# Project House Price

## Step-1: clean the test data (30 points)

 the test data columns must be the same as the training data columns the same names in the same order

Do NOT delete rows in test data

Do NOT modify the training data (clean)

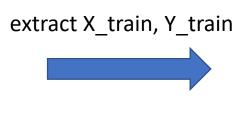
upload your ipynb file to Blackboard
 for example: house\_price\_step1\_your\_name.ipynb

### Step-2: predict house price (70 points)

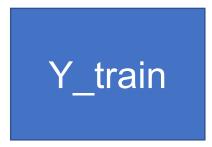
- This step-2 has two parts: Part-1 (40 points) and Part-2 (30 points)
- Submit two ipynb files to Blackboard house\_price\_step2\_part1\_your\_name.ipynb house\_price\_step2\_part2\_your\_name.ipynb

load training data (clean) by Pandas

Training Data (clean)



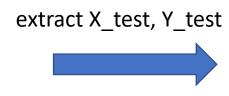




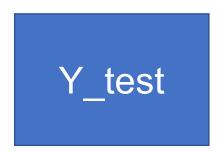
Y\_train has only one column "SalePrice" from the training data X\_train has most of the columns from the training data and it does not have "SalePrice"

load test data (clean) by Pandas

Test Data (clean)







Y\_test has only one column "SalePrice" from the test data X\_test has most of the columns from the test data and it does not have "SalePrice"

Convert Pandas datatype to Numpy array

X\_train=X\_train.values

Y\_train=Y\_train.values.reshape(-1)

X\_test=X\_test.values

Y\_test=Y\_test.values.reshape(-1)

Normalize X\_train and X\_test using MinMaxScaler (see examples presented in class)

use four types of regressors with their default parameter values

KNeighborsRegressor
LinearRegression
DecisionTreeRegressor
RandomForestRegressor
(see examples presented in class)

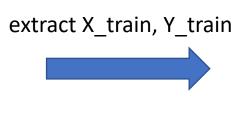
For each regressor model:

fit the model to the training data evaluate the model on the test data to get MSE, MAE, and MAPE show the "45 degree line plot (y\_true vs y\_pred)" for each model evaluated on the test data report the result in a table:

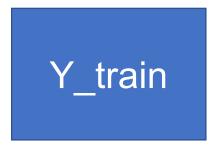
	KNeighbors Regressor	LinearRegre ssion	DecisionTree Regressor	RandomFore stRegressor
MSE				
MAE				
MAPE				

load training data (clean) by Pandas

Training Data (clean)



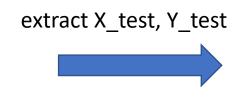




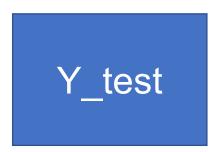
Y\_train has only one column "SalePrice" from the training data X\_train has most of the columns from the training data and it does not have "SalePrice"

load test data (clean) by Pandas

Test Data (clean)







Y\_test has only one column "SalePrice" from the test data X\_test has most of the columns from the test data and it does not have "SalePrice"

Convert Pandas datatype to Numpy array

X\_train=X\_train.values

Y\_train=Y\_train.values.reshape(-1)

X\_test=X\_test.values

Y\_test=Y\_test.values.reshape(-1)

Normalize X\_train and X\_test using MinMaxScaler (see examples presented in class)

use three types of regressors, and find the best hyper-parameters of the regressors. To reduce the computation cost, you only need to optimize one (*parameter*) for each regressor

KNeighborsRegressor (*n\_neighbors*)
DecisionTreeRegressor (*max\_depth*)
RandomForestRegressor (*max\_depth*)
(see examples presented in class)
parameter range: 1 to 100, with the step of 10

report the evaluation result of the best models in a table:

	KNeighbors Regressor	DecisionTree Regressor	RandomFore stRegressor
MSE			
MAE			
MAPE			