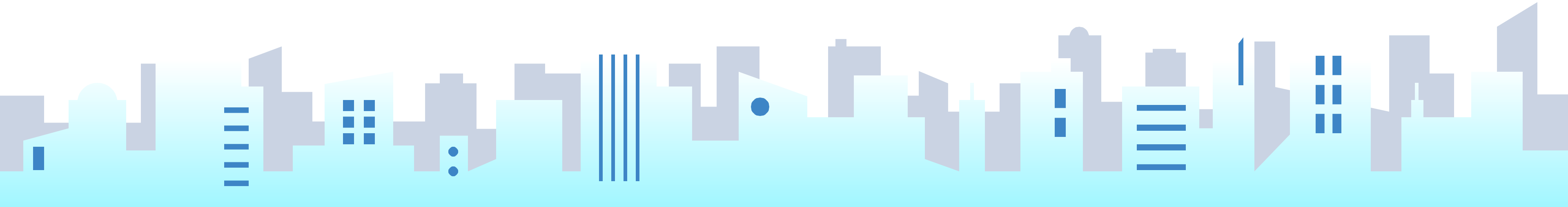
HOUSING PRICE REGRESSION ANALYSIS

MICHELLE (QIN) PENG

GitHub:
Username: misseuro
URL: https://github.com/misseuro/housingprice/blob/master/FinalProject_StatisticLearning.ipynb

DATASET OVERVIEW



Data Source:

