Establishing the goals of your project

What is your project fundamentally about?

Recently, there have been an increase in the machine learning development. When building machine learning projects, the common challenge is overfitting or underfitting the model. In our project, the goal is to automate the building, testing, and the deploying of the new ML code [1]. Through the automation, the project aims to create an iterative process to improve the existing machine learning model. Consequently, through the programs, a working machine learning code will be built.

What are you intending to design/build/investigate?

The target is to build a machine learning continuous integration system. While there are existing systems to automate tests on repository code, there remains plenty of areas of development for the machine learning coding aspects.

The project will be divided into the training and the data sections. Based on the training and the testing data, the program automates error analysis, which allows the program to continue the iterations. Using the tests, the project aims to simplify the process to validate the tests and ensure that the product transitions to the deployment stage within a shorter timeframe.

Figure 5 provides a visual representation for the inputs. The inputs will be divided into two sub sections: the code (ML training and benchmark) and the datasets to train the ML algorithms.

What do you intend to deliver as the project results?

Figure 4 provides an overview of the project diagram, in which we allow connections with the existing source control systems. Using the figure, I aim to deliver an interactive training program on ML code. Using each test result and the performance statistics, we aim to retrain the model through automation.

In ML testing, compare the previous and the current test accuracies to determine which version of code to become the updated version. Applying the methods help to consolidate knowledge of the cost of testing and apply the tests on different data sample sizes.

Use the difference between the test set results and the expected results to calculate the test score. Test score will help to compare the machine learning model with the expected result. After a series of continuous iteration testing the code and assess the model with the desired model, the test score will be expected to converge.

What would constitute, in your own and your supervisor’s eyes, a 100% satisfactory solution?

Resolve commit conflicts made by different users as show in Figure 1.

Front-end

Objective	Stage
Provide different interfaces with each functionality.	⏳

Back-end

Objective	Stage
Enables the ML code to run different versions of local code.	⏳
Develop version control and suitably handle the commit conflicts.	⏳
Provide a platform to build and run the code.	⏳
Merge function runs tests on the training and the datasets.	⏳
Using the merge results, generate performance statistics.	⏳
Include external servers to run and test the program in different environment. [Figure 3]	⏳
Create an iterative process to train and benchmark the ML code.	⏳
Create a benchmark. The benchmark will provide a report on the repository stats e.g., accuracy and sensitivity.	⏳
Calculate a prediction accuracy score on the training and testing datasets.	⏳

Data Control

Objective	Stage
For large datasets, conduct research into different data version controls and select a suitable method to process the datasets.	⏳
Allowing the user to determining the training and validation sample size.	⏳

Testing

Objective	Stage
Set up a test system to compare the new model with the existing model using a common test score.	⏳

In the worst case what is the minimum that needs to be completed to achieve a pass?

Planning

Objective	Stage
Build the diagram outlining the program structure.	⏳

Front-end

Objective	Stage
Create a web interface for the continuous integration platform	⏳

Back-end

Objective	Stage
Provide unit testing for each functionality	⏳
Split the data into training and testing groups.	⏳
Present the benchmark statistics in a visually aesthetic format.	⏳
For each failed build and test, identify the bugs for the users.	⏳
Provide an outcome for each training process.	⏳
Identify and select a version control system to ensure that the program is up to date.	⏳
Connect the ML-CI tool with an existing source control (e.g., GitHub or Jenkins)	⏳
I can successfully run the back-end code using the command prompt platform.	⏳
Integrate the system with an existing ML service.	⏳

What are your personal aims that you hope to achieve?

