Steps to push a file on GitHub using cmd prompt

cls – Clear the command prmpt screen

dir - directory command is used to check the number of folders in comman prompt.

1. **mkdir** *file_name*- make directory command creates a folder in the directory/path mentioned before.
   (In our case we are making a directory by the name "projects" inside C drive)

2. **cd** *file_name* – goes inside the file and opens it.
   (In our case we are opening the directory "projects" inside C drive)

3. **git clone** *http_link_from_github*– it is used to clone the github repository in the directory/file.

   *To clone a repository it is important to mentioned the http link of the Github repository. To copy the link, one should first go t Github copy the link and paste it in command prompt right after mentioning git clone.

   (In our case we are cloning the directory "Python_Linear_Regression" which we created inside the GitHub) (check fig 2).

   So cloning creates an identical repository folder inside your computer on the mentioned path.

4. Cloning will now create a folder inside the path by the same name as the repository on Github. **So, now in order to upload the files, first copy the files you wish to upload in the folder inside your PC**
   (In our case we are copying the .ipynb jupyter notebook and data_source in the directory Python_Linear_Regression which lies under C drive).

5. **git status** – so git status will tell you what is the current status of the file/directory that you created.
   (For instance, in our case it will show the .ipynb jupyter notebook and data_source are yet to be commited. This is so because we manually copied them in Step 4. Hence, they will appear to be red in colour.)

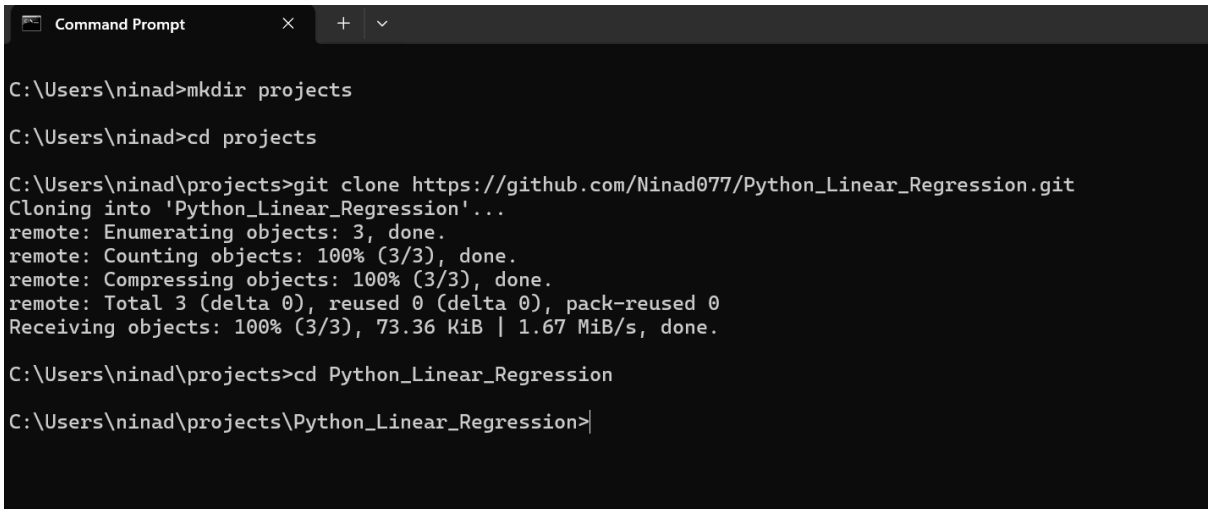
6. **git add . –** so this command will add these non-committed files into the repository. If one checks the git status again, the red-coloured files will be turned to green indicateing they're commited now.

7. Once done check git status again (Repeat Step 5).

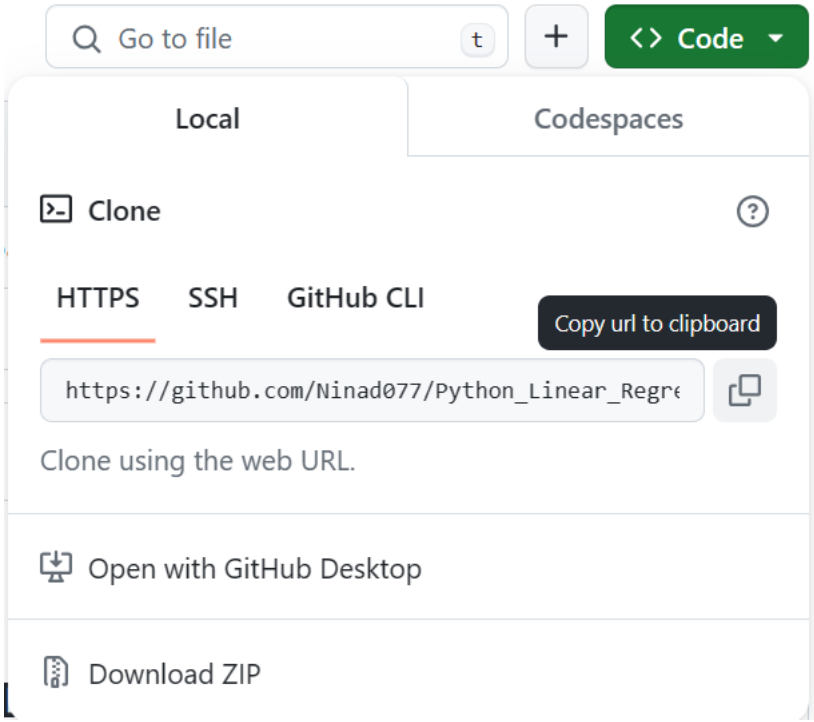
8. **git commit -m "Comment" -** This command is used to add a comment to a commit. In our case we have used the comment as "Files Added"

9. **git push** – Git push is used to push these commited files from the local directory to the GitHub repository. Once pushed using cmd refresh your GitHub to view the files. If git push shows an error try using git push --force.

The process flow is displayed in the following figures:



To copy the HTTPS file:



```
C:\Users\whamd\projects>cd LinerRegression

C:\Users\whamd\projects\LinerRegression>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Session 1.ipynb
        advertising.csv

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\whamd\projects\LinerRegression>git add .
warning: in the working copy of 'Session 1.ipynb', LF will be replaced by CRLF the next time Git touches it

C:\Users\whamd\projects\LinerRegression>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Session 1.ipynb
        new file:   advertising.csv
```

Command Prompt

```
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ninad>cd projects

C:\Users\ninad\projects>cd Python_Linear_Regression

C:\Users\ninad\projects\Python_Linear_Regression>git push
To https://github.com/Ninad077/Python_Linear_Regression.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/Ninad077/Python_Linear_Regression.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

C:\Users\ninad\projects\Python_Linear_Regression>git push --force
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.93 KiB | 197.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Ninad077/Python_Linear_Regression.git
 + f741b7c...fbe1e5e main -> main (forced update)

C:\Users\ninad\projects\Python_Linear_Regression>
```