Kaggle’s Ames Housing dataset

Programming Language: Python

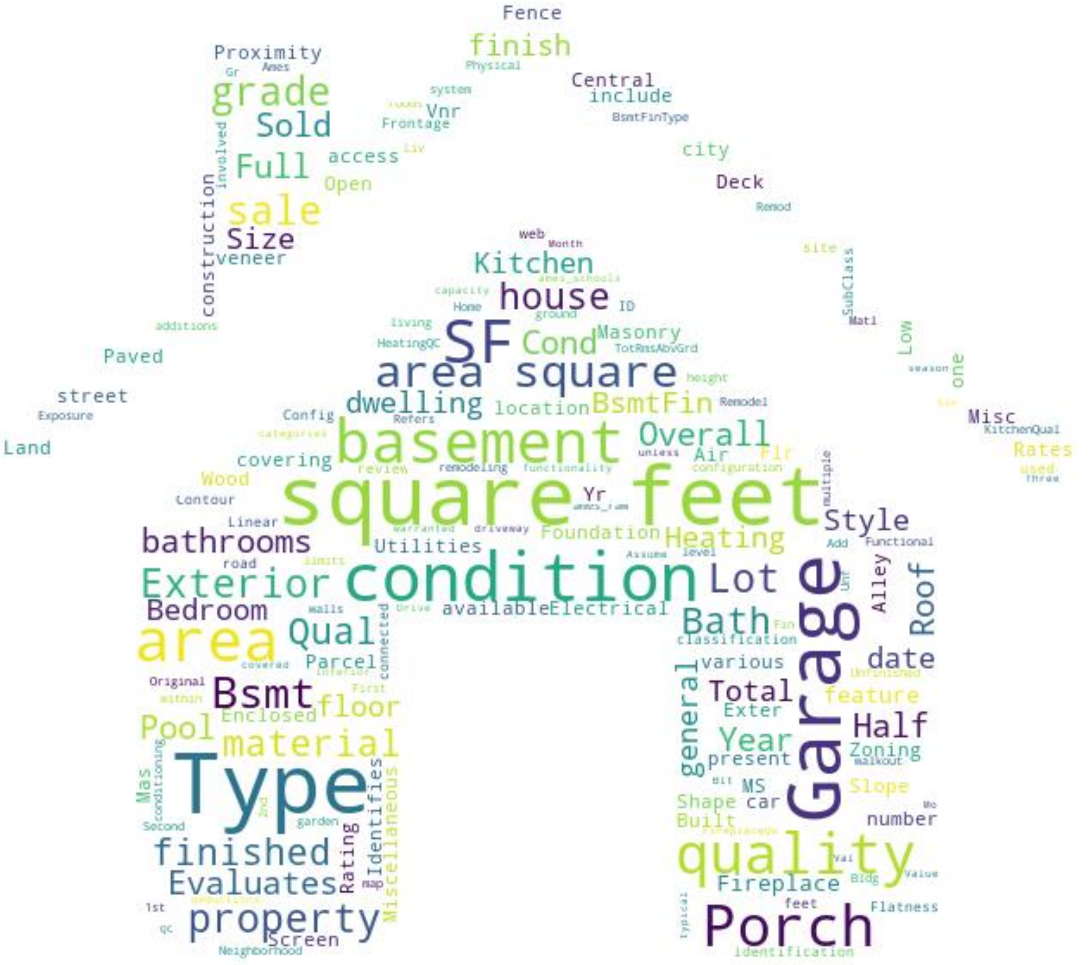
Number of Observations: 1460

Response Variable: SalePrice

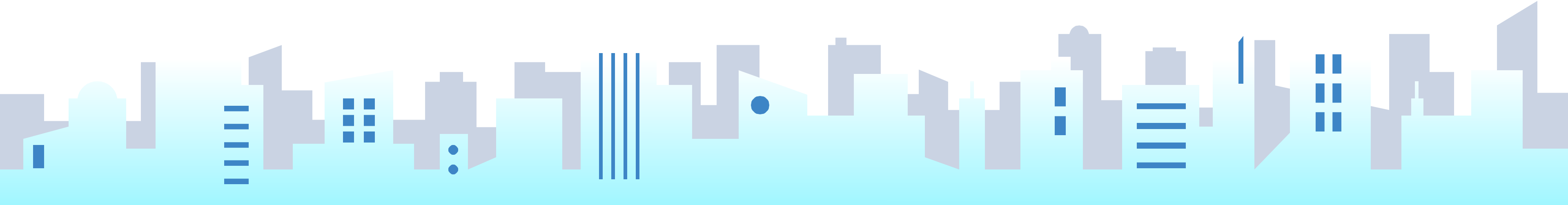
Features: 81

