

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

Step-2: Part-1

load training data (clean) by Pandas

Training Data
(clean)

extract X_train, Y_train



X_train

Y_train

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)

extract X_test, Y_test



X_test

Y_test

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"

Step-2: Part-1

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)

Step-2: Part-1

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:

	KNeighborsRegressor	LinearRegression	DecisionTreeRegressor	RandomForestRegressor
MSE				
MAE				
MAPE				

Step-2: Part-2

load training data (clean) by Pandas

Training Data
(clean)

extract X_train, Y_train



X_train

Y_train

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)

extract X_test, Y_test



X_test

Y_test

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"

Step-2: Part-2

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)

Step-2: Part-2

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			