Practical-5

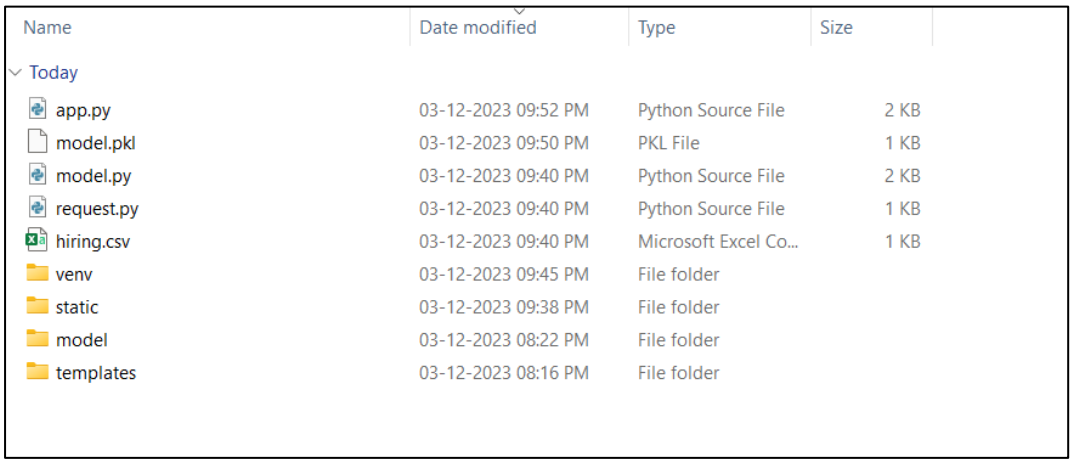
Aim: Deployment of ML project using Flask

Task 1: Install the required libraries:

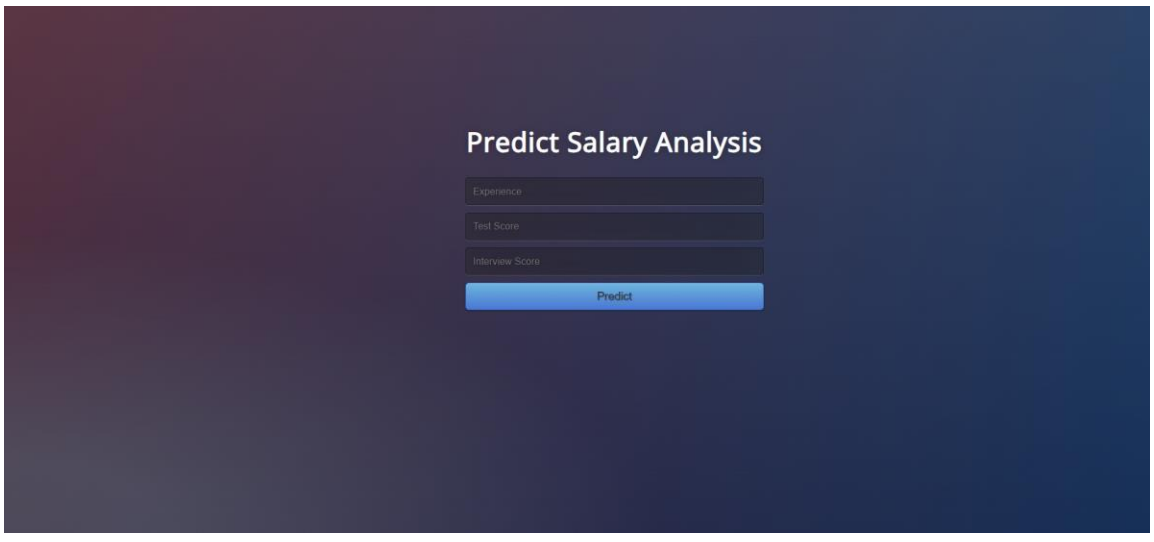
Install Flask library (https://flask.palletsprojects.com/en/2.3.x/installation/)

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:

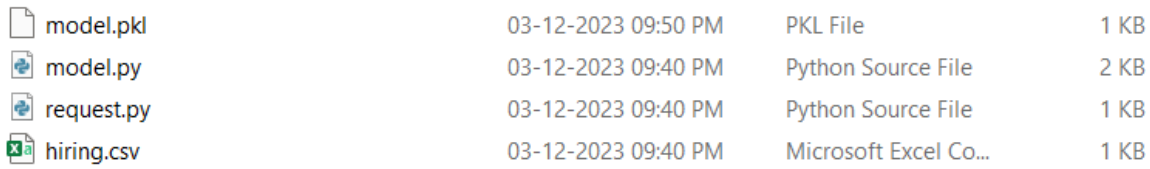
- Create the Templates for the various views of the project



Name	Date modified	Type	Size
Today			
app.py	03-12-2023 09:52 PM	Python Source File	2 KB
model.pkl	03-12-2023 09:50 PM	PKL File	1 KB
model.py	03-12-2023 09:40 PM	Python Source File	2 KB
request.py	03-12-2023 09:40 PM	Python Source File	1 KB
hiring.csv	03-12-2023 09:40 PM	Microsoft Excel Co...	1 KB
venv	03-12-2023 09:45 PM	File folder	
static	03-12-2023 09:38 PM	File folder	
model	03-12-2023 08:22 PM	File folder	
templates	03-12-2023 08:16 PM	File folder	



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



- Create the app.py file to serve the deployment.

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
from sklearn.linear_model import LinearRegression

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

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Employee Salary should be $
{}'.format(output))

@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)
```

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

app.py > ...

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
from sklearn.linear_model import LinearRegression

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

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
```