Numerical Variables: 38

Categorical Variables: 43

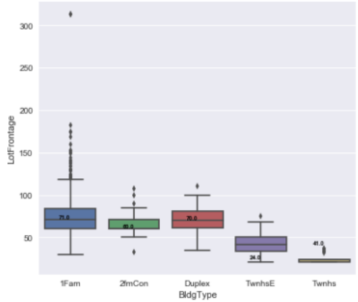


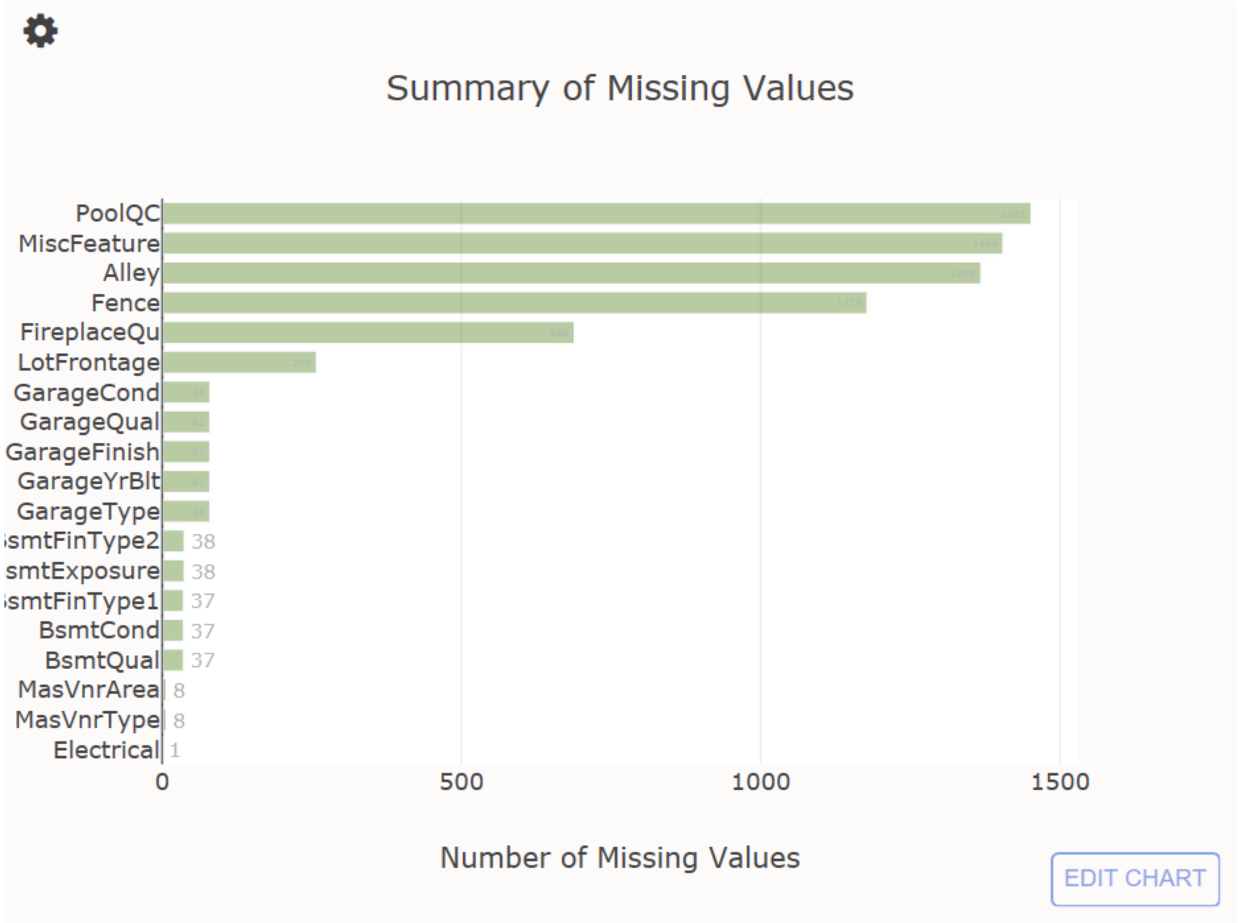
DATA PREPROCESSING



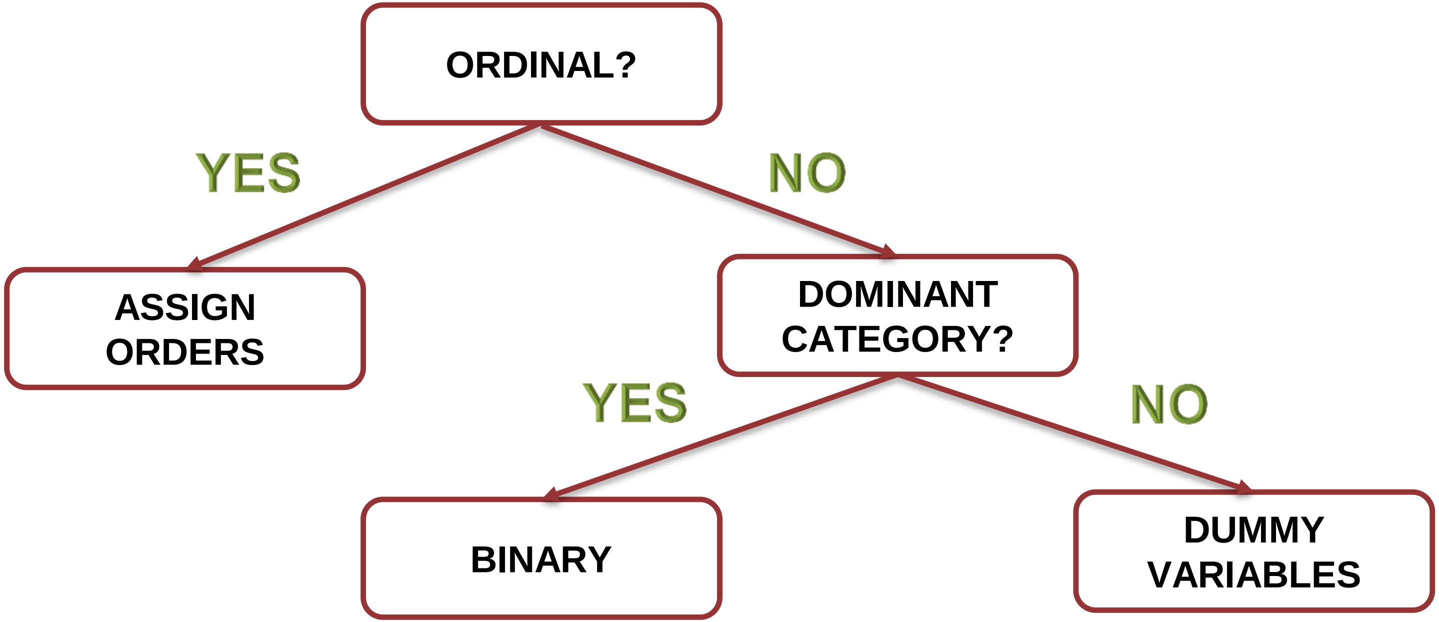
IMPUTE – STEP1 MISSING VALUES:

