Dana Conley

CS 333 Dev Ops

April 25, 2023

Final Project Design Document

Existing Functionality & Technologies Used

The existing project is focused on machine learning using decision trees. The project's current functionalities include data training and testing through calculating the total entropy and information gain for each feature, then choosing the feature with the highest information gain. The tree is updated in the format [left, right, feature], where the feature assessed is added, and the terminating leaf is added in either the left or right spot, accordingly. In the spot that isn't filled with a value, the function is then called recursively, passing in the new dataset. This repeats until the tree is complete and all samples have a designated location, or until the max depth is reached or information gain is equal to zero. This project is built using Python, and the data being trained is stored as NumPy arrays.

Plan for Unit Test Coverage

I plan to provide unit test coverage of at least 75% by testing each of the following functionalities: building the numpy array, building a list from the array, building a dictionary, training the data, testing the data, making a prediction for a data sample's label, building the decision tree, building a random forest, and testing the random forest. Each of these functionalities take place in the code, and therefore I'll have unit tests to cover all of them, which will provide adequate function coverage as well as overall code coverage. Additionally, I will utilize different unit tests within each function to test the different branches and edge cases. I'll ensure that each statement is tested and that any boolean sub expressions are tested to account for condition coverage. I'll measure test coverage using coverage.py, since my code is based in Python. By using the coverage report, I will be able to keep track of code, statement, and branch coverage.

I also plan to use the Hybrid testing method to implement integration tests. I'll use integration tests to cover how these functionalities work with each other as well as the project's I/O. For instance, I can test how the different data storage types work together (arrays, lists, and dictionaries), how the data storage types are passed into the decision tree training, the connection between training the data and building the tree, how a prediction works with a trained data sample, and the connection between building a decision tree and a random forest.

Plan for Centralizing Source Code

I plan on using Git and Jenkins to centralize my source code and automate the building and testing of the code. I am already familiar with Git, so it is an obvious choice for me to use familiar technology for the execution of my tests. Although I'm not directly familiar with Jenkins, I have done research on it as well as other technologies, and I've found that it is open-source and should be fairly easy to learn its functionalities. Jenkins can easily pull from a GitHub repository, which will make building and testing quick and efficient.

Plan for Automating Build

I plan on automating the build and deployment of my finished software using Git and Docker. This will be useful because Docker Hub will automatically build from the source code and push to a Docker repository. I am choosing to use these technologies because I am already familiar with Git and somewhat familiar with Docker. Plus, I have found several resources that walk new users through setting up a Docker repository to automate builds.