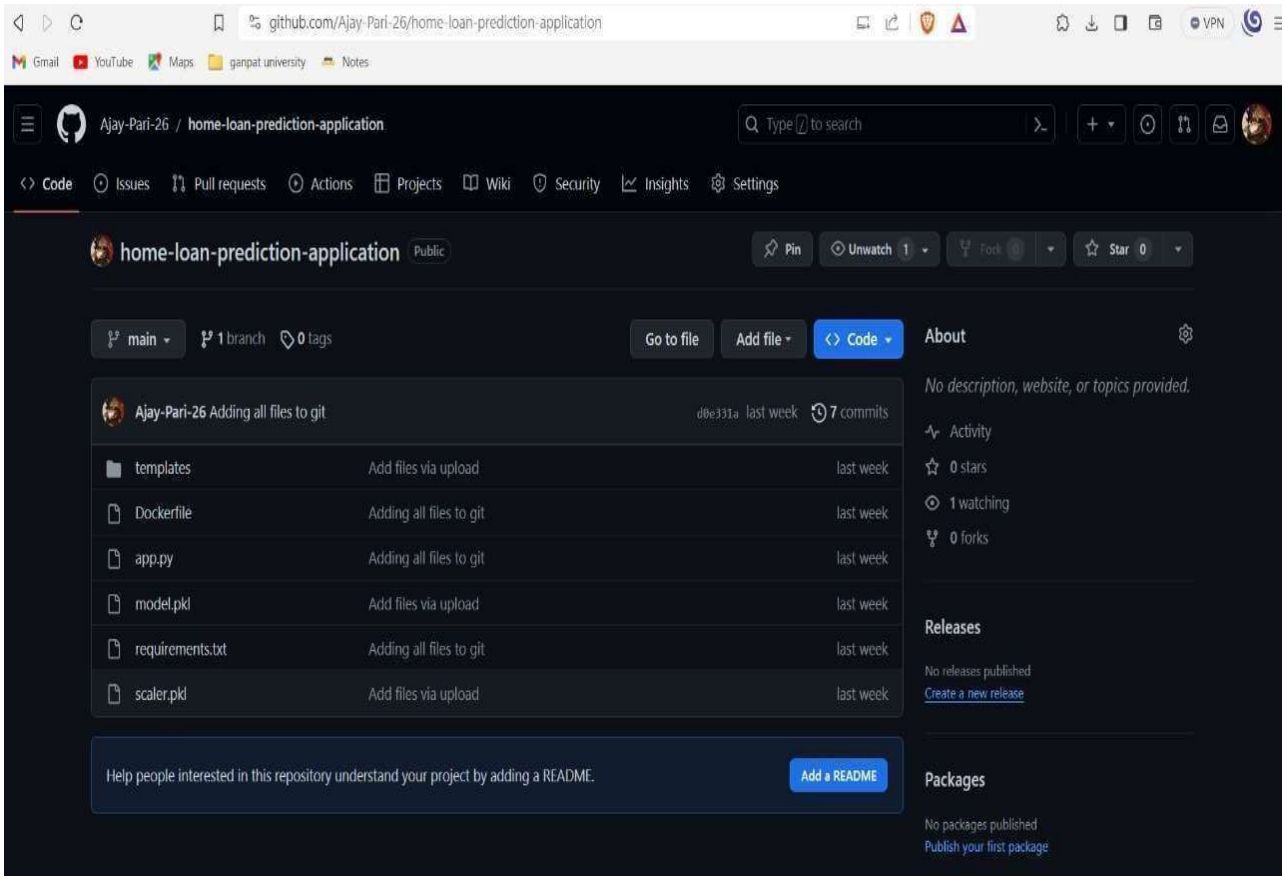
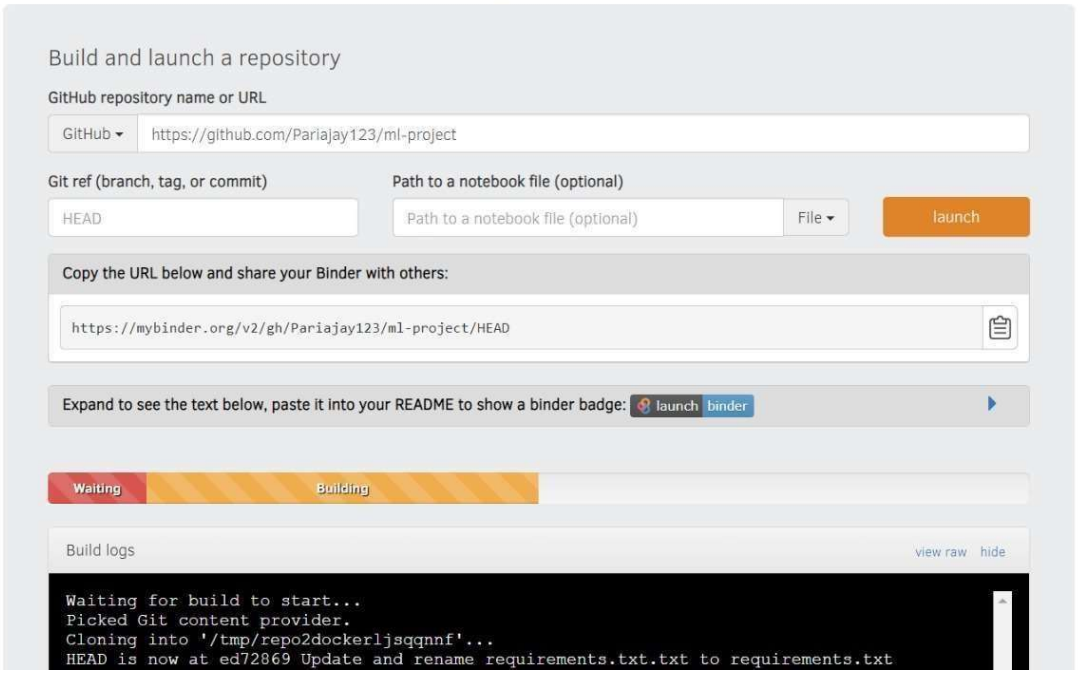
Practical-3

