1. **Set up the github**
   1. **new environment**
   2. **setup.py**
   3. **requirements.txt**
2. **Src folder and build the package**

**Create repo in GITHUB:** MLPro

**Open Anaconda prompt:**

cd G:\My Drive\DheepthiReddy\Work\DataScience\DS\_KN\Projects

G:

Code . - (This will launch a VS Code instance)

**VS Code:**

Open new terminal

Access conda prompt from the terminal by selecting command prompt in the terminal window.

**Creating a new env:**

conda create -p venv python==3.8 -y

venv: environment name

python==3.8: version used of python

-y: yes to all the instalments that happen later on

To clear screen:

Cls –

To activate environment:

conda activate venv/

connecting to git:

git init

**create a README.md file in the folders**

git add README.md

git commit -m "README.md file created"

git status

git branch -M main

git remote add origin <https://github.com/Dheepthi-Reddy/MLPro.git>

git push -u origin main

create a .gitignore file in the project and choose the template as python in the github browser dashboard and commit the changes

take a pull in the venv:

git pull

**create setup.py:**

It is responsible in creating the ML project as a package and can be deployed in Pypi

\_\_init\_\_.py: to find the folder as a package

Write in setup.py and reuirements.txt

pip install -r requirements.txt

**mlproject.egg-info** folder is created after the command was run

1. create **components** folder in src and a init.py file to it:components folder is for all the modules that are to be created in it.
   1. **data\_ingestion.py** file is in components folder: code to read the data, like separating train data and test data
   2. **data\_ingestion.py** file is in components folder: code to read the data, like separating train data and test data
   3. **data\_transformation.py** file in components folder: code related to transformation like changing categorical features to numerical features, handling one-hot encoding and…
   4. **model\_trainer.py** file in components folder: this file to train the model and kinds of the model used and to call confusion metrics to solve classification problem….
2. Create **pipeline** folder in src: kind of pipeline that needs to be created like, training and prediction pipelines.
   1. **train\_pipeline.py** file is created in pipeline folder: has code for all the training pipeline and from this pipeline we try to call all the components of components folder
   2. **predict\_pipeline.py** file is created in pipeline folder:
3. **exception.py**, **logger.py**, **utils.py**(common functionality) files in src folder.
4. Execute logger.py in terminal:

python src/logger.py

a new log folder is created with a .log file in the format we created