1) Delete the columns with over 90% missing values
2) Assign 0 to Missing values of Ordinal variables
   eg: Fence, FireplaceQu
3) Assign group medians to missing values:
   eg: LotFrontage





ENCODING – STEP2
CATEGORICAL VARIABLES



ORDINAL?

YES

NO

ASSIGN ORDERS

DOMINANT CATEGORY?

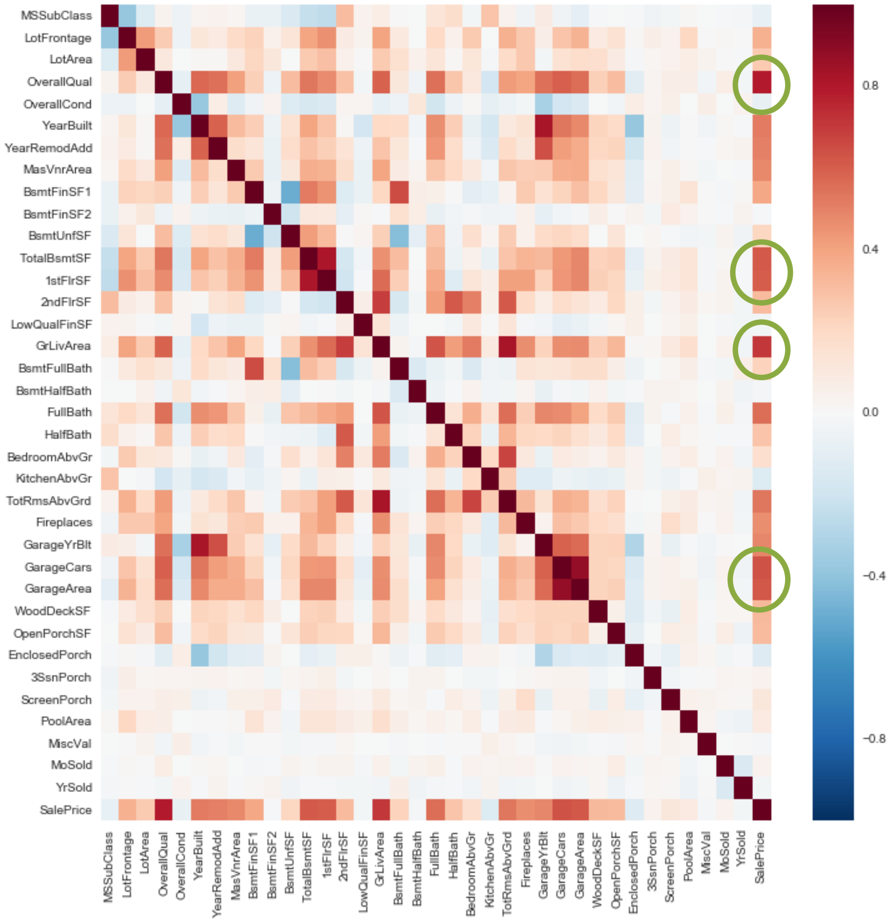
YES

NO

BINARY

DUMMY VARIABLES

FEATURE – STEP3 ENGINEERING



OverallQual: [Ordinal]
Rates of the overall material and finish of the house from 1 to 10

