This will outline how to retrieve your code, how to retrieve your data, how to retrieve the dependencies (libraries) for running your code, and how to run your code.

# Assignment Details

- @uthor: Ashish Panchal

- GTuser : apanchal33

- Designation: Student- OMSCS Georgia tech

- email : [apanchal33@gatech.edu](mailto:apanchal33@gatech.edu)

- Subject : CS7641- Machine learning

# Title : Instructions for implementing Code for Assignment 1 – Supervised learning

This document details the steps for retrieval of code, data and dependencies, followed by steps for execution of the code.

# The code has directory structure, as follows:

1. Assignment\_1\_main: this is the main folder of the project, containing the following:
   1. File: Assignment\_1\_notebook.ipynb: this is the main ipython notebook, which imports data, dependencies, dataset class, loss curve function , learning curve function, learning curve function for neural network, best model identification function, and code for training and testing different algorithms.
   2. Folder: Figure: this folder the base location for saving plots
   3. Folder: Dataset : this folder used to save ADA object of class dataset defined in assignment\_1\_notebook. Each object per dataset contains all the relevant information generated for all algorithms for specific dataset, including loss and learning curves, best found estimators along with scored for different parameters.
   4. Folder: Raw\_data: this folder contains raw csv data downloaded form Kaggle
      1. WineQT\_2.csv : dataset 1
      2. Genderclass.csv : dataset 2
   5. File: apanchal33\_assignment.pdf : Assignment 1 report

## 1. Retrieval of the Code:

The code is saved at Sharable Box location : <https://gatech.box.com/s/r7so9lfw5qflqzfh27np85v5zqgplir6>

## 2. Retrieval of the data:

There are two datasets used in this code. A copy of these dataset are provided in the code folder, however can be downloaded and added separately. Follow the instructions below.

Step1 . Create a folder Raw\_data if not present inside folder Assignment\_1\_main:

Step 2. Download the following csv data files and ensure the rename the same as defined in the directory description:

For both the files navigate to the Kaggle url and click on download button in the data section, please note this may require you to login through your Kaggle profile, if not present, you may create the same using your email id.

* Dataset 1:
  + Name: Wine quality dataset:
  + File name : WineQT\_2.csv
  + url : www.kaggle.com/datasets/rajyellow46/wine-quality
* Dataset 2:
  + Name: Gender Classification dataset:
  + File name : Genderclass.csv
  + url : [www.kaggle.com/datasets/elakiricoder/gender-classification-dataset](http://www.kaggle.com/datasets/elakiricoder/gender-classification-dataset)

## 3. Retrieval of the Dependencies:

The code uses particular version of different python packages,

These packages with their versions are listed in the requirements.txt file.

Please execute the installation using : ‘pip install -r requirments.txt’ using preferred command line tool or environment from “Assignment\_1\_main” location.

## 4. Execution of code:

The code notebook is created in a monolithic fashion. It contains all the required class and function definitions used in the code. And can be executed in a serialized way from first cell to the last.

The code is divided in to 4 parts

1. Importing of libraries
2. Handling data:
   1. Definition of data\_set\_module class : an Analytical object for storing analysis objects for specified data. For further details please refer code
   2. Importing data, processing data and short EDA.
   3. Splitting of training and test data.
   4. Creation and saving of dataset object.
3. Definition of utility functions. There are 5 functions created and used. Please refer the code for more details
   1. Loss\_curve
   2. Find\_best
   3. save\_loss\_curve
   4. plot\_learning\_curve
   5. GS\_get\_best\_estimator
   6. epoch\_learning\_curve
4. Implementation and Analysis of Algorithms : there are 5 algorithms which explored, as mentioned in the report, each algorithm analysis contains 3 parts each per data set.
   1. Loss curve
   2. Find best
   3. Learning curve
   4. \* Learning curve epoch : only for NN

Each part create creates objects for analysis and adds to the “ADA” dictionary for the algorithm key for the specific dataset. At the same time these also generate and save loss and learning curve In the figure folder.

Loss curve figures follows the below nomenclature:

('Loss Curve-'+algorithm+' - '+ data\_set+'-'+GS\_params['scoring']+misc)

Where :

* + Algorithm : is the name of algorithms passed as string argument to the loss\_curve function.
  + Data\_set: is the name of dataset used, passed as string argument to the loss\_curve function.
  + GS\_params['scoring'] : is the name of ‘scorer’ used in grid search, passes as string value to the key 'scoring' for the grid search parameters, GS\_params, passed as an argument to the loss\_curve
  + Misc: is the user defined parameter name, passed as string argument to the loss\_curve function. In most cases it the name of parameter under study.

Learning curve figures follows the below nomenclature:

('Learning Curve-'+algorithm+' - '+ data\_set+'-'+ learning\_curve\_config['scoring']+'-'+misc)

Where most details are same as loss curve, however GS\_params[‘scoring’] is changed with learning\_curve\_config['scoring'], however holds the same context.

Lastly Learning curve for epoch NN follows the below nomenclature:

('Learning Curve-Epoch - '+algorithm+' - '+ data\_set\_name+'-'+scoring+'-'+misc)

Here the “scoring” is a string argument specifically provided as a string argument replacing learning\_curve\_config['scoring']

And ‘data\_set\_name” is used instead of ‘data\_set’ as a string argument.

1. Ways for execution:
   1. There are two ways of execution of the code.
      1. Use prebuilt data\_set\_module objects, which were created while experimentation
         1. For this, raise the data\_set\_module\_flag to “1” in the 2nd cell
         2. These object can be directly used for different algorithmic analysis and can be used at any part of code
      2. Create new datasets:
         1. Keep the data\_set\_module\_flag to “0” in the 2nd cell.
         2. This may limits the use of the dataset, that is, the code for every algorithm has to be executed in a serialized fashion. Although it is still possible to jump to any of the algorithms, except Adaboost, which is dependent on decision trees algorithm.