**End-to-End ML Project with CI/CD Pipeline**

Pipelines using 🡪 Github Actions

Concepts that are covered:

1. Data Ingestion
2. Data transformation
3. Data Evaluation
4. Model training
5. Model Evaluation
6. Model Deployment

Deployment on AWS S3.

Day 1: 25-07-2024 14:19

Agenda:

1. Set up the GitHub {Repository}

Set up an environment

Setup.py

Requirements.txt

Learnings:

1. How to change the terminal path from c to d:

**Cd /d [path]**

Syntax to change the working directory to another drive (for example from drive C: to drive D:).--> cd /d [path] 🡪 cd /d D:\project-datascience

1. Setting up environment
2. Using pip install virtualenv 🡪 virtualenv [directory\_name]
3. Using python –m venv [directory\_name]
4. Creating a GitHub repo initializing it to the project and trying out Push, pull. Created .gitignore file to ignore venv, libs, and extra.
5. Created setup.py 🡪 this helps in wrapping up our application as a package, later anyone can use it by the pip install command. We can also deploy it in **PYPI PYTHON PAGE**
6. In setup.py we created a function that will install all the packages listed in the requirements.txt file
7. In requirements.txt we have mentioned ‘-e .’ this will provide a link to the setup.py, so whenever we run the install –r requirements.py command, it will go to setup.py and run it. Leading to the creation of the info file. This whole step is an outline of a project. The main project resides in the src folder

Day 2: 30-07-2024 10:44

Agenda:

1. Creating logging and exception-handling files
2. Defining a project structure

Created a folder components and added a \_\_init\_\_.py file so that we can use it later.

In components, we do create a few files for certain tasks:

1. Data Ingestion: Reading of data from a specific location 🡪 Databases, Files or folders, etc.
2. Data transformation/validation: To transform the ingested/read data
3. Model train: After transformation, we train the model and analyze model performance using various metrics

We create logging, utils, and exception-handling files for our project

We start with the sys library: It has information of all erros, has control over python runtime envn