

DSL 810 (Data Driven Design)

Instructor: Dr Jay Dhariwal, Dept of Design, IIT Delhi

This handout describes how to deploy your machine learning models into apps.

Case 1: Using own models or pre-trained models (which don't require any libraries)

You have learnt to go through the data driven design process to come up with your own machine learning models which could be used for predictions.

For example:

1. Linear Regression models:

$$y = f(x),$$
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots$$

Predict \hat{y} if x_1, x_2, x_3, \dots are given

Please refer to the regression session video and the handout pdf for more details.

2. Logistic Regression models:

Likewise, we can estimate the probability of classifying the data into one class vs the another based on the sigmoid function:

$$\hat{p} = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots)}}$$

For binary classification, if $p < 0.5$ then the data point belongs to one class else the other. Please refer to the logistic regression session video and the handout pdf for more details.

In these cases, one could use any **web app** or **mobile app** which can take inputs from the user, apply the function and output the prediction.

We have already discussed about html+css+javascript for making web apps. Let's take an example using [Javascript HTML DOM](#) for input and use functions to process the input to produce output.

Let's build a Mobile App in **MIT App Inventor** to achieve the same.

Lets walk through the basics of making a mobile app by walking through the Buzzing Bee and Text to Speech examples.

Next we would create input text boxes for collecting data from the user and process that data using regression/classification models to share the output with the users. That would be MachineLearning1.aia file to view the UI and the code on MIT App Inventor platform and MachineLearning1.apk file to run it on your Android Mobile.

Case 2: Deploying models created in Google Colab in Web Apps and Mobile Apps

We are very used to carrying out the data driven design process on a data science platform such as Google Colab. How do we deploy machine learning models created on this platform to the Web Apps or Mobile Apps?

Flask (a Python based web app framework) helps us in connecting the front end (Web App/Mobile App) to the back end (a Google Colab notebook) with the help of the Python library flask-ngrok.

Step1: Let's run the most basic version of this by connecting a Google Colab notebook to a public URL (<http://xxxxxxx.ngrok.io>) using Flask.

File: *Flask1.ipynb*

This [link](#) explains the meaning of different syntax in the code in *Flask1.ipynb* file.

Step2: Let's now connect a Google Colab notebook to a formatted html page on the google drive folder using Flask. Please create a templates folder in the google drive under "Colab Notebooks" folder and in this folder, add the index.html file.

File: *Flask_html.ipynb*

This [link](#) explains the meaning of different syntax in the code in *Flask_html.ipynb* file.

Step3: We now take the next step of connecting a mathematical equation created in Google Colab to take inputs for the predictors from an html page "index_.html" and run this equation on Colab notebook using these inputs and show the output on another html page "predict2.html".

File: *Flask_input_and_output.ipynb*

Step4: We are now ready to take the next step of connecting a machine learning model (linear regression model) created in Google Colab to take inputs for the predictors from an html page “index_.html” and run a model on Colab using these inputs and show the output on another html page “predict2.html”.

The syntax used on index_.html page can be understood by referring to HTML forms on https://www.w3schools.com/html/html_forms.asp

File: *Flask_regression.ipynb*

This [link](#) explains the meaning of different syntax in the code in *Flask_regression.ipynb* file.

Step5: Likewise, this framework could be used for classification models or other kinds of models as well. Logistic Regression based classification is implemented in the below file.

File: *Flask_classification.ipynb*

Step6: Deploying a machine learning model on a Mobile App

When the Colab notebook *mitappinventor_regression.ipynb* is run, then to view the result on the webpage, please add <http://xxxxxx.ngrok.io/1/10/75> to test if the prediction is seen.

Make sure to change the xxxxx.ngrok.io link in the MIT App Inventor .aia file to run this regression model from the Mobile App.

Files: *mitappinventor_regression.ipynb*, *flask_api_1.aia*

Likewise, other models from Colab notebook could be run using a Mobile App.

Use of slider for input.

In general, the graphically appealing visualizations of inputs and output are easier to understand.

This Flask web development server-based framework can be useful for quick deploying of machine learning models and testing. The app can be taken to the production mode once the iterative testing is done.

Case 3: Deploying models created in MATLAB

We use [MATLAB AppDesigner](#) for this purpose. There are some beginner tutorials given at this [youtube channel](#). You can create MATLAB apps online which would have the .mlapp extension. These apps could be opened and run using the MATLAB AppDesigner.

Currently, one can package an app only in the PC version. The file that gets generated has the .mlappinstall extension. This app can be installed in the online as well as the PC version. To make a WebApp, MATLAB Web Compiler needs to be installed in the MATLAB PC version. Besides this, MATLAB Web App Server needs to be installed. I didn't find this process so straightforward. Web App can't be created in the MATLAB online version as of now 😞.

I would suggest that if you would like to use MATLAB then please use AppDesigner to make an App with your machine learning model.

Case 4: Deploying models in JavaScript

This [article](#) provides some insights into why or why not JavaScript should be used for deploying machine learning models on the web.

[ml5.js](#) is recommended for beginners in machine learning using Javascript. [Tutorial from Daniel Shiffman](#).

We can look at a pre-trained model "Mobilenet" using ml5.js. Example: [hello-ml5](#) [Image classification with Mobilenet video tutorial](#)

It is trained to identify about [1000 objects](#) with over 14 million images.

Please run a local web server ([Why?](#)) (remember MAMP?) to run the ml5.js examples.