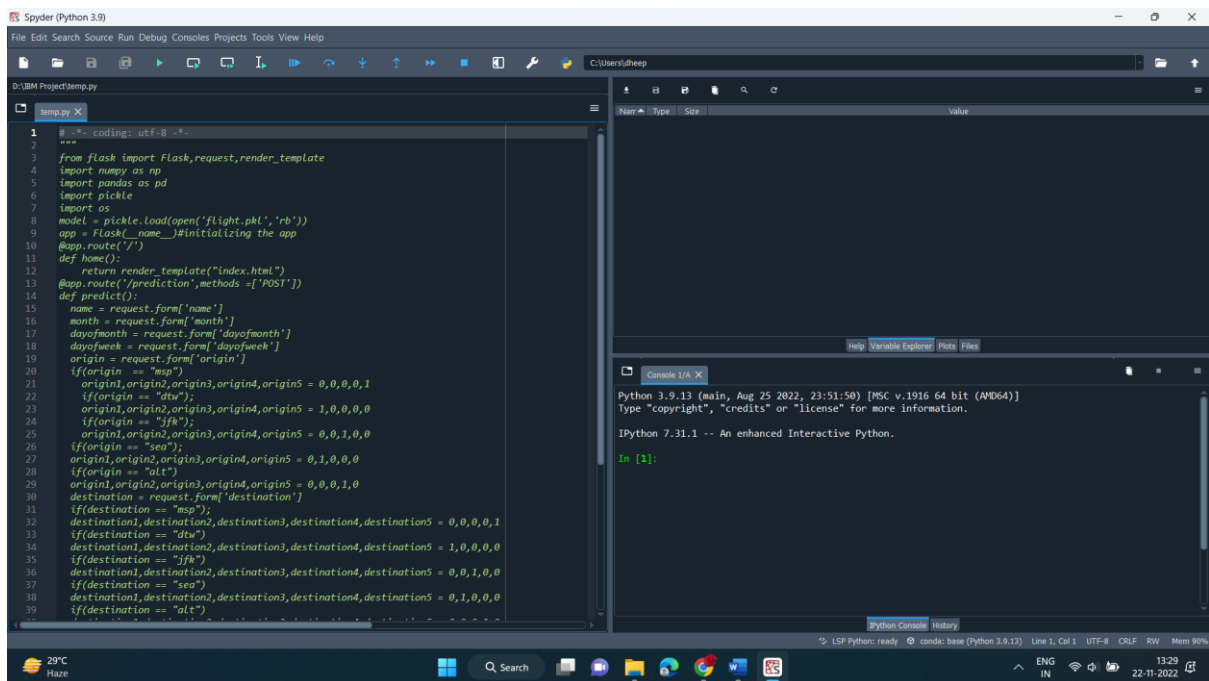


INTEGRATE FLASK WITH SCORING END POINT

DATE	01 NOVEMBER 2022
TEAM ID	PNT2022TMID17753
PROJECT NAME	DEVELOPING A FLIGHT DELAY PREDICTION MODEL BY USING MACHINE LEARNING



The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script named `temp.py` with the following code:

```
1 #-*- coding: utf-8 -*-
2 """
3 from flask import Flask, request, render_template
4 import numpy as np
5 import pandas as pd
6 import pickle
7 import os
8 model = pickle.load(open('flight.pkl', 'rb'))
9 app = Flask(__name__) #initializing the app
10 @app.route('/')
11 def home():
12     return render_template("index.html")
13 @app.route('/prediction', methods = ['POST'])
14 def predict():
15     name = request.form['name']
16     month = request.form['month']
17     dayofmonth = request.form['dayofmonth']
18     dayofweek = request.form['dayofweek']
19     origin = request.form['origin']
20     if(origin == "msp"):
21         origin1,origin2,origin3,origin4,origin5 = 0,0,0,0,1
22         if(origin == "dtw"):
23             origin1,origin2,origin3,origin4,origin5 = 1,0,0,0,0
24         if(origin == "jfk"):
25             origin1,origin2,origin3,origin4,origin5 = 0,0,1,0,0
26         if(origin == "sea"):
27             origin1,origin2,origin3,origin4,origin5 = 0,1,0,0,0
28         if(origin == "alt"):
29             origin1,origin2,origin3,origin4,origin5 = 0,0,0,1,0
30     destination = request.form['destination']
31     if(destination == "msp"):
32         destination1,destination2,destination3,destination4,destination5 = 0,0,0,0,1
33     if(destination == "dtw"):
34         destination1,destination2,destination3,destination4,destination5 = 1,0,0,0,0
35     if(destination == "jfk"):
36         destination1,destination2,destination3,destination4,destination5 = 0,0,1,0,0
37     if(destination == "sea"):
38         destination1,destination2,destination3,destination4,destination5 = 0,1,0,0,0
39     if(destination == "alt")
```

