

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.

The screenshot shows a GitHub repository page for 'MLOPs_main_21012532006', which is public. It has 1 branch (master) and 0 tags. The repository was created by 'kruti2021' with an initial commit 6 minutes ago. The commit message is 'Initial commit'. The files listed are: .idea, Sample.txt, linear_regression_model.joblib, prac2.ipynb, scaler_object.joblib, and train_data.npy. A banner at the bottom encourages adding a README.

File	Commit	Time
.idea	Initial commit	6 minutes ago
Sample.txt	Initial commit	6 minutes ago
linear_regression_model.joblib	Initial commit	6 minutes ago
prac2.ipynb	Initial commit	6 minutes ago
scaler_object.joblib	Initial commit	6 minutes ago
train_data.npy	Initial commit	6 minutes ago

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/>))

The screenshot shows the MyBinder interface for building and launching a repository. It includes fields for the GitHub repository name or URL, the Git ref (branch, tag, or commit), and the path to a notebook file (optional). The 'launch' button is visible. Below the form, there is a section for copying the URL and sharing the Binder with others. A progress bar at the bottom shows the status: Waiting, Building, and Pushing.

Build and launch a repository

GitHub repository name or URL

Git ref (branch, tag, or commit)

Path to a notebook file (optional)

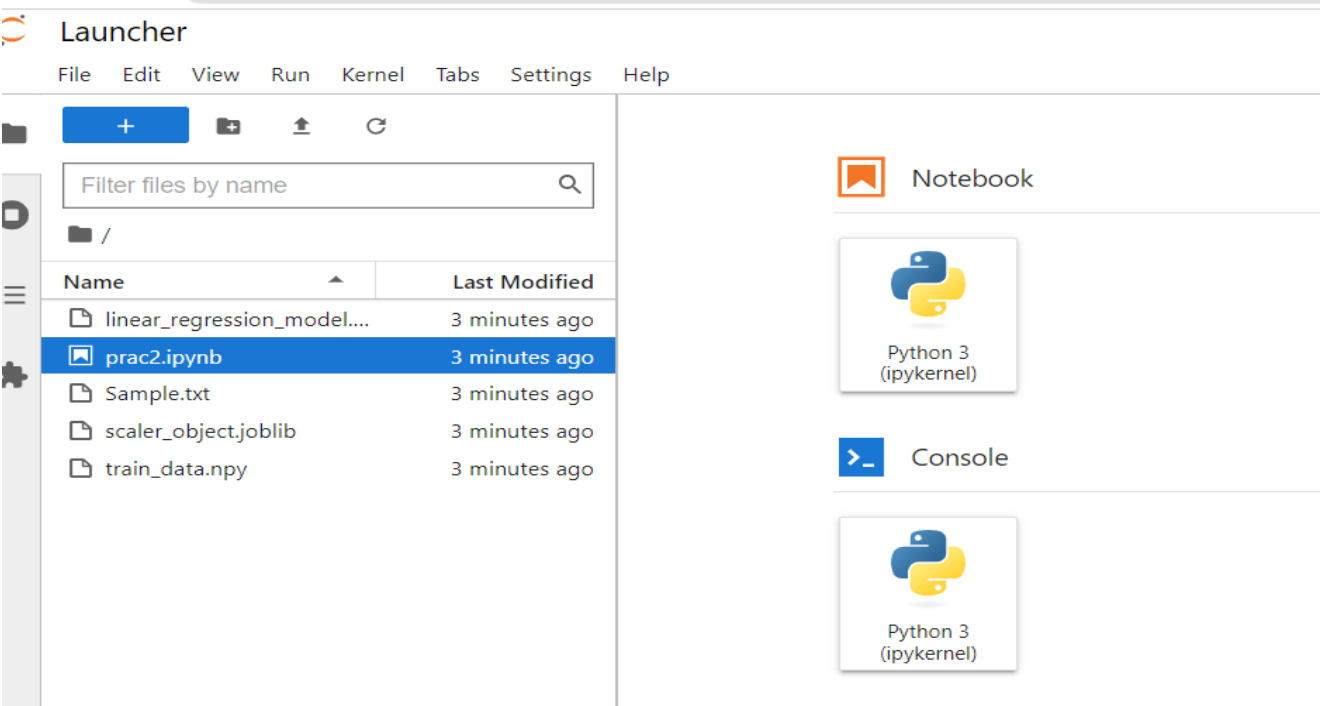
launch

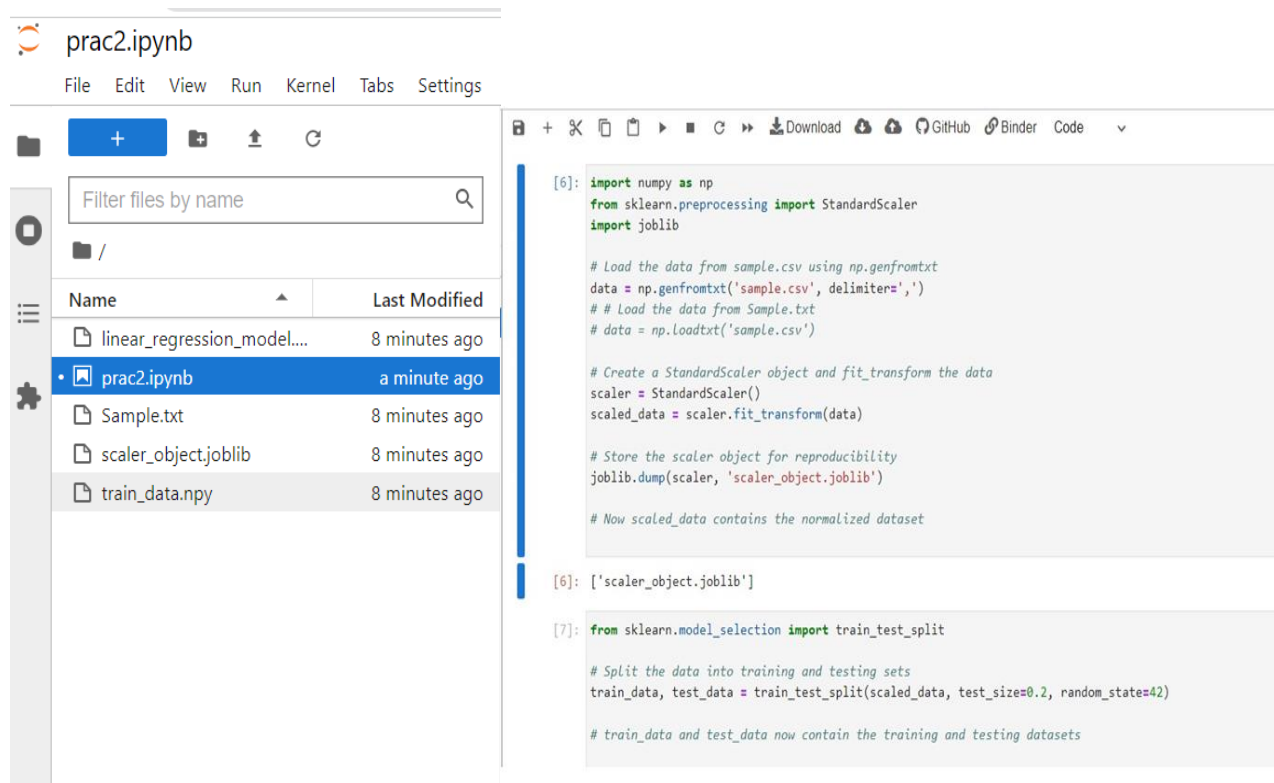
Copy the URL below and share your Binder with others:

https://mybinder.org/v2/gh/Kruti507/MLOPs_main_21012532006/HEAD?labpath=https%3A%2F%2Fgithub.com%2FKruti507%2FMLOPs_main_21012532006%2Fprac2.ipynb

Expand to see the text below, paste it into your README to show a binder badge: [launch](#) [binder](#)

