Some Important point related to pipe line

Code 1 # to create a new environment
venv

>>> conda create -p venv python=3.8
Press enter
>>> y

Note - -p will create environment in local directory
-n " " " " by default dir.

Code 2 # to activate that new environment

i) go to the venv (environment) file copy the path and

ii) conda activate (paste the path here)

Code 3 # Now we have to configure our git.

i) Now we want to make a repo do github direct link with VS code

ii) for check make to file one is -
→ .gitignore
→ README.md

Note → git push origin main -f.

iii) Now use terminal :-

i) git remote -v (to check whether our VS code folder is connect to the git or not)

ii) Check the code of github copy & paste
→ git init
→ git add .
→ to remove all the added thing (git reset)
→ git commit -m "abcd"
→ git branch -M main

1 Inside Machine learning Deployment there is -

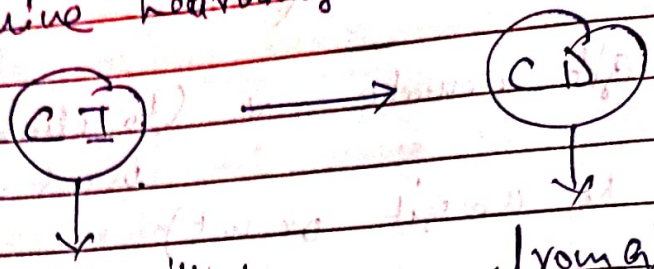
→ CI (Continuous Integration) mean if three people writing code then they are uploading a in a central repository (Github)

{CI → CT → CD}

→ CD (continuous Deployment)

neuro and

2 Machine learning Development



from local to github — from github to cloud

i) Model Module Packaging
ii) Docker

ML-Ops → Development → CT → Ops

3 If we want to consider a file as a package then Simply create __init__.py

4 To create package and install in the own project or environment

i) first create __init__.py file that we want to make package & install

ii) Next make setup.py file in project folder.

iii) write code in the setup.py file-

iii) Code in setup.py file is -

```
>>> from setuptools import setup, find_packages

>>> setup (name = "census-income",
           version = "0.0.1",
           author = "Sumay",
           author_email = "sumaykumar369@gmail.com",
           packages = find_packages (),
(# important statements.) install_requires = ["pandas", "numpy",
                                                "flask"]
           )
```

∴ after this go to terminal & write

```
>>> python setup.py install (write in terminal)
```

Note :- (-e .) (write it in requirement.txt before using setuptools or setup.py)
(-e .) (-e space dot)

Then use pip list & you will able to see

How to make changes directly to the github write code in terminal

>>> git add .

>>> git commit -m "add file"

>>> git push -u origin main

Q In dataset if there is Pincode column what to considered it Categorical or Numerical a feature and what to apply on it.

Ans → Pincode will considered as a Categorical feature

→ we will apply Target guided Ordinal Encoding

→ we will perform grouping of the pincode and we find mean of the dependent feature

Date
Page

Data Ingestion → Data transformation

① Data Ingestion (to divide data into training, testing, validation)

↓

Data validation

↓

Model training

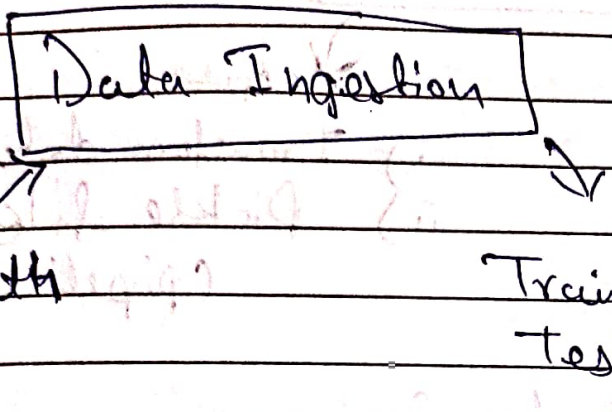
↓

Model Evaluation

↓

Model Deployment

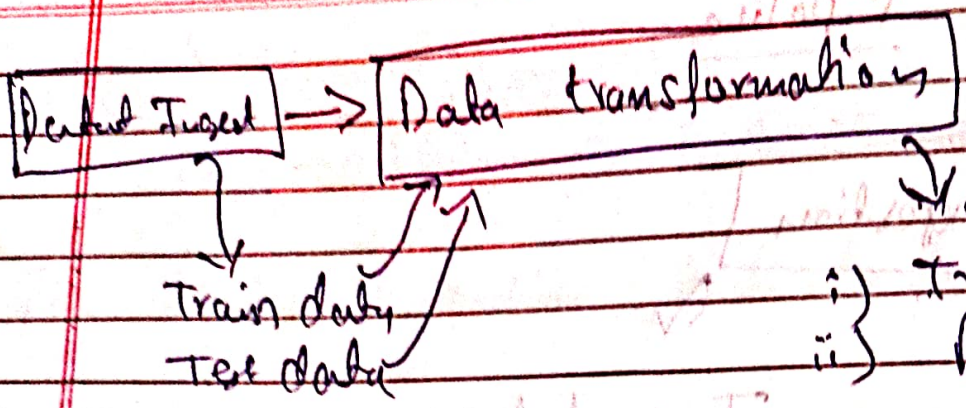
Pipeline



∴ this path should to be store in some class that why we have to create some seperate class and no other functionality that in Initialise that Data ingestion Configuration.

Note :- artifacts folder is create to store all the file like pkl, train, test

Date
Page



Data Transformation Todo is —

Feature engineering
- Feature Scaling
- Handling Missing value
- Handling categorical
- Handling numerical feature

Q What is artifact file?

Ans Artifact file generate output and source code in artifact folder.

Ans The setup.py is responsible in creating ML application as a package. We have to install this project.

With the help of setup.py we will build our entire ML application as a package and deploye in Py and from there anybody can do the installation and anybody can use it

Q Why we create __init__.py in every folder?

Ans To consider a folder a package itself. It will also help in to import