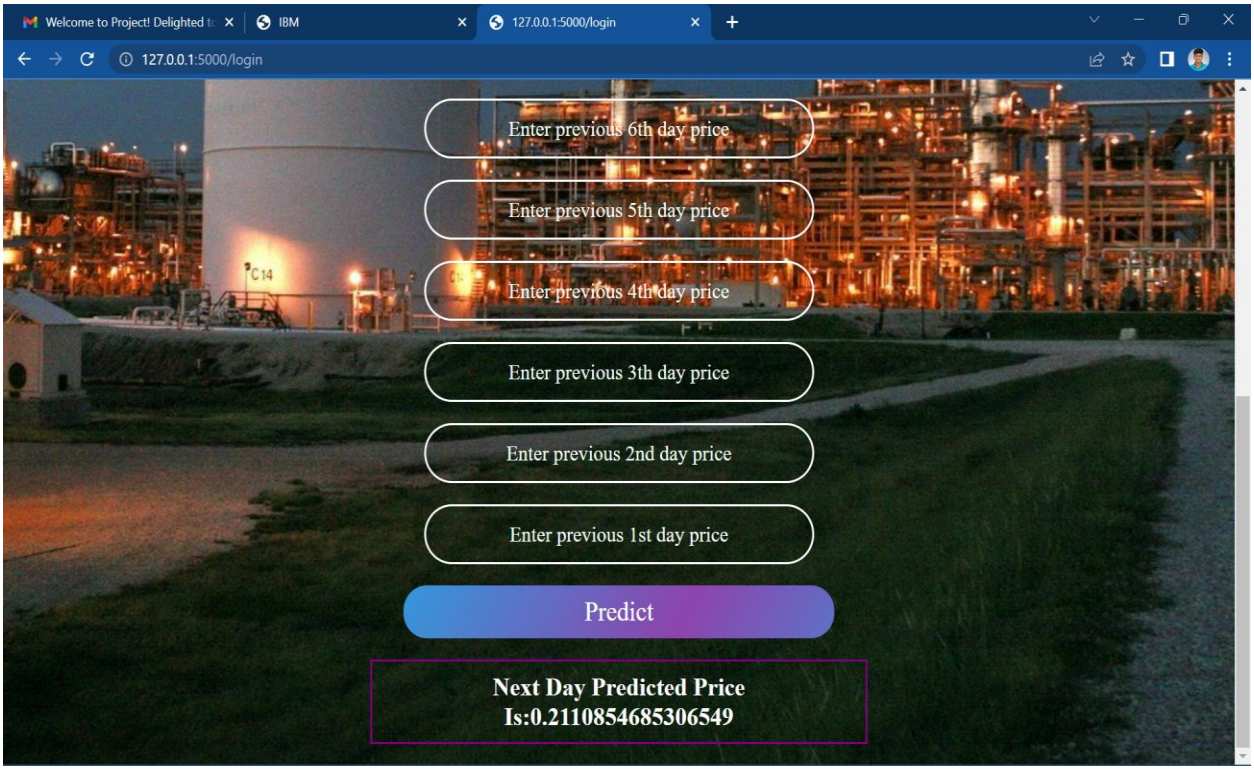
```
    lst_output=[]
    n_steps=10
    i=0
    while(i<1):
        if(len(temp_input)>10):
            x_input=np.array(temp_input[1:])
            print("{}day input{}".format(i,x_input))
            x_input=x_input.reshape(1,-1)
            x_input=x_input.reshape((1,n_steps,1))
            yhat=model.predict(x_input,verbose=0)
            print("{} day output {}".format(i,yhat))
            temp_input.extend(yhat[0].tolist())
            temp_input=temp_input[1:]
            lst_output.extend(yhat.tolist())
            i=i+1
        else:
            x_input=x_input.reshape((1,n_steps,1))
            yhat=model.predict(x_input,verbose=0)
            print(yhat[0])
            temp_input.extend(yhat[0].tolist())
            print(len(temp_input))
            lst_output.extend(yhat.tolist())
            i=i+1

    return render_template("web.html",showcase='Next Day Predicted Price Is:'+str(lst_output[0][0]))

if _name=='main_':
    app.run(debug=True,port=5000)
```


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\navee\OneDrive\Desktop\ibm\Flask\app.py
app.py
import numpy as np
from flask import Flask,render_template,request
from tensorflow.keras.models import load_model
app=Flask(__name__)
model=load_model('crude.h5',)
@app.route('/')
def home():
return render_template("index.html")
@app.route('/about')
def home1():
return render_template("index.html")
@app.route('/predict')
def home2():
return render_template("web.html", showcase="")
@app.route('/login',methods=['POST'])
def login():
x_input=[]
for i in request.form:
x_input.append(float(request.form[i]))
x_input=np.array(x_input).reshape(1,-1)
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
lst_output=[]
n_steps=10
i=0
while(i<1):
if(len(temp_input)>10):
x_input=np.array(temp_input[1:])
print("{}day input{}".format(i,x_input))
x_input=x_input.reshape(1,-1)
x_input=x_input.reshape((1,n_steps,1))
yhat=model.predict(x_input,verbose=0)
print("{} day output {}".format(i,yhat))
temp_input.extend(yhat[0].tolist())
temp_input=temp_input[1:]
C:\Users\navee
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help Variable Explorer Plots Files
Console 1/A
failed call to cuInit: UNKNOWN ERROR (303)
2022-11-18 10:58:02.197496: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: MSI
2022-11-18 10:58:02.197573: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: MSI
2022-11-18 10:58:02.199664: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
LSP Python: ready
conda: base (Python 3.9.12)
Line 20, Col 44
ASCII
CRLF
RW
Mem 75%


Anaconda Prompt (Anaconda3) - python app.py
(base) C:\Users\navee>cd C:\Users\navee\OneDrive\Desktop\ibm\Flask
(base) C:\Users\navee\OneDrive\Desktop\ibm\Flask>python app.py
2022-11-18 10:59:38.481532: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-18 10:59:38.481840: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-18 10:59:40.474957: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-18 10:59:40.475285: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-18 10:59:40.479523: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: MSI
2022-11-18 10:59:40.479945: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: MSI
2022-11-18 10:59:40.480640: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
2022-11-18 10:59:41.891538: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-18 10:59:41.891874: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-18 10:59:43.865708: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-18 10:59:43.865987: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-18 10:59:43.870168: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: MSI
2022-11-18 10:59:43.870528: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: MSI
2022-11-18 10:59:43.871023: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debugger is active!
* Debugger PIN: 342-672-433
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)


Welcome to Project! Delighted t
IBM
127.0.0.1:5000/login
127.0.0.1:5000/login
Enter previous 6th day price
Enter previous 5th day price
Enter previous 4th day price
Enter previous 3th day price
Enter previous 2nd day price
Enter previous 1st day price
Predict
Next Day Predicted Price
Is:0.2110854685306549