1. Project Description

My project aims to automate the testing to help to build new models. The project seeks to use a website platform where you can enter data and make predictions. Using the website, the user can check whether the code is correct. Whenever the user makes new changes to the data, the program will reset the machine learning development cycle by rebuilding the application. Each build will train and test the model and determine whether the tests passed. A continuous integration system is a sandbox environment for testing the model before moving the program onto the deployment stage. There are many areas for the ML continuous integration system: to improve the models in the financial forecast sector.

Context: current project web servers run several ML methods. We need to create a server to automate the running of the ML methods.
The project breaks down into several key concepts:

1. ML Life Cycle: highlights the development cycle for improving an ML model. The developer will use the training and testing ML data files for evaluating the model. Understanding the ML life cycle helps with enhancing the model validation process.
2. ML Continuous Integration system: provides a platform to take the ML Code and datasets to run automated training. If the tests pass, the platform will execute user-defined downstream processes.

2. Goals

The project focuses on validating the models with an external source control system (GitHub Actions). After the validation, a user can deploy the model on external sources,e.g., a website, app store or cloud service.

1. Review and configure the ML pipelines for training.
2. Integrate GitHub to monitor the code changes.
3. Implement a tracking method on the datasets.
4. Any changes to the code or dataset will run the ML training pipeline.
5. Optimize for the best ML model by comparing new scores with historical ML models' scores.
6. Run automated steps (such as sending an email or pushing code to servers) when ML benchmarks pass or fail targets.

3. Background

Continuous integration requires testing data as an essential component. In traditional models, the program feedback on the test accuracy scores to the developers to improve the model. Testing scores may cause overfitting because they may mislead the user towards a skewed result. A possible solution is to allow the CI system to select independent testing data for each training process. In my project, the developer will only received the pass or fail statements as feedback. CI system reruns whenever there's an edit. Tracking the changes in the ML code and datasets can help the customer identify the point to roll back when making a mistake during the development process. A developer should focus on the testing and release stages of the ML development cycle for designing the ML continuous integration system.