End to End Machine Learning project

## Tutorial 1: GitHub & Code setup

* Steup a new environment
* Setup.py
* Requirements.txt
* Steps:

1. Create a new repository “mlproject”
2. Create a new folder in which entire project will be developed
3. Copy the path of the folder & open anaconda prompt
4. Code . this will open vscode
5. First we need to make sure that we are in sync with the GitHub repository
6. Open new terminal in vs code and create new environment

(select cmd in terminal and use below command)

conda create -p venv python==3.8 -y

activate the virtual environment

conda activate venv/

1. Clone the repository and sync with GitHub

* Initialize git repository (git init)
* Add README.md file, before adding create README.md file (either in git or in vs code) (in README.md file we can write descriptions, what are all the steps required)
* Add README.md file in GitHub repository

git add README.md

commit the file,

git commit -m "First commit"

* Push the file to git repository

git branch – M main

* Add origin (so it is in sync with the GitHub repository)

git remote add origin <https://github.com/AD/mlproject.git>

git remote – v

to push the data into github repo we need to set git global

(check git global config and add username and email of your own)

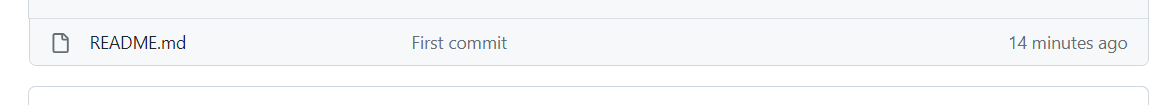
git config --global user.name "John Doe"

git config --global user.email johndoe@example.com

* Push the data

git push -u origin main

by this now we can see the file in GitHub repository.



* Create a new .gitignore file in GitHub repository, select python and commit changes

git pull   
  
so all the updation will be done in vscode as well.

* Setup a setup.py and requirements.txt

(setup.py will be responsible for making or ML application as a package, which can be used further.)

from setuptools import find\_packages , setup #find all the packages

setup (

    name = "mlproject",

    version = "0.0.1",

    author= "yourname",

    author\_email= "youremail id",

    packages=find\_packages()

    install\_requires = ['pandas', 'numpy', 'seaborn '] #automatic installation can be done

)

* Create a new folder “src” in vs code (source) and inside it create \_\_init\_\_.py
* When in setup.py find\_packages() is running it will go and find in how many files have \_\_init\_\_.py running. Considering source as a package , it will try to build it.

1. We create a function , when we need to install many packages,

get\_requirements () #which will take requirements.txt it should be able to read all those files.

def get\_requirements (file\_path:str)-> List[str]:

    '''

    this function will return the list of requirements

    '''

    requirements = []

    with open (file\_path) as file\_obj:

        requirements = file\_obj.readlines()  #the line will get readed, but \n will get added

        requirements = [ req.replace("\n","") for req in requirements.txt]  #replace \n with blank

A screenshot of a computer

Description automatically generated

Here -e. will automatically trigger setup.py

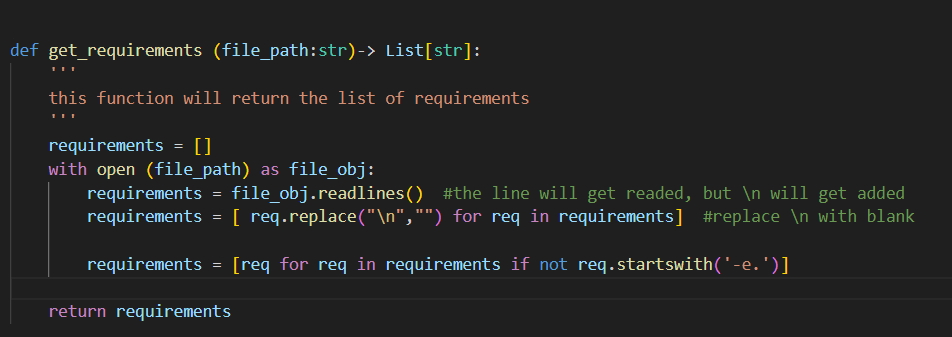
When we install requirements.txt the setup.py will also run.

A black background with white text

Description automatically generated

To solve the above problem use :

pip install -r requirements.txt



A screen shot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

Build is happening,

Mlproject.egg-info will tell us the packages which are getting installed.

We have also created a source folder, and build the packages.