Andrew ID: csagbo
Name : Carmel Prosper SAGBO
Github: labayifa

Mini 2 Lab 3
April 16, 2024

This lab was about building a Dashboard for our Machine Learning Model:

- I implemented a dashboard to visualize the Iris dataset.
- The dashboard is a web application that allows users to interact with the data.
- The dashboard is built using the Dash library in Python.
- The dashboard has the following features:
    - Explore Iris training data
        + Load the Iris dataset
        + Upload a new dataset
        + Visualize the dataset with different plots (Histogram, Scatterplot)
        + Dataset overview in a table
    - Build the model and perform training. Here we can
        + Build a new model by selecting a given uploaded dataset through it's ID
        + Retrain an existing model using its ID and a given dataset ID.
    - Score model:
        + We score a row of data which is a set of entries for which we classify the species of Iris.
    - Test Iris data
        + We can test the model with a given dataset ID and get the accuracy of the model.

1. What did you find easy to do in Dash:
    + Dash helps me to integrate all my API resources easily into a dashboard that we can use for a quick presentation. I found it interesting the way we can bind an input and the output result to a function for a bidirectional communication between the model and the dashboard with just a simple function.
2. Likewise, what was hard to implement or you didn’t wind up getting it to work?
    + The issue I faced was not knowing that the order for the *Input* matters while we were writing the callback annotations.
    + Apart from that I started by using *Input* for both the buttons and fields and this gave me some bad behaviors. I went through dash documentation and realized that I can use *State* to hold the "Text fields".
3. What other components, or what revised dashboard design, would you suggest to better assist in explaining the behavior of the Iris model to a client?

- + I think that there is more we can do with Dash, but this is still limited if have to include some more specific components and interactions for more flexible usage and access controls.

4. Can you think of better ways to link the “back end” Iris model and its results with the front-end Dash functions?

- + We can build a custom set client performing most of the operation and bind it simply to our Dash but for an improved performance we can completely shift from Dash and use other Frontend tools like React, VueJs or Angular.

The whole work is in the "**submission**" folder with a "**README.md**" instruction on each file also hosted on GitHub at : https://github.com/labayifa/ai_design_mini_2_s24.git