**Step 1:** github repository (also called as repo)

* Go to github.com and create a new repository for your project
* Link that repo locally to your system so that push can be enabled. Follow below steps on your command prompt.

1. **Clone the Repo Locally**

*git clone https://github.com/brownscholes/Exoplanet-data-analysis.git*

*cd Exoplanet-data-analysis*

What this does: Copies your GitHub repo to your local machine.

Why: So you can work on the project locally, make changes, and push updates back to GitHub.

1. **Create Folder Structure**

*mkdir data notebooks models scripts results #creates folders*

*echo. > README.md #creates .txt files*

*echo. > requirements.txt*

*echo. > .gitignore*

* data/: Holds your dataset(s). Keeps raw and processed data files out of the main directory.
* notebooks/: Jupyter notebooks for EDA, model building, etc.
* models/: Where you'll save trained model files (e.g., .pkl or .h5).
* scripts/: Reusable Python scripts or helper functions.
* results/: Final plots, charts, and output files.
* README.md: Explains what your project does and how to use it. First impression for anyone visiting the repo.
* requirements.txt: Lists all libraries your project depends on (e.g., pandas, scikit-learn, matplotlib), so others can reproduce your work.
* .gitignore: Tells Git which files/folders to ignore when pushing to GitHub (e.g., large datasets, temporary files).

1. **Open .gitignore txt and past below**

*\_\_pycache\_\_/*

*\*.ipynb\_checkpoints*

*.DS\_Store*

*data/*

*models/*

Why: You don’t want to accidentally push large files or OS junk (like .DS\_Store on macOS).

Ignoring data/ and models/ is common to keep your repo lightweight (you can always share datasets via links or instructions).

**Step 2:** Open Jupyter notebook in that folder and start the project