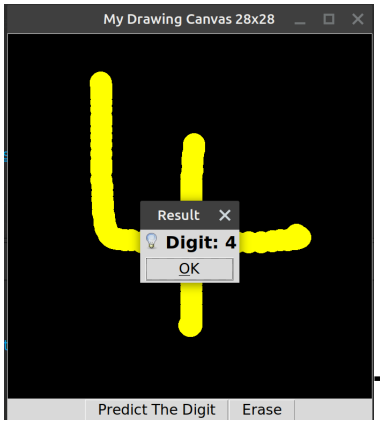
DA5402: Assignment #3

We have learned about REST APIs, which has become a popular vehicle for exposing ML functionalities crafted in a Python environment. Let's build our first AI application which would allow an user to draw a number using the mouse pointer on an UI canvas, followed by recognising the number using an underlying ML classifier model.

There is an application available already at https://github.com/sudarsun/HandwrittenDigitClassifier. The underlying model of this application is not great (a simple MLP) despite the accuracy level on the testing data is ~90%. The apparent performance is ~60% when you draw the digit on the UI canvas. Let's not worry about the correctness for now.



Task (50 points)

If you look at the code base, you would observe that the model part and the UI part are tightly coupled. Basically, the modeling piece is being invoked as a native library call from the UI application. The objective of the task is to make them loosely coupled. Essentially, the modeling piece, i.e., the prediction functionality has to become an REST API end point *(30 points)*. And the UI part should invoke the REST API call to get and display the predicted digit *(20 points)*.

You may choose to send the 28 x 28 gray scale image directly over the REST API (POST) method by putting the data in the message body. Or you may send the flattened 1-d normalized vector (each element will be in the 0.0-1.0 range) as a binary data to the API endpoint. If you choose the former, the API endpoint has to flatten the image, followed by normalizing the vector before giving it for model inference.