Homework Assignment 5 (Programming Only)

You choose **only one** of the following tasks.

There will be no step-by-step instructions, and you need to write the code from scratch using Jupyter notebook.

Write comments in your file, so that I can understand what you are doing.

Programming bugs (i.e., your Python program crashes) are not acceptable. If there is an error message in your file, you will lose a random number of points. Note: "warning" is not an error. a random number of points = 100*np.random.rand()

You may use multiple Jupyter notebook files: one file for one sub-task

Task-1: Stochastic Grid Search for the optimization of multiple hyperparameters of random forest for regression.

You have done single-hyperparameter optimization in homework 3. In this task, you will develop a method to find the best values of all the hyperparameters of random forest on the same dataset in H3P2T2.
Grid search: discretize the hyperparameter space using a grid, and evaluate model performance using hyperparameters from grid nodes
Random search: generate random hyperparameters and evaluate model performance.
Stochastic grid search: combine grid search and random search, that is, randomly sample hyperparameters from a grid in the hyperparameter space. This method does not need deep nested for loops, which makes it easy to control the time cost. You will develop this method: write your own code from scratch, instead of using RandomizedSearchCV in sk-learn

You split the data once to get training, validation, and test sets.
To simplify your task, you only need to consider the parameters in blue color.

class sklearn.ensemble.RandomForestRegressor(n_estimators=100, *, criterion='mse', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0, max_samples=None)

Remember that: to get the best max-depth (or other hyperparameters) of random forest, you may need to consider both model accuracy and complexity. Thus, you need to define a metric to measure the goodness of a model.

Compare your method with RandomizedSearchCV in sk-learn. It does not matter if your implementation is not better than that in sk-learn.
Compare your method with auto-sklearn at https://automl.github.io/auto-sklearn/master/

Task-2: Classification of Fashion-MNIST images.

In this task, you will build four models to classify the images in the Fashion-MNIST dataset.

(1) logistic regression
(2) Random Forest
(3) MLP
(4) 2D CNN

Compare the performance, and find the best one.

do hyperparameter optimization (at least one parameter for each model, except (1)). You will get zero score, if there is no hyperparameter optimization for the models. For model (1), you do not need to do hyperparameter optimization.

Do NOT copy the existing models from Github. (if you do so, you will get zero score).

Do NOT use test set for hyperparameter optimization. (if you do so, you will get zero score).

Do NOT use weighted accuracy. The standard accuracy is fine.

use the code below to load data**:**

```
import tensorflow as tf
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

Your score is the max of test accuracy scores of the four models.

Task-3: Transfer learning for COVID-19 diagnosis using CT images.

You will develop a 2D CNN for COVID-19 diagnosis using CT images.
I will provide you the dataset.
(1) define a CNN based on Resnet-18 for binary classification
(2) train the CNN from scratch
(3) train the CNN using transfer learning
Compare the two models in (2)&(3) to see if transfer learning brings extra benefits or not.

Task-4: Apply what you have learned to a new dataset in your field of study

You need to send your proposal to me and get written approval from me.

It should be not too easy and not too hard.