

SPRINT 4

Project Deliverables (Flask Code & Deployment)

Team ID	PNT2022TMID35159
Project Name	Efficient Water Quality Analysis & Prediction using Machine Learning

App.py:

```
app.py x Water_quality.ipynb home.html 2 water_potability.csv
app.py > Python > hello
1 from flask import Flask, request, render_template
2 import pickle
3 import pandas as pd
4 import numpy as np
5 import joblib
6 scaler = joblib.load("my_scaler.save")
7
8
9 app = Flask(__name__)
10 model = pickle.load(open('model.pkl', 'rb'))
11
12 @app.route("/home")
13 @app.route("/")
14 def hello():
15     return render_template("home.html")
16
17 @app.route("/predict", methods = ["GET", "POST"])
18 def predict():
19     if request.method == "POST":
20         input_features = [float(x) for x in request.form.values()]
21         features_value = [np.array(input_features)]
22
23         feature_names = ["ph", "Hardness", "Solids", "Chloramines", "Sulfate",
24                         "Conductivity", "Organic_carbon", "Trihalomethanes", "Turbidity"]
25
26         df = pd.DataFrame(features_value, columns = feature_names)
27         df = scaler.transform(df)
28         output = model.predict(df)
29
30         if output[0] == 1:
31             prediction = "safe"
32         else:
33             prediction = "not safe"
```

The flask file (app.py) which we have used as a framework which will present (home.html) file to user and model.pkl file to use the trained model to predict whether *the water is safe for consumption or not*

```
@app.route("/predict", methods = ["GET", "POST"])
def predict():
    if request.method == "POST":
        input_features = [float(x) for x in request.form.values()]
        features_value = [np.array(input_features)]

        feature_names = ["ph", "Hardness", "Solids", "Chloramines", "Sulfate",
                          "Conductivity", "Organic_carbon", "Trihalomethanes", "Turbidity"]

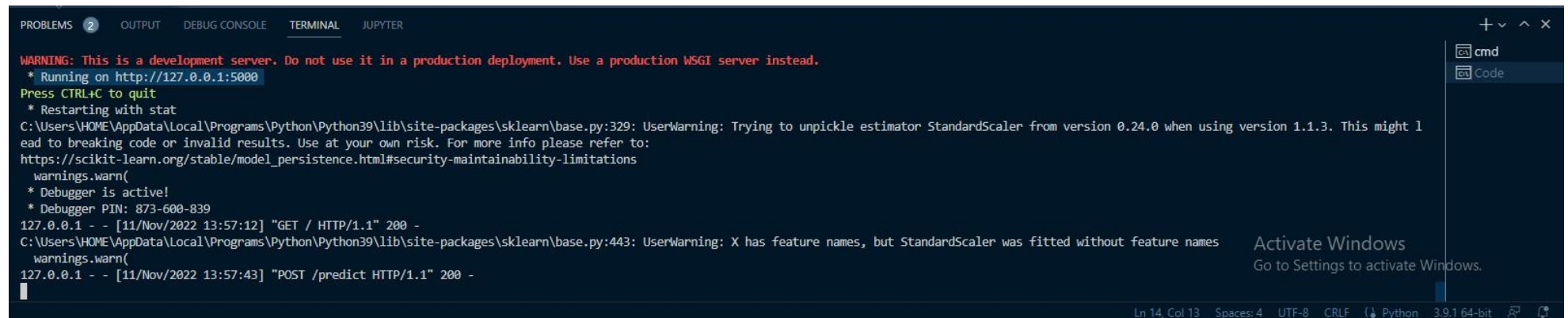
        df = pd.DataFrame(features_value, columns = feature_names)
        df = scaler.transform(df)
        output = model.predict(df)

        if output[0] == 1:
            prediction = "safe"
        else:
            prediction = "not safe"

        return render_template('home.html', prediction_text= "water is {} for human consumption ".format(prediction))

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

To run our ML model, we have to run **app.py** model where it gives a port number in terminal. We have to copy and paste that link in our browser to use the prediction model



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
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
C:\Users\HOME\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator StandardScaler from version 0.24.0 when using version 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to: https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
* Debugger is active!
* Debugger PIN: 873-600-839
127.0.0.1 - - [11/Nov/2022 13:57:12] "GET / HTTP/1.1" 200 -
C:\Users\HOME\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:443: UserWarning: X has feature names, but StandardScaler was fitted without feature names
warnings.warn(
127.0.0.1 - - [11/Nov/2022 13:57:43] "POST /predict HTTP/1.1" 200 -
```

In our case, it is running on <http://127.0.0.1:5000> (the default port number for flask is 5000)

OUTPUT:

The screenshot displays a web browser window with the URL `127.0.0.1:5000/predict`. The page has a light blue background and is titled Water Quality_prediction. Below the title, there is a section labeled "Enter values" in blue text. This section contains nine input fields arranged in three rows:

- Row 1: `pH value`, `Hardness`, and `Solids`.
- Row 2: `Chloramines`, `Sulfate`, and `Conductivity`.
- Row 3: `Organic_carbon`, `Trihalomethanes`, and `Turbidity`.

Each input field is a light gray box with its respective parameter name inside. Below these fields is a prominent blue button labeled "Water quality Test". Underneath the button, the prediction result is displayed in a large, dark gray font: "water is safe for human consumption". In the bottom right corner, there is a small, faint watermark that reads "Activate Windows Go to Settings to activate Windows."

The browser's address bar shows several open tabs, including "Gmail", "YouTube", "Maps", "perfect Plan B # 25...", "My Courses | Perfe...", "Python for Empho...", "Home | Codewar", "Certifications | InfoTQ", "ACTE Internship En...", "Online Courses - La...", "Driver's IDE Sheet...", "TCS NQT 2023 Re...", "Last 30 days of TCS...", and "Bubble Sort Algori...". The Windows taskbar at the bottom shows the search bar, task view button, and several application icons (File Explorer, Edge, Chrome, etc.). The system clock in the bottom right corner indicates the time is 04:30 PM on 01-11-2022.

Test case 1 : (water is safe for human consumption)

127.0.0.1:5000/predict

Water Quality_prediction

Enter values

pH value : Hardness : Solids :

Chloramines : Sulfate : Conductivity :

Organic_carbon : Trihalomethanes : Turbidity :

Water quality Test

water is safe for human consumption

Activate Windows
Go to Settings to activate Windows.

Type here to search

04:59 PM 11-11-2022

Test case 2: (water is not safe for human consumption)

127.0.0.1:5000/predict

Water Quality_prediction

Enter values

pH value : Hardness : Solids :

Chloramines : Sulfate : Conductivity :

Organic_carbon : Trihalomethanes : Turbidity :

Water quality Test

water is not safe for human consumption

Activate Windows
Go to Settings to activate Windows

Type here to search

05:00 PM
11-11-2022