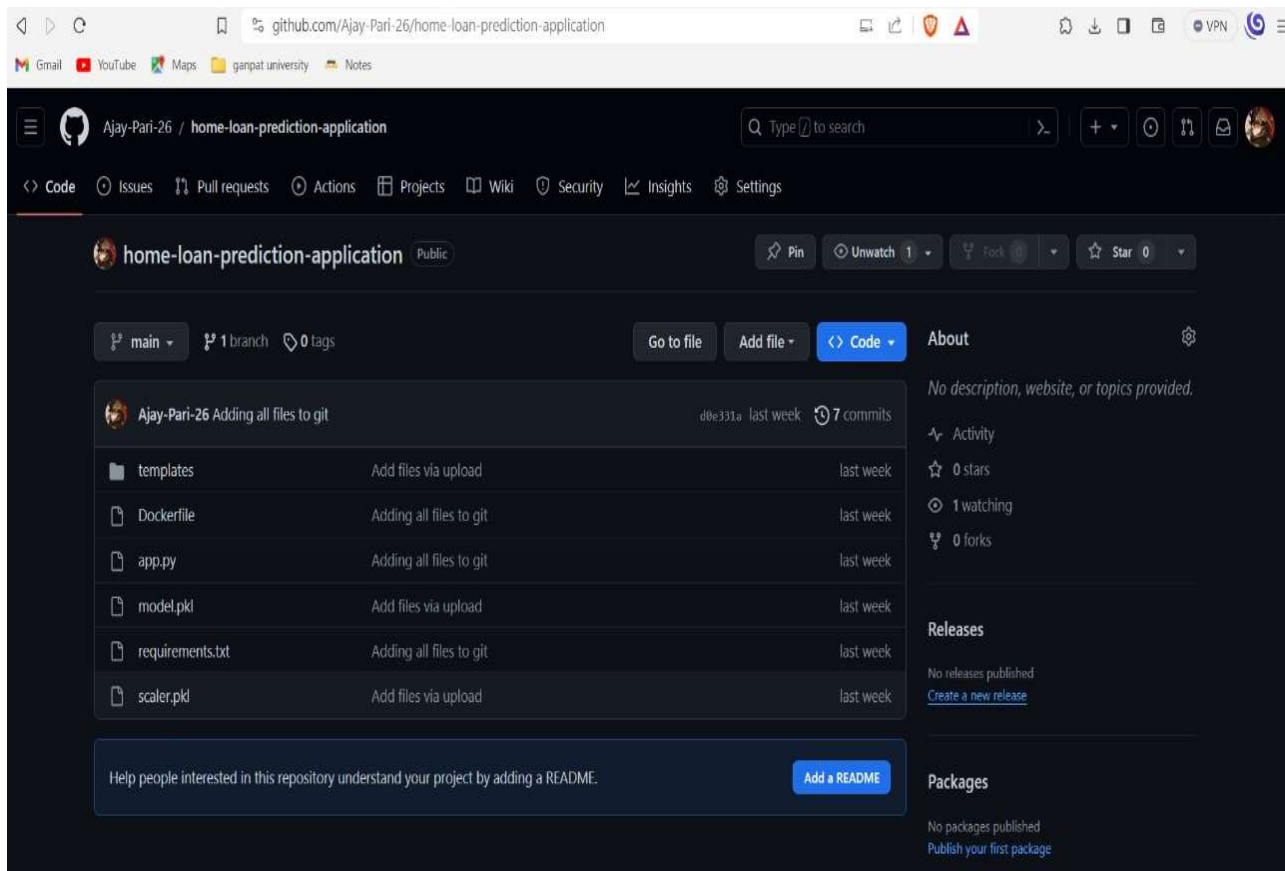


Practical-3

Generation of Reproducible and Interactive ML Project.

Task 1: Create the Github repository for the house rate prediction project created in practical 2.



Task 2: Integrate your repository with the binder to make your project interactive. (Hint: refer to the following link for the steps: (<https://mybinder.org/>))

Build and launch a repository

GitHub repository name or URL

GitHub

Git ref (branch, tag, or commit)

Path to a notebook file (optional)

Copy the URL below and share your Binder with others:

Expand to see the text below, paste it into your README to show a binder badge:

Waiting

Build logs [view raw](#) [hide](#)

```
Waiting for build to start...
Picked Git content provider.
Cloning into '/tmp/repo2dockerljsqnnf'...
HEAD is now at ed72869 Update and rename requirements.txt.txt to requirements.txt
```

notebooks.gesis.org/binder/jupyter/user/pariajay123-ml-project-0fq1gqk4/lab/tree/house%20price%20prediction/a...

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ house price prediction /

Name	Last Modified
templates	3 minutes ago
app.py	3 minutes ago
model.pkl	3 minutes ago
requiremen...	3 minutes ago
scaler.pkl	3 minutes ago

Launcher

```
1 from flask import Flask, render_template, jsonify, request
2 import numpy as np
3 import pickle
4 app = Flask(__name__)
5 def ValuePredictor(to_predict_list):
6     X_test = np.array(to_predict_list).reshape(1, 1)
7     #Load the instance of StandardScalar object
8     scaler = pickle.load(open("scaler.pkl", "rb"))
9     #Normalize the data
10    X_test_Normalized = scaler.transform(X_test)
11    loaded_model = pickle.load(open("model.pkl", "rb"))
12    result = loaded_model.predict(X_test_Normalized)
13    return result[0]
14 @app.route('/result', methods = ['POST'])
15 def result():
16     if request.method == 'POST':
17         to_predict_list = request.form.to_dict()
18         to_predict_list = list(to_predict_list.values())
19         to_predict_list = list(map(int, to_predict_list))
20         prediction = ValuePredictor(to_predict_list)
21         return render_template("result.html", prediction = prediction)
22 @app.route("/")
23 def hello_world():
24     return render_template("home.html")
```

Would you like to receive official Jupyter news?
Please read the privacy policy.

[Open privacy policy](#)