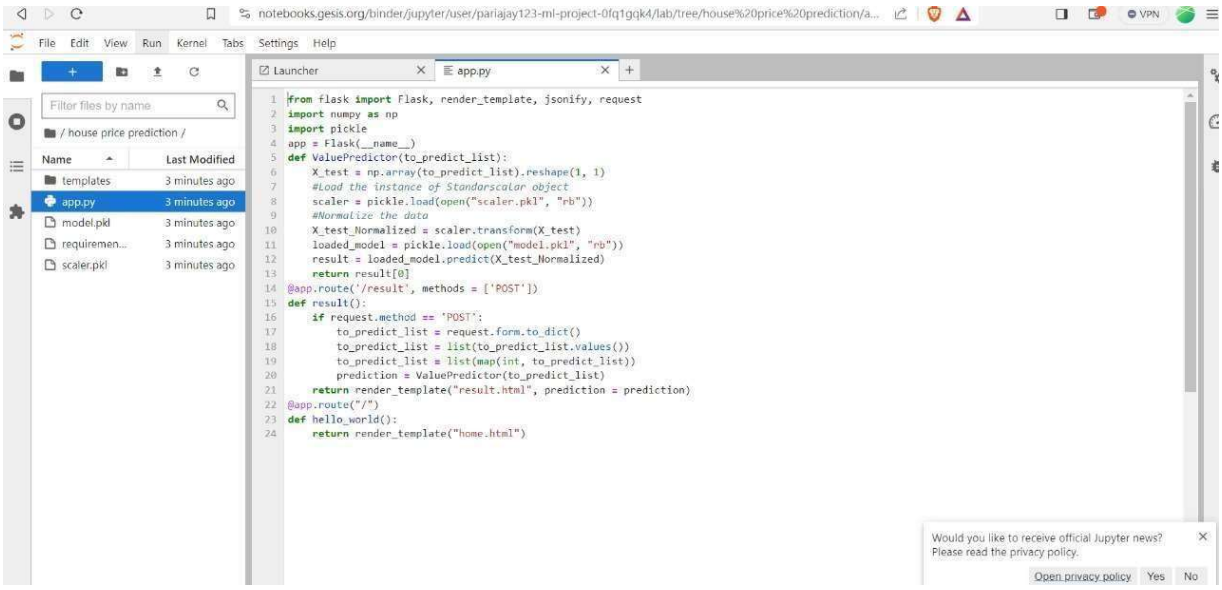
Generation of Reproducible and Interactive ML Project.

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
https://github.com/Pariajay123/ml-project
Git ref (branch, tag, or commit)
HEAD
Path to a notebook file (optional)
Path to a notebook file (optional)
File
launch
Copy the URL below and share your Binder with others:
https://mybinder.org/v2/gh/Pariajay123/ml-project/HEAD
Expand to see the text below, paste it into your README to show a binder badge:
launch binder
Waiting
Building
Build logs
view raw
hide
Waiting for build to start...
Picked Git content provider.
Cloning into '/tmp/repo2dockerljsqqnnf'...
HEAD is now at ed72869 Update and rename requirements.txt.txt to requirements.txt


File Edit View Run Kernel Tabs Settings Help
Launcher
app.py
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
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.pkl", "rb"))
#Normalize the data
X_test_Normalized = scaler.transform(X_test)
loaded_model = pickle.load(open("model.pkl", "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("home.html")
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No