A3 (20%) Submission due on 13 November 2020 (Friday)

Write a single python file to perform the following tasks:

(a) Get dataset "`from sklearn.datasets import fetch_california_housing`". Split the database into two sets: one set for training, and the remaining set for testing.
NOTE 1: Please use "`from sklearn.model_selection import train_test_split`" with "`random_state=N`" and "`test_size=TestSize`".
NOTE 2: The offset/bias column will not be needed here to augment the input features in regression trees and random forest.

(b) Train a decision tree regressor ("`from sklearn.tree import DecisionTreeRegressor`") using the training set with maximum depths from 1 to MaxTreeDepth utilizing "`criterion='mse'`" at "`random_state=0`".
NOTE: All remaining parameters should not be set (i.e., use default values).

(c) Compute the training mean squared error (mse) based on "`from sklearn.metrics import mean_squared_error`".

(d) Compute the prediction mse for the test set.

(e) Repeat (b) to (d) using the random forest regressor "`from sklearn.ensemble import RandomForestRegressor`" (utilizing "`criterion='mse'`" at "`random_state=0`" too).
NOTE: All remaining parameters should not be set (i.e., use default values).

Submit a single python file with filename "A3_StudentMatriculationNumber.py". It should contain a function `A3_MatricNumber` that takes the following inputs and returns the following outputs in the following order:

Python function inputs:

- `N`: an integer to set the random state for `train_test_split`.
- `TestSize`: a fraction that falls between 0 and 1 (e.g., 0.2, 0.8, etc.) for `train_test_split`
- `MaxTreeDepth`: an integer that specifies the maximum depth of decision tree. For example, if `MaxTreeDepth` is 5, then your code should train decision trees and random forests for max tree depths 1, 2, 3, 4 and 5.

Python function outputs in the following order:

- `X_train`: training numpy feature matrix of dimensions (number_of_training_samples × 8). (1%)
- `X_test`: test numpy feature matrix of dimensions (number_of_test_samples × 8). (1%)
- `y_train`: training numpy target array of length number_of_training_samples. (1%)
- `y_test`: test numpy target array of length number_of_test_samples. (1%)
- `ytr_Tree_list`: list of training set predictions for trees with maximum depth from 1 to `MaxTreeDepth`. For example, ytr_Tree_list[0] should be a numpy array of length number_of_training_samples containing the training set predictions of the tree trained with max depth 1. ytr_Tree_list[1] should be a numpy array of length number_of_training_samples containing the training set predictions of the tree trained with max depth 2. (2%)
- `yts_Tree_list`: list of test set predictions for trees with maximum depth from 1 to `MaxTreeDepth`. For example, yts_Tree_list[0] should be a numpy array of length number_of_test_samples containing the test set predictions of the tree with max depth 1 (trained using the training set). yts_Tree_list[1] should be a numpy array of length number_of_test_samples containing the test set predictions of the tree with max depth 2 (trained using the training set). (2%)
- `mse_trainTree_array`: numpy array of length `MaxTreeDepth` containing the mean squared errors for the training set. For example, mse_trainTree_array[0] is the training set mse for the tree trained with max depth 1. mse_trainTree_array[1] is the training set mse for the tree trained with max depth 2. (2%)
- `mse_testTree_array`: numpy array of length `MaxTreeDepth` containing the mean squared errors for the test set. For example, mse_testTree_array[0] is the test set mse for the tree with max depth 1 (trained using the training set). mse_testTree_array[1] is the test set mse for the tree with max depth 2 (trained using the training set). (2%)
- `ytr_Forest_list`: list of training set predictions for forests with maximum depth from 1 to `MaxTreeDepth`. For example, ytr_Forest_list[0] should be a numpy array of length number_of_training_samples containing the training set predictions of the forest trained with max depth 1. ytr_Forest_list[1] should be a numpy array of length number_of_training_samples containing the training set predictions of the forest trained with max depth 2. (2%)

- `yts_Forest_list`: list of test set predictions for forests with maximum depth from 1 to `MaxTreeDepth`. For example, yts_Forest_list[0] should be a numpy array of length number_of_test_samples containing the test set predictions of the forest with max depth 1 (trained using the training set). yts_Forest_list[1] should be a numpy array of length number_of_test_samples containing the test set predictions of the forest with max depth 2 (trained using the training set). (2%)
- `mse_trainForest_array`: numpy array of length `MaxTreeDepth` containing the mean squared errors for the training set. For example, mse_trainForest_array[0] is the training set mse for the forest trained with max depth 1. mse_trainForest_array[1] is the training set mse for the forest trained with max depth 2. (2%)
- `mse_testForest_array`: numpy array of length `MaxTreeDepth` containing the mean squared errors for the test set. For example, mse_testForest_array[0] is the test set mse for the forest with max depth 1 (trained using the training set). mse_testForest_array[1] is the test set mse for the forest with max depth 2 (trained using the training set). (2%)

Please use the python template provided to you. Remember to rename both "A3_StudentMatriculationNumber.py" and "`A3_MatricNumber`" using your student matriculation number. For example, if your matriculation ID is A1234567R, then you should submit "A3_A1234567R.py" that contains the function "A3_A1234567R". Please do NOT zip/compress your file. Because of the large class size, **points will be deducted if instructions are not followed**. The way we would run your code might be something like this:

```
>> import A3_A1234567R as grading
>> N = 5
>> TestSize = 0.3
>> MaxTreeDepth = 5
>> X_train, y_train, X_test, y_test, ytr_Tree_list, yts_Tree_list,
mse_trainTree_array, mse_testTree_array, ytr_Forest_list, yts_Forest_list,
mse_trainForest_array, mse_testForest_array = grading.A3_A1234567R(N,
TestSize, MaxTreeDepth)
```

Submission folder: LumiNUS >> files >> A3
(The submission folder in LumiNUS will be closed on 13 November at 2359 hour)