Week4: Deployment on Flask

Name: Shreya Dwivedi
Batch code: LISUM 13:30
Submission date: 10/01/2022

Step 1: Create a Machine Learning Model, eg: Linear Regression Model

```python
from operator import le
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings('ignore')
import pickle

data=pd.read_excel("CCPP/Folds5x2_pp.xlsx")
df=pd.DataFrame(data)

y=df['PE']
X=df.drop(['PE'],axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
liner_reg=LinearRegression()
liner_reg.fit(X_train,y_train)
print(X_test)
y_pred=liner_reg.predict(X_test)

pickle.dump(liner_reg, open('model.pkl','wb'))
```

Step 2: Create a index.html file which will be the homepage of the app and will contain a form which will trigger the requests of the user such as Get/Post/Put/Delete, etc.

```html
<!DOCTYPE html>
<html lang="en">
<head>
<body>
    <form action='/predict' method="post" class="col s12">
        <div class="row">
            <div class="input-field col s4">
                <label for="first_name"><b>Temperature</b></label>
                <br>
                <input placeholder="Temperature in Celsius" name="Ambient Temperature" id="first_name" type="text" class="validate">
            </div>
            <div class="input-field col s4">
                <label for="last_name"><b>Exhaust Vacuum </b></label>
                <br>
                <input id="last_name" name="Exhaust Vacuum" placeholder="Exhaust Vacuum" type="text" class="validate">
            </div>
            <div class="input-field col s4">
                <label for="_name"><b>Ambient Pressure</b></label>
                <br>
                <input id="_name" name="Ambient Pressure" placeholder="Ambient Pressure" type="text" class="validate">

            </div>

            <div class="input-field col s4">
                <label for="_name"><b>Relative Humidity</b></label>
                <br>
                <input id="_name" name="Relative Humidity" placeholder="Relative Humidity" type="text" class="validate">

            </div>
            <div class="row center">

                <button type="submit" class="btn-large waves-effect waves-light orange">Predict Electrical Energy</button>
            </div>
        </form>
            <br>
                {{pred}}<br>
            </div>
        </div>
            <br><br>
        </div>
        </div>>
        </body>
    </html>
```
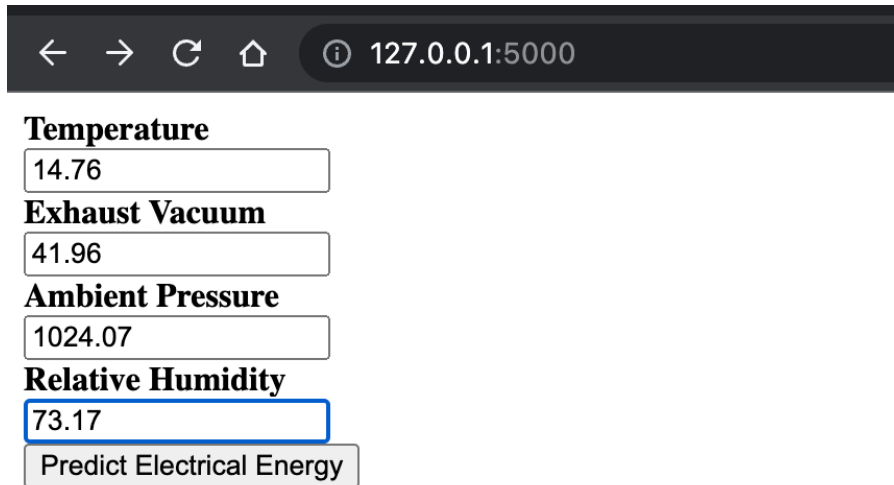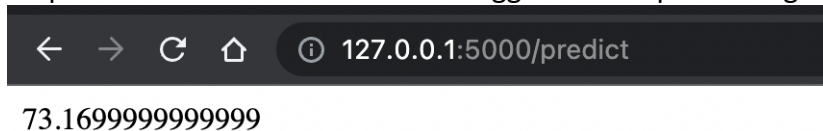
Step 3: Create an app which will redirect to a web page to take features as input and predict corresponding y:

```python
from crypt import methods
from flask import Flask,request, url_for, redirect, render_template
import pickle
import numpy as np

app=Flask(__name__)
model=pickle.load(open('model.pkl','rb'))

@app.route('/')
def home():
    return render_template('./index.html')

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

    int_features=[float(x) for x in request.form.values()]
    final=[int_features]
    # print("##########1: ",int_features)
    # print("##########2: ",final)
    prediction=model.predict(np.array(final))
    print("##########3: ",prediction)
    output= str(prediction[0])
    return output


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

Step 4: Run the above app and start the server

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER
○ (base) shreyadwivedi@Shreyas-Air Data_Glacier % /usr/bin/python3 /Users/shreyadwivedi/Downloads/Data_Glacier/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 422-093-992
```

Step 5: Open the server on the browser

← → C ⌂ ⓘ 127.0.0.1:5000

Temperature
14.76

Exhaust Vacuum
41.96

Ambient Pressure
1024.07

Relative Humidity
73.17

Predict Electrical Energy

Step 6: Enter Values in the form to trigger Post request and get the predicted Y value.

← → C ⌂ ⓘ 127.0.0.1:5000/predict

73.1699999999999