Here is what I would see as core functionality;

1. Ability for user to configure ML training pipelines (steps of code that train a Learner) ☑

   Source:https://towardsdatascience.com/from-ml-model-to-ml-pipeline-9f95c32c6512

   Create pipeline as set of pre-processing steps and a model.

2. Ability to monitor one or more code repositories for changes (i.e., GitHub, GitLab, etc...) ☑
3. Ability to monitor a data repository for changes to datasets
4. Ability to run an ML training pipeline in response to changes in either code or datasets
5. Ability to store/compare ML results with previous run
6. Ability to trigger automated actions-based results of training

This is obviously a minimum description of functionality, and each step can be made more or less sophisticated.

For each step, some options might be;

1. Configuration
   a. Required: Configuration by a text document (i.e., yaml as per like CWL)
   b. Optional: Configuration by code (i.e., configuration by DSL like either the chef or puppet systems)
   c. Optional: Configuration by GUI
2. Code monitor
   a. Required: Ability to watch code in a single type of code repository
   b. Optional: Ability to watch code across many types of code repo
3. Data monitor
   a. Required: Ability to watch data changes in a data repository
   b. Optional: Ability to watch data changes across a range of different data stores
4. Running Pipeline
   a. Required: Ability to run ML training pipeline on a single machine
   b. Optional: Ability to scale running ML training across many machines
   c. Optional: Ability to scale and run ML training across different processors (CPU, GPU, tensor cores....)
5. Results comparison
   a. Required: Collate the results as configured in a way that can viewed by the user/admin
   b. Optional: Compare results with prior runs of the ML training pipeline
   c. Optional: store results/comparisons (i.e., in a database)
   d. Optional: display results/comparisons in GUI
6. Automated actions
   a. Required: Ability to email results/comparisons to user/admin

b. Optional: Ability to trigger other server/code actions (e.g., CI processes like Jenkins)

So, this is how I would view the system and some things I believe are required. You may have a different view. And I encourage you to think about this and make your own decisions about what you feel is an important part of the system. For instance, in point 1 I have listed that text-based configuration (something like CWL) would be a good core piece of functionality. You may feel that configuration by DSL is better as it is more versatile. I would suggest GUI configuration is a poor idea to start with because it can always be added later as a wrapper to a text or DSL system.

And so, this is with the other points. It is up to you to work out what is required for each main piece of functionality and for you to decided what is the required minimum and what would be good optional extras.