**Note:**

When you write a **code.** in anaconda prompt and enter, it executes a VS code instance.

1. **Set up the GitHub repository. Why it is required?**

When we are working in the team, we need to work in sync. Hence in order to syn the work, we use git and syn the code

1. **Setup new environment**

Before implementing project, it is imp that things are in place & in order to avoid any issue with packages and libraries support later on etc.

1. **Code to setup new environment in visual studio**

***Code:******conda create -p venv python == 3.8 -y***

[Here new environment named **venv** is created and python 3.8 is used, y means it will automatically install and will not ask for approval every time]

1. **Code to activate the environment in visual studio**

**Code: *conda activate venv/***

[Where venv is create environment]

1. New git repository: **go to git and create new repository**

It consists of set of instructions follow & create it

* Initialize **empty git repository** in system folder

**Code**: git init

* **Git add README.md file**

[No need of code. directly creates new README.md file using right click- new file]

We can add any imp statements or note in readme file

* **Syn git repository**

Run following command in terminal:

git remote add origin https://github.com/pratikchothe1/mlprojects.git

To check git execution, enter **git remote -v**

* How to find the sync email account with git

**Code**: git config –global user. Email

* How to push the code to GitHub repository

**Code:** git push -u origin main [or git push -u origin master]

* . gitignore

Create new file in GitHub repository. Create. gitignore file and commit changes

This is how git mapping is done

1. **Create setup.py file and requirements.txt files**

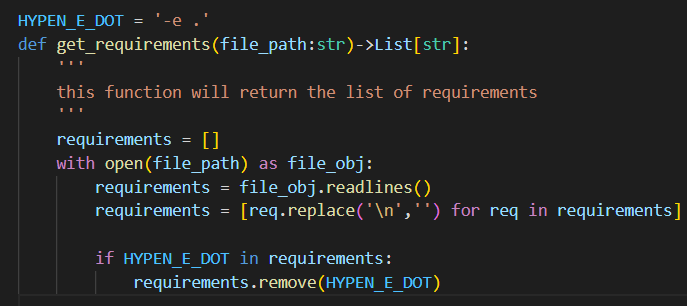
* **Requirement.txt** consist of all the packages that we really need to install while starting projects.
* Setup.py is responsible for creating machine learning application as package

**Note**: In python Pypi, we have lots of packages like seaborn which we can directly install, similarly we can make packages and even deploy it at Pypi for that we require setup.py file.

* When we try to run the requirements.txt file we must able to run the setup.py so for that we put **-e .** in last line of requirements.txt

[-e . automatically triggers setup.py file]

* But in setup.py while actually running the code, -e . should not come in the list we are making. Hence for that that we drop the -e . using logic in function



* Final command to run in terminal

**pip install -r requirements.txt**

[This will install the all the packages installed in requirements.txt file and also will map it to setup.py as we have mentioned -e . in req.txt file]

1. **Mini project structure [setup.py]**
2. **Create a package [Requirements.txt]**