# Practical-5

**Aim**: Deployment of ML projects using Flask Deploying machine learning models with Flask in an ML lab is crucial because it enables researchers and data scientists to turn their models into accessible APIs or web applications. This facilitates collaboration, testing, and real-world usage of ML models, making them more practical and applicable to various domains. Flask provides a lightweight, flexible framework for building and deploying these interfaces quickly and efficiently.

In this lab, you will deploy the ML project using Flask. Perform the following tasks:

Task 1: Install the required libraries:

Install Flask library (https://flask.palletsprojects.com/en/2.3.x/installation/ )

```
C:\Users\gchaw\OneDrive\Documents\sem7\ML_OPS\p4>pip install flask
Requirement already satisfied: flask in c:\users\gchaw\appdata\local\program
s\python\python311\lib\site-packages (2.3.3)Requirement already satisfied: W
erkzeug>=2.3.7 in c:\users\gchaw\appdata\local\programs\python\python311\lib
\site-packages (from flask) (2.3.7)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\gchaw\appdata\local
\programs\python\python311\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\gchaw\appdata
\local\programs\python\python311\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\gchaw\appdata\local\
programs\python\python311\lib\site-packages (from flask) (8.1.6)
Requirement already satisfied: blinker>=1.6.2 in c:\users\gchaw\appdata\loca
l\programs\python\python311\lib\site-packages (from flask) (1.6.2)
Requirement already satisfied: colorama in c:\users\gchaw\appdata\local\prog
rams\python\python311\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\gchaw\appdata\loc
al\programs\python\python311\lib\site-packages (from Jinja2>=3.1.2->flask) (
2.1.3)
```

Task 2: Follow the steps described in theory material to deploy the model using Flask. Run the flask application to execute the deployed model. The steps are briefly described in the following:

● Create the Templates for the various views of the project

Index.html

```html
<html>
<body>
<h3>House Rate Prediction Form</h3>
<br>
<div>
<form action="/result" method="POST">
<label for="area_feature">House Area [in sq. feet] : </label>
<input type="text" id="area_feature" name="area_feature"> range:[1500-15000]
<br>
<br>
<input type="submit" value="Submit">
</form>
```

```html
</div>
</body>
</html>
```

Result.html

```html
<!doctype html>
<html>
<body>
<h1> The predicted amount of house is {{ prediction }}</h1>
</body>
</html>
```

```python
from flask import Flask, render_template, jsonify, request
import numpy as np
import pickle


app = Flask(__name__)


def ValuePredictor(to_predict_list):
  X_test = np.array(to_predict_list).reshape(1, 1)


  #Load the instance of Standarscalar object
```

```
scaler = pickle.load(open("scaler", "rb"))
#Normalize the data
X_test_Normalized = scaler.transform(X_test)
```

● Import the Model, Dataset, and Scalar objects into the project folder

« ML_OPS › p4 ›          ⌄   C      Search p4                              🔎

| Name | Date modified | Type | Size |
|------|---------------|------|------|
| 📁 __pycache__ | 26-09-2023 15:08 | File folder | |
| 📁 templates | 26-09-2023 14:42 | File folder | |
| 📄 app | 26-09-2023 15:07 | Python Source File | |
| 📄 model | 26-09-2023 15:01 | File | |
| 📄 scaler | 26-09-2023 15:01 | File | |
| 📄 script1 | 03-10-2023 15:01 | Python Source File | |

● Create the script.py file to serve the deployment

App.py

```
  loaded_model = pickle.load(open("model", "rb"))
  result = loaded_model.predict(X_test_Normalized)
  return result[0]


@app.route('/result', methods = ['POST'])
def result():
  if request.method == 'POST':
    to_predict_list = request.form.to_dict()
    to_predict_list = list(to_predict_list.values())
    to_predict_list = list(map(int, to_predict_list))
    prediction = ValuePredictor(to_predict_list)


    return render_template("result.html", prediction = prediction)


@app.route("/")
def hello_world():
  return render_template("index.html")
```

**OUTPUT:**

```
C:\Users\gchaw\OneDrive\Documents\sem7\ML_OPS\p4>flask --app app.py run
 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [03/Oct/2023 23:27:47] "GET / HTTP/1.1" 200 -
```

**House Rate Prediction Form**

House Area [in sq. feet] : 1600          range:[1500-15000]

Submit

**The predicted amount of house is 1872.8747612793236**