Monthly updates on the application.

Datasets: background using various sources.

Is it possible to automate each part in the server?

Create small versions of the application

Basic: appearance – divide the program into the front end and the back end.

<https://neptune.ai/blog/ways-ml-teams-use-ci-cd-in-production#:~:text=Continuous%20integration%20(CI)%20is%20the,(to%20build%20the%20application)>.

Backend: run preliminary work before the machine learning program.

Design an automated system: run a sequence of instructions.

Predictor: job

Each job is a workflow with a configured engine to run the workflows.

Input: machine, types, and outputs.

Executable: the directory to run the application.

Example:

Job name and runnable determines whether it is possible to run.

Custom data validation methods, there is drop-down.

An administrator can write down the drop-down items.

Set the tasks into order and save the tasks.

**Is there going to be overwrite conflicts between the standard outputs and the outputs?**

If there are some commands, files outputs are written on system, standard output, and standard error. Outputs are sent to standard out and written to others e.g. png.

Standard output -> file

Out glob: check whether the file has been examined.

<https://www.commonwl.org/user_guide/topics/yaml-guide.html>

To save work from writing a workflow language

User configures and determines the benchmarks for the machine learning algorithm

Gain performance statistics on the changes

Record the previous running statistics to make decisions.

Writing own or pre-existing data storage systems.

Example input: mnist, consider the scale of the project for the engineering

Example output: performance number from the classification.