# Default Machine Learning Project Guidelines

DTSC691: Applied Data Science
Eastern University

## Purpose:

This project provides an opportunity for students to complete an end-to-end machine learning project, including model deployment. The purpose of this project is to give students an opportunity to identify a useful application of ML models, employ one or more ML models to the problem, interpret results, and create a user-interface that enables users to interact with your ML model(s). The emphasis for this project is placed on the quality of the procedure for applying ML models to a problem, as well as the user-interface.

## Description:

The default ML project contains, but is not limited to, the following elements. Please note that despite this being a "default" ML project, there will still be much variation between specific projects. The following project elements are required to be present in all default ML project submissions, but may take a different form for each student. Accordingly, these required elements are intended to give your project a backbone for structure, not to restrict the design of your work.

- **Data Acquisition**: You must procure an acceptable data set that we can easily be given access to upon request, barring any data privacy concerns (remember, we can sign NDAs). It is perfectly acceptable to use datasets from Kaggle and similar sites, despite there typically being hundreds or thousands of freely available ML notebooks posted alongside each dataset. More emphasis will be placed on the user-interface component of this project during grading should you choose to make use of a Kaggle dataset, since this is where we can be sure that you have performed a sufficient amount of work that is original to you.

- **Data Preparation and Cleaning**: Elements of this component may overlap with the next project component entitled "Exploratory Data Analysis". Your data set may require imputation, if there are any missing values. Your data set may require feature transformations/engineering in case of skewed value distributions

or multicollinearity. These are just a couple examples of procedures you might need to conduct to prepare your data for ML model training. More emphasis will be placed on other components of this project should you acquire an initially clean data set.

- **Exploratory Data Analysis**: Before training any models you must understand your data. You should supply the appropriate summary statistics and visualizations for your features, as well as any other relevant information to characterize your data.

- **ML Model Training**: You must very clearly explain your procedure and intent. You must clearly define your training features and your training response/target (in the case of supervised learning). You must clearly explain why you are training ML models, and how training these models offers insights to the problem at hand. You should provide an overview of the training process that might include some of the following information: models used, number of training iterations/epochs, performance metrics used, procedure for elucidating optimal hyperparameter values, final hyperparameter values employed, procedure for final model selection and justification.

- **ML Model Results and Discussion**: You should discuss the metrics used to evaluate model performance (i.e. MSE, precision, recall, et cetera). What do these performance metrics suggest about the model? How will your trained model be used? What do the model results suggest about the problem at hand? For example, if using MSE for a regression problem, you should give your audience some idea of the magnitude of the MSE of the model relative to the magnitude of the input data in order to get some notion of "goodness" in the fit.

- **ML Model User-Interface Demonstration**: A strong emphasis in this project is on the creation of a user-interface to showcase/deploy your machine learning model. This is important because not only must you be proficient in the building of ML models, but you must also be able to deploy them to a client/boss in a practical manner. Accordingly, we require you to build your model, save your model (using, for example, joblib or pickle), and finally deploy your ML model in a web application using Flask (or Streamlit, Dash, Django, or any other comparable web application framework). We are strongly recommending using Flask.

  Here is a helpful Youtube Flask Tutorial, as well as a link to give you a better idea of how to deploy a an ML model in a Flask app.

You may choose to host your web application locally (i.e. host your app running on a web server from your local machine), in which case we will primarily be interacting with your web application through your video walkthrough that you will submit as part of your project; however, you will also submit all project files so we can launch your web app on our local machines if necessary. On the other hand, you may choose to host your web application using a cloud host (such as Heroku, AWS, or other comparable web hosting platforms). If you choose to launch your web application using a cloud service, then we will be eager to interact directly with your web application live online at the time of grading.

**Flask/User-Interface Requirements:**

You are required to build a personal website that contains the following Flask web pages:

a) (Optional) A biographical homepage, or you could simply have your resume page be your homepage.
b) A resume page with some or all of the following sections:
    - Education
    - Work experience
    - At least one image, picture, or graphic
c) A general projets page, wherein you will link to your other completed projects
d) A specific page for this project, wherein your predictive ML model is deployed

# Deliverables:

The default ML project requires, but is not limited to, the following final submission materials:

- **Project Files**:
    - All code for data cleaning, exploratory analyses, model training, and results interpretation.
    - All code for user-interface/web application
    - Please note that machine learning projects are almost never conducted in Jupyter notebooks, but rather in Python modules/scripts. Jupyter notebooks were used as an effective teaching tool in DTSC670 and DTSC680. Nonetheless, it is perfectly acceptable to conduct some or all of your analyses in either one or more Python modules, or a single Jupyter notebook. Either way, you are expected to discuss the code and your procedure in the video walkthrough.

- **Video Walkthrough**: The video walkthrough is the primary way that we interact with your project. You must submit a <30 minute video walkthrough wherein you outline your work. It is your job to ensure that you strike the appropriate balance between communicating the big picture, but also the necessary details given the time permitted.

- **Project PDF**: Submission materials ***of ALL kinds*** must be converted to a PDF, and submitted as a single, merged PDF document. You must save all python files, other code files, and all submission materials in a PDF form, and merge these PDFs together. You ***must NOT*** take screenshots of any submission materials, and paste them into another document as a means of PDF submission.