```
import
pandas as

        import numpy as np

        import requests

        import os



        from flask import Flask,request, render_template

        app=Flask(__name__,template_folder='templates')


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

        def index():

            return render_template('index.html')

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

        def about():

            return render_template('intro.html')

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

        def page():

            return render_template('upload.html')

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

        def predict():

            print("[INFO] loading model...")

            input_features = [float(x) for x in request.form.values()]

            features_value = [input_features]

            print(features_value)



            features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',

                'city_code', 'region_code', 'category']

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

            API_KEY = "V0FedIvcsn9vpDN7cIG2cmB8T8zpenX6vPs8tufhqE6b"

            token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":

            API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})

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

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

    # NOTE: manually define and pass the array(s) of values to be scored in the next line

    payload_scoring = {"input_data": [{"values": features_value}]}

    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/cfbed64a-29cb-44e2-bc53-
e0a418c3077e/predictions?version=2022-11-14', json=payload_scoring,

    headers={'Authorization': 'Bearer ' + mltoken})

    print("Scoring Endpoint")

    print(response_scoring.json())

    pred = response_scoring.json()


    output=pred['predictions'][0]['values'][0][0]

    print(output)

    return render_template('upload.html', prediction_text=output)



if __name__ == '__main__':

      app.run()
```