Sprint-3

Project name	Developing a flight delay prediction using machine learning
Team ID	PNT2022TMID04987

FLASK: (app.ibm.py)

```
import time

import requests


import flask

from flask import request, render_template

from flask_cors import CORS

import requests


# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.

API_KEY = "give your api key"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:4fJbepuok7oCO1UkeKU831Sq5rz5-JP0R_hljeB2oaEL'})

mltoken = token_response.json()["access_token"]
```

```
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}



app = flask.Flask(__name__,
static_url_path='c:/Users/ELCOT/Desktop/templates/app.py')

CORS(app)

@app.route('/', methods=['GET'])

def sendHomePage():

    return render_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

    EnterflightNumber  = str(request.form['Enter  flight Number'])

    Month = int(request.form['month'])

    Dayofmonth=  int(request.form['day  of  month'])

    Dayofweek= float(request.form['day of week'])

    origin=float(request.form['origin'])

    Destination=float(request.form['destination'])

    scheduleddeparturetime=time(request.form['scheduled departure
time'])

    scheduledarrivaltime=time(request.form['scheduled arraival time'])

    Actualtime=time(request.form['actual  time'])

    X = [('Enter flight Number,Month,Day of month,Day of
week,origin,Destination,scheduled departure time,scheduled arrival
time,actual  time')]
```

```
    payload_scoring =
"input_data";[("field")];("EnterflightNumb','Month','Dayofmonth','Dayof
week','origin','
Destination','scheduledarrivaltime,'scheduledarrivaltime','actual time"),
"values": X

    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/a7a269f3-d3c1-4e2d-85b2-
47e1bf6bbfee/predictions?version=2022-10-13', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})

    print(response_scoring)

    predictions = response_scoring.json()

    predict = predictions['predictions'][0]['values'][0][0]

    print("Final prediction :",predict)


    # showing the prediction results in a UI# showing the prediction
results in a UI

    return render_template('predict.html', predict=predict)


if __name__== '__main__' :

    app.run(debug= False)
```