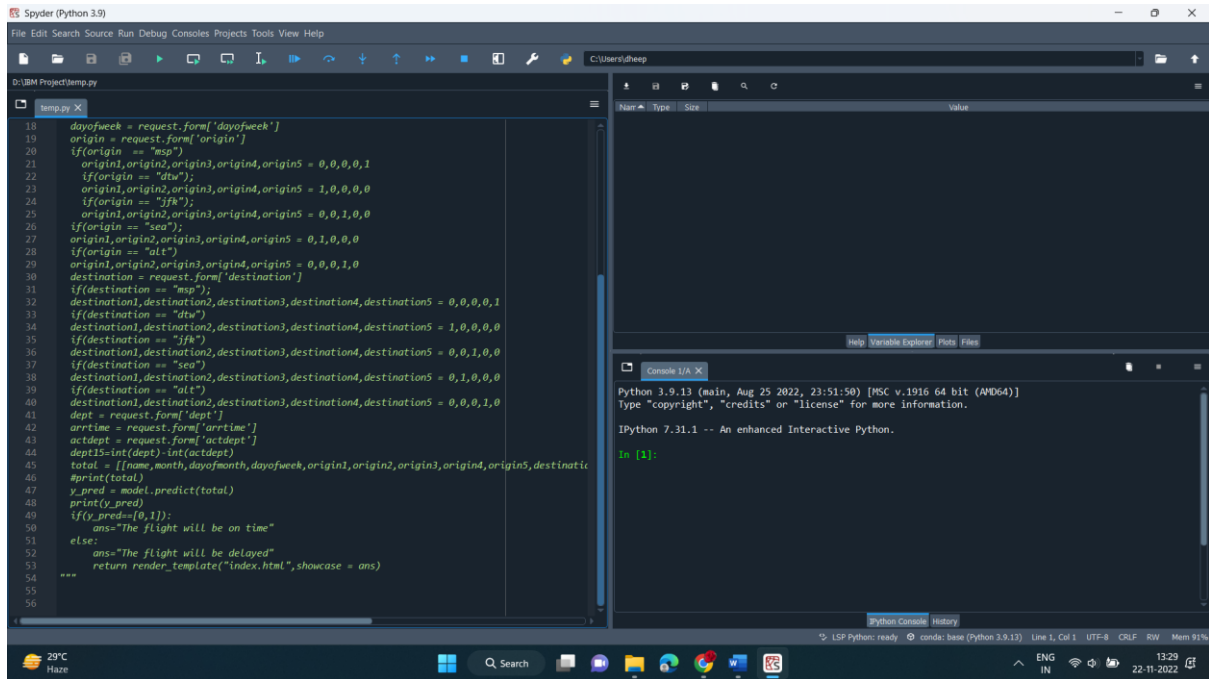
The right-hand side of the IDE shows the IPython console with the following output:

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]:
```

The bottom status bar indicates the environment is LSP Python, ready, using conda: base (Python 3.9.13), with line 1, column 1, UTF-8 encoding, and 90% memory usage.



```

def home():
return render_template('index.html')

def y_predict():
geography = request.form("geography")
gender= request.form("gender")
age= request.form("age")
tenure= request.form("tenure")
creditscore= request.form("creditscore")
balance= request.form("balance")
noof = request.form["no of"]
hascreditcard = request.form{ "has credit card"}
isactivemember = request.form{ "isactivemember"}
estimatedsalary = request.form["estimated salary"]
if(geography == "Spain"):
s1,s2,s3 = 0,0,1
if(geography == "Germany"):
s1,s2,s3 = 0,1,0

```

```

if(geography == "Newyork"):
s1,s2,s3 = 0,1,0
if(gender == "female"):
gender = 0
if(gender == "male"):
gender = 1
if(isactivemember == "no"):
isactivemember = 0
if(isactivemember == "yes"):
isactivemember = 1
if(hascreditcard == "no"):
hascreditcard = 0
if(hascreditcard == "yes"):
hascreditcard = 1
t=[[int(s1),int(s2),int(s3),int(creditscore),int(gender),int(age),int(tenure),int(balance),int(noof)
,int(hascreditcard),int(isactivemember),int(estimatedsalary)]]
print(t)
payload_scoring =
("input_data":[{"field":["G1","G2","G3","CreditScore","Gender","Age","Tenure","Balance
","numofproducts","HasCrCard","IsActiveMember"])
print("Scoring response')
predictions = response_scoring.json()
print(predictions)
pred = predictions['predictions'][0]['values'][0][0]
if(pred == 0):
output = "he will not get excited"
print(" he will not get excited")
else:
output = "he gets excited"
print("he gets excited")
return render_template("index.html",prediction_text=output)

```

```
if __name__ == "__main__":  
    app.run(debug=True)
```