SPRINT-4

TRAIN THE MODEL ON IBM

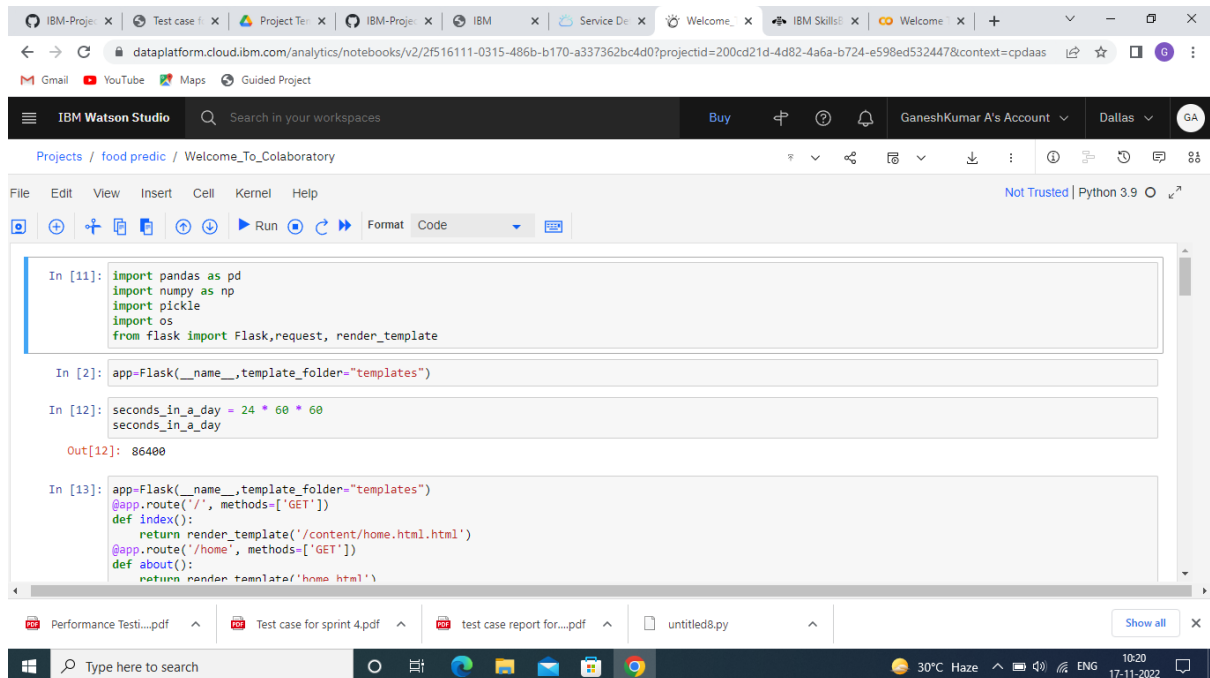| Team id	PNT2022TMID45520
Project name	DemandEst - AI powered Food Demand Forecaster

Register For IBM Cloud :

- Please   register for IBM
- Please log in to IBM Account
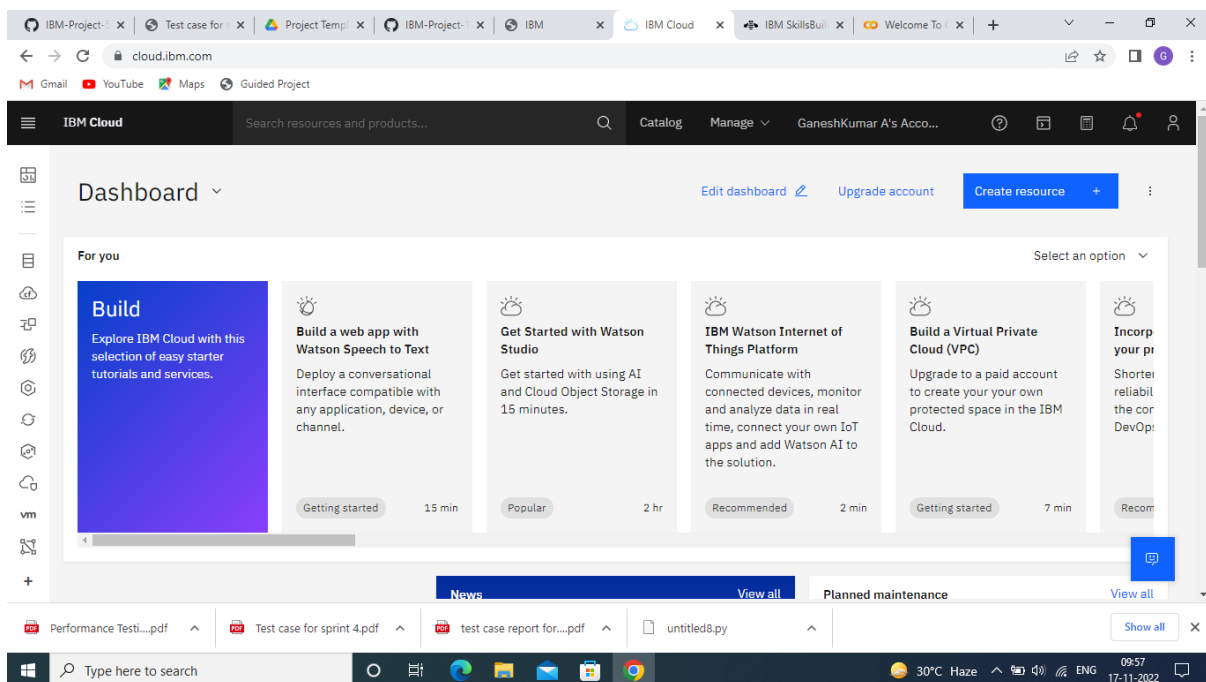
Train The ML Model On IBM :

Integrate Flask With Scoring End Point :

import the necessary packages

import Integrate Flask With Scoring End Point :

 # pandas as pd

import numpy as np

import pickle

import os

import requests

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

```python
API_KEY = "68w9XBNJLBQFtHM2rG_aouV4LmlF-EtecYrhIQBQbt_K"

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}



from flask import Flask, request, render_template



app = Flask(__name__, template_folder="/content/app.py")



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

def index():

    return render_template('/content/home.html.htmlhome.html')




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

def about():

    return render_template('/content/home.html.html')




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

def page():

    return render_template('/content/home.html.html')
```

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

def predict():

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

    # model = pickle.load(open('fdemand.pkl', 'rb'))

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

    print(input_features)

    features_value = [[np.array(input_features)]]

    print(features_value)



    payload_scoring = {"input_data": [{"field": [['homepage_featured',
'emailer_for_promotion', 'op_area', 'cuisine',

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

                        "values": [input_features]}]}



    response_scoring = requests.post(

        'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/80afcaad-591d-4869-bf54-
17bbb8c70ea3/predictions?version=2022-11-14',

        json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})

    print("Scoring response")

    print(response_scoring.json())

    predictions = response_scoring.json()

    print(predictions)

    print('Final Prediction Result', predictions['predictions'][0]['values'][0][0])

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

```
    # prediction = model.predict(features_value)

    # output=prediction[0]

    # print(output)'''


print(pred)

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




if __name__ == '__main__':

    app.run(debug=False)
```