1. Improve and demonstrate Python skills at creating the coded solutions. 🎯
2. Automated testing aims to improve the understanding of the software development cycle [Figure 2]. Within the cycle, improve the training for the testing stage and identify the differences between the traditional software testing with testing ML software. 🎯
3. Learn and apply the ML development cycle into the current CI system. 🎯
4. Demonstrate building machine learning to minimize the errors from overfitting. 🎯
5. Improve upon the existing automated testing strategies and improving automated training and benchmarking mechanisms for the ML system. 🎯
6. Apply my research skill into planning and coding the project. 🎯
7. Make some progress within the CI development for the machine learning type of code. 🎯

Figures

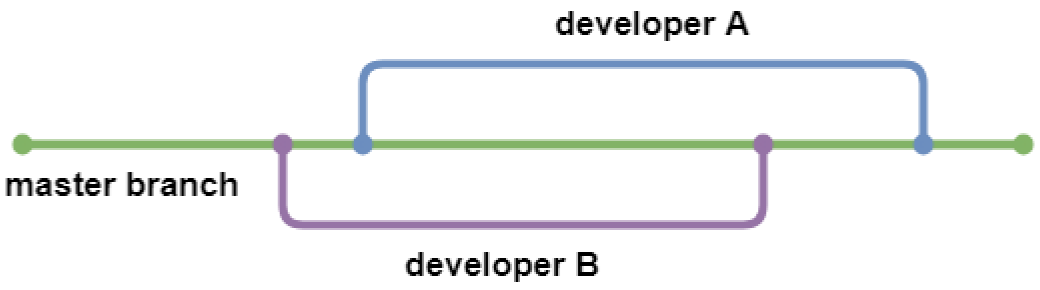

Figure 1

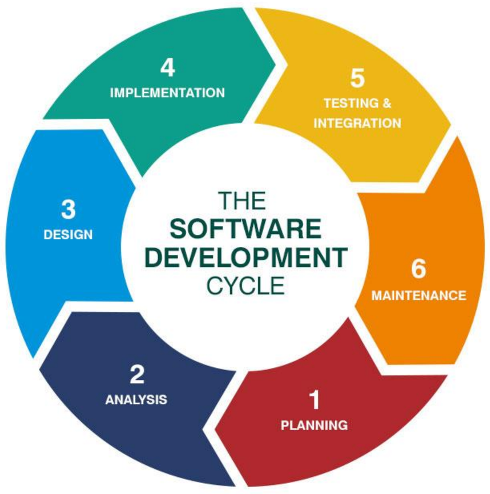

Figure 2

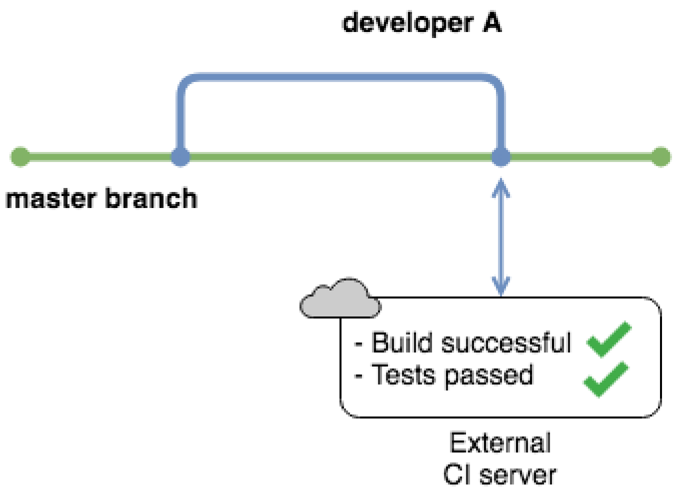

Figure 3

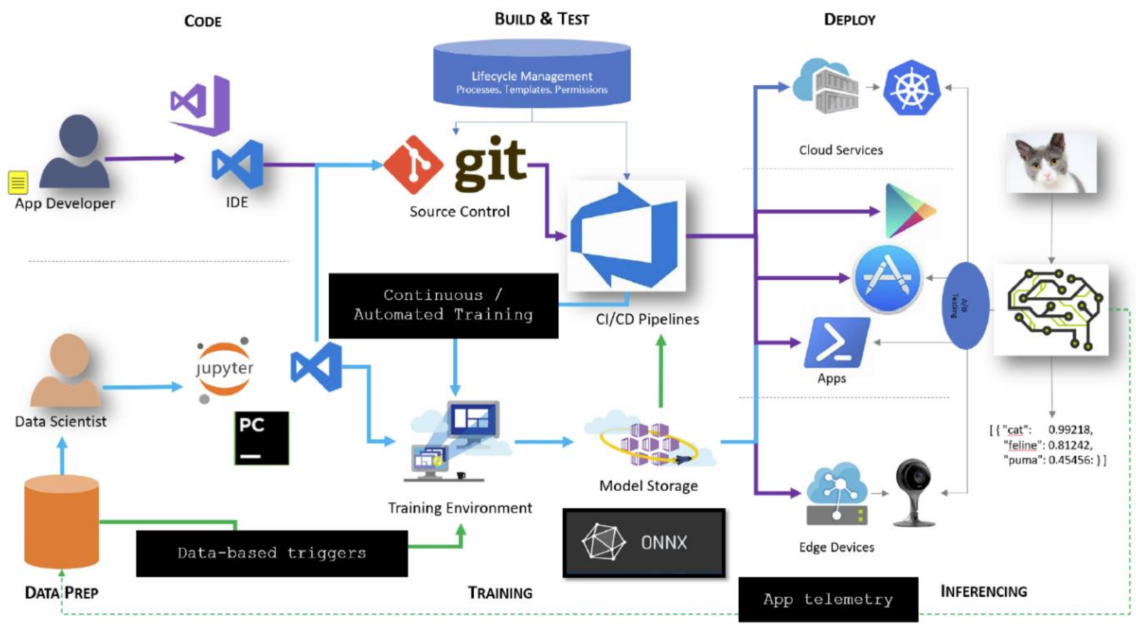

Figure 4

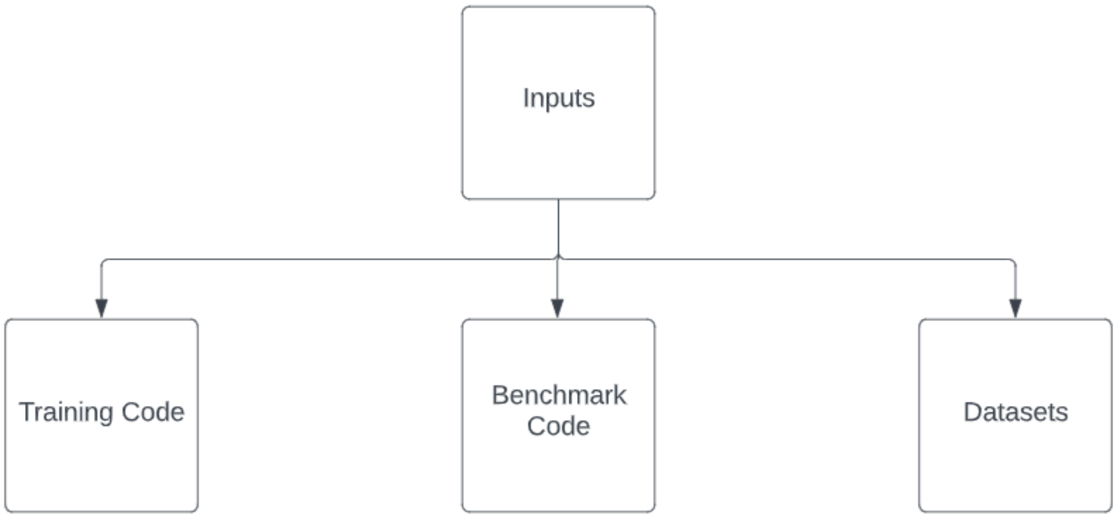

Figure 5

Sources

1. Karlaš, Bojan, et al. "Building continuous integration services for machine learning." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020.
2. Danglot, Benjamin, et al. "An approach and benchmark to detect behavioral changes of commits in continuous integration." Empirical Software Engineering 25.4 (2020): 2379-2415.