Minh Vu, Duc Nguyen – Final Project checkpoint

Professor Steven Bogaerts

Data Mining Checkpoint Report

1. Introduction:

Using gradient boosting regression to predict housing price

2. Read Data Function:

First, we read the training and testing data set using pd.read_csv.

3. Create Preprocessing function:

A. Using featured engineering to add two more variables: YrOld(YrSold – YearBuilt), YardSize(LotArea – GrLivArea)

B. Using Correlation matrix, draw the matrix of new correlation between variables. Figuring out YrOld have no relationship with Saleprice. Thus we commented out the YrOld addition.

Looking at the matrix to find the 10 least correlated variables (using the nsmallest()) to sale price and drop them from the training data set.

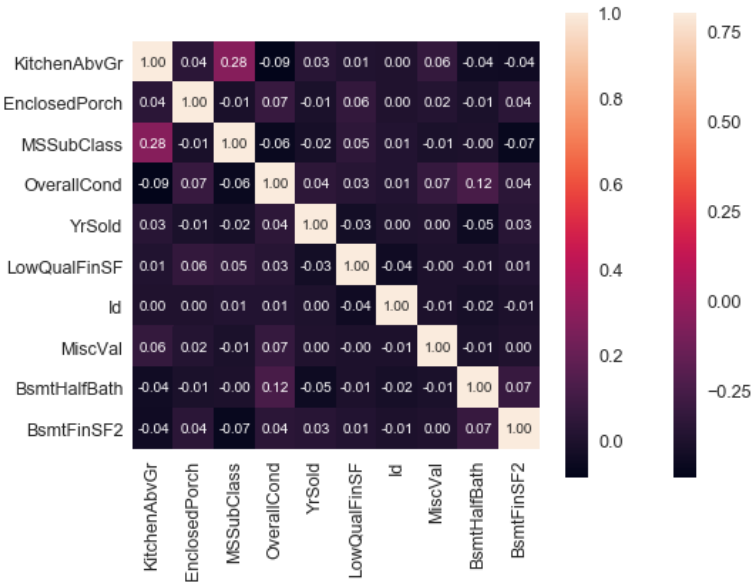

Image 1. Top 10 worst correlation with sale price.

C. Then we use isnull() to find the percentage of missing data of each column.

	Total	Percent
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageQual	81	0.055479
GarageType	81	0.055479
GarageYrBlt	81	0.055479
GarageFinish	81	0.055479
GarageCond	81	0.055479
BsmtFinType2	38	0.026027
BsmtExposure	38	0.026027
BsmtCond	37	0.025342
BsmtQual	37	0.025342
BsmtFinType1	37	0.025342
MasVnrArea	8	0.005479
MasVnrType	8	0.005479
Electrical	1	0.000685
Neighborhood	0	0.000000

Image 2. First 20 columns rank from highest NaN percentage to lowest.

D. For every column that have a 50% missing value or higher, we will drop it.

E. Then we find other column with integer value, and fill NaN with the mean() of that column

F. Similarly we find the column with string value and fill NaN with the mode() of that column (we tried using get_dummies function here to change all the string value to integer to prepare to use the model later on but since the number of possibilities in the train data set and the test data set is different -> it creates a different number of addition columns -> cannot run prediction model on the test set. -> we fix it later on by concat and split it in main())

4. Predicting function

A. Preprocess train data set
B. Preprocess test data set
C. Concatenate train and test data set
D. Use get_dummies to change the string value to integer
E. Then we standardize the data frame.

$$z = \frac{X - \mu}{\sigma} =$$

Z = z-score

X = value of a column

μ= mean of a column

σ = Standard deviation

F. Split the concatenated data set back into train and test with its original row length.
    a. Now we have a train and test set with all integer value.
G. Create an object of Gradient Boosting Repressor Class
H. Use the object to do cross_val_score on train data set
I. Fit the object with train data set and output series("SalePrice")
J. Use the object to predict test price

5. Result

We got the MSE of training data set of around 725482737.318 (sqrt = ~~26700)

And we created a prediction file (submission.csv) for the prediction model of the testing data set:

Id	SalePrice
1461	122413.444
1462	160805.571
1463	177228.795
1464	182980.312
1465	199187.226
1466	172605.26
1467	165264.566
1468	163212.841
1469	194982.252
1470	130152.22
1471	208523.906
1472	97190.3351
1473	95115.3587
1474	155481.532
1475	140909.607
1476	415570.943
1477	279366.327
1478	307183.88
1479	275395.108