1. **What is Pipelining? How can it be implemented in Python?**

Pipelining is used to chain multiple estimators into one. This is done to automate the machine learning process. This is beneficial in data processing. Some codes will transform features such as Normalize numerical values or turn text into vectors, or fill up missing data. They are called ***transformers.*** Other codes will predict variables by fitting an algorithm such as a random forest or support vector machine (SVM), they are called ***estimators***. So, in a pipeline, we first sequentially apply a list of transformers (data modelling) and then a final estimator (ML model). In short, pipelines are set up with the ***fit/transform/predict*** functionality.

Transformers	**Estimators**
Must implement **fit ()** and **transform ().**	Should implement **fit ()** and **predict ().** The estimator must implement fit () however, not necessarily implement predict.

2. **What is the difference between a research project and production project in ML? What are problems that need to be mitigated in production models using pipelining?**

Research project	Production project
A research project is a scientific endeavor to answer a research question.	Project transforms inputs into outputs.

Workflow management system (WMS):

WMS is needed to move and transform data.

Airflow:

A key difference is that airflow pipelines are defined as code and that tasks are instantiated dynamically. In Airflow, the workflow is defined as a collection of tasks with directional dependencies. It is similar to a directed acyclic graph (DAG). Each node in the graph is a task, and edges define dependencies among the tasks. The main components of Airflow are:

- Metadata Database - stores the state of tasks and workflows
- Scheduler - uses the DAGs definitions, and tasks in the metadata DB, and decides what needs to be executed

- Executor – A message queuing process which decides which worker will execute each task.

Airflow follows *“Set it and forget it”* approach which means once a DAG is set, the scheduler will automatically schedule it to run according to the specified scheduling interval. To understand this concept better, airflow is compared with Luigi.

Luigi:

Luigi is a python package to build complex pipelines. The main components of Luigi are **Tasks** and **Targets**. A target is a file which is outputted by a task whereas a task performs all computation works and consumes the targets generated by other tasks which will serve as input for this task.