Waiting Building Pushing





The screenshot displays a JupyterLab environment. On the left, a file explorer sidebar shows a directory structure with files: `linear_regression_model...`, `prac2.ipynb` (selected), `Sample.txt`, `scaler_object.joblib`, and `train_data.npy`. The main area contains a code editor with two code cells. The first cell, labeled [6], imports `numpy`, `StandardScaler` from `sklearn.preprocessing`, and `joblib`. It loads data from `sample.csv` using `np.genfromtxt`, creates a `StandardScaler` object, fits it to the data, and saves the scaler object to `scaler_object.joblib`. The second cell, labeled [7], imports `train_test_split` from `sklearn.model_selection` and splits the `scaled_data` into `train_data` and `test_data` with a test size of 0.2 and a random state of 42.

```
[6]: import numpy as np
      from sklearn.preprocessing import StandardScaler
      import joblib

      # Load the data from sample.csv using np.genfromtxt
      data = np.genfromtxt('sample.csv', delimiter=',')
      # Load the data from Sample.txt
      # data = np.loadtxt('sample.csv')

      # Create a StandardScaler object and fit_transform the data
      scaler = StandardScaler()
      scaled_data = scaler.fit_transform(data)

      # Store the scaler object for reproducibility
      joblib.dump(scaler, 'scaler_object.joblib')

      # Now scaled_data contains the normalized dataset

[6]: ['scaler_object.joblib']

[7]: from sklearn.model_selection import train_test_split

      # Split the data into training and testing sets
      train_data, test_data = train_test_split(scaled_data, test_size=0.2, random_state=42)

      # train_data and test_data now contain the training and testing datasets
```