GrLivArea: [Numeric]
Above ground Living Area

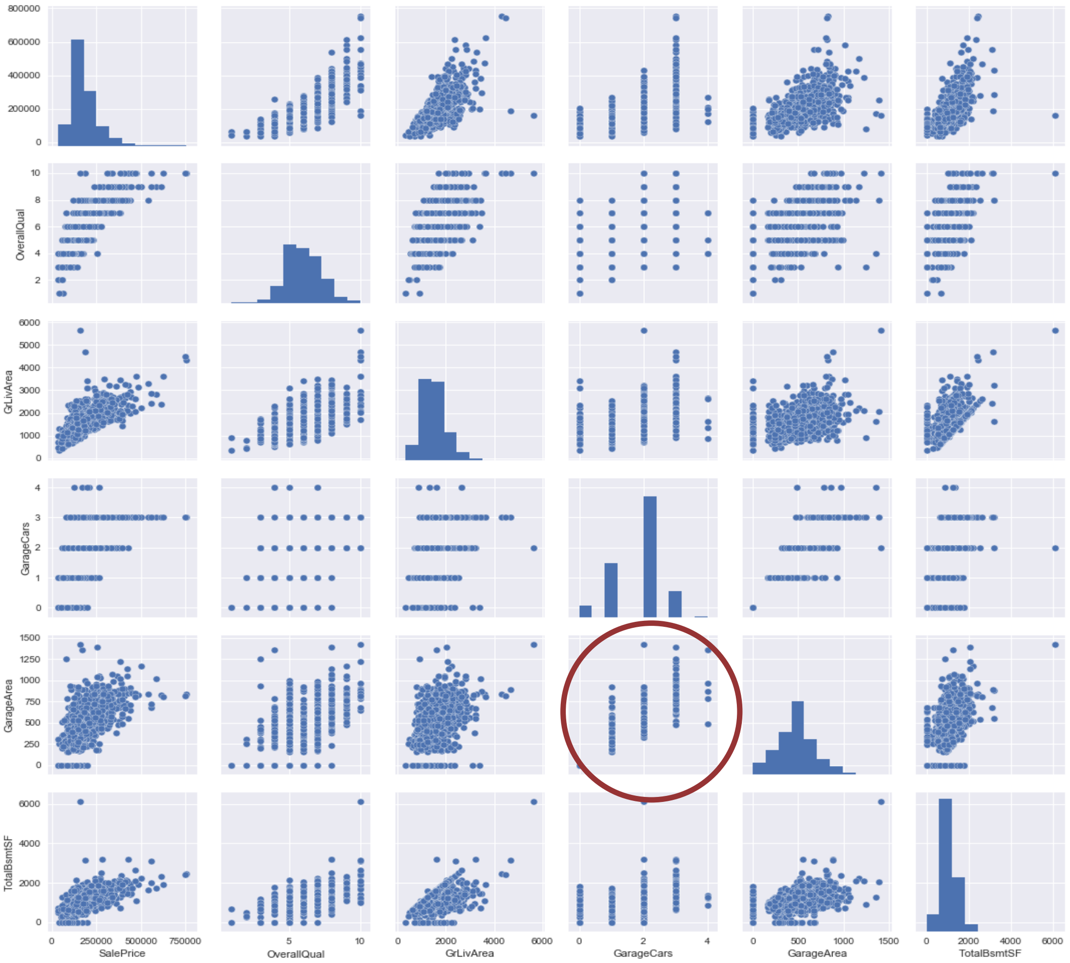
GarageCars: [Numeric]
Size of garage in car capacity

GarageArea: [Numeric]
Size of garage in square feet

TotalBsmtSF: [Numeric]
Total Square Feet of Basement

MULTICOLLINEARITY

VIF	FEATURES
5.290448	Fireplaces
5.481232	FireplaceQu
5.769224	BsmtQual
6.085164	TotRmsAbvGrd
6.259153	GarageYrBlt
7.244331	GarageCars
7.303996	GarageArea
16.606101	YearBuilt
19.048027	GarageQual
20.162878	GarageCond
20.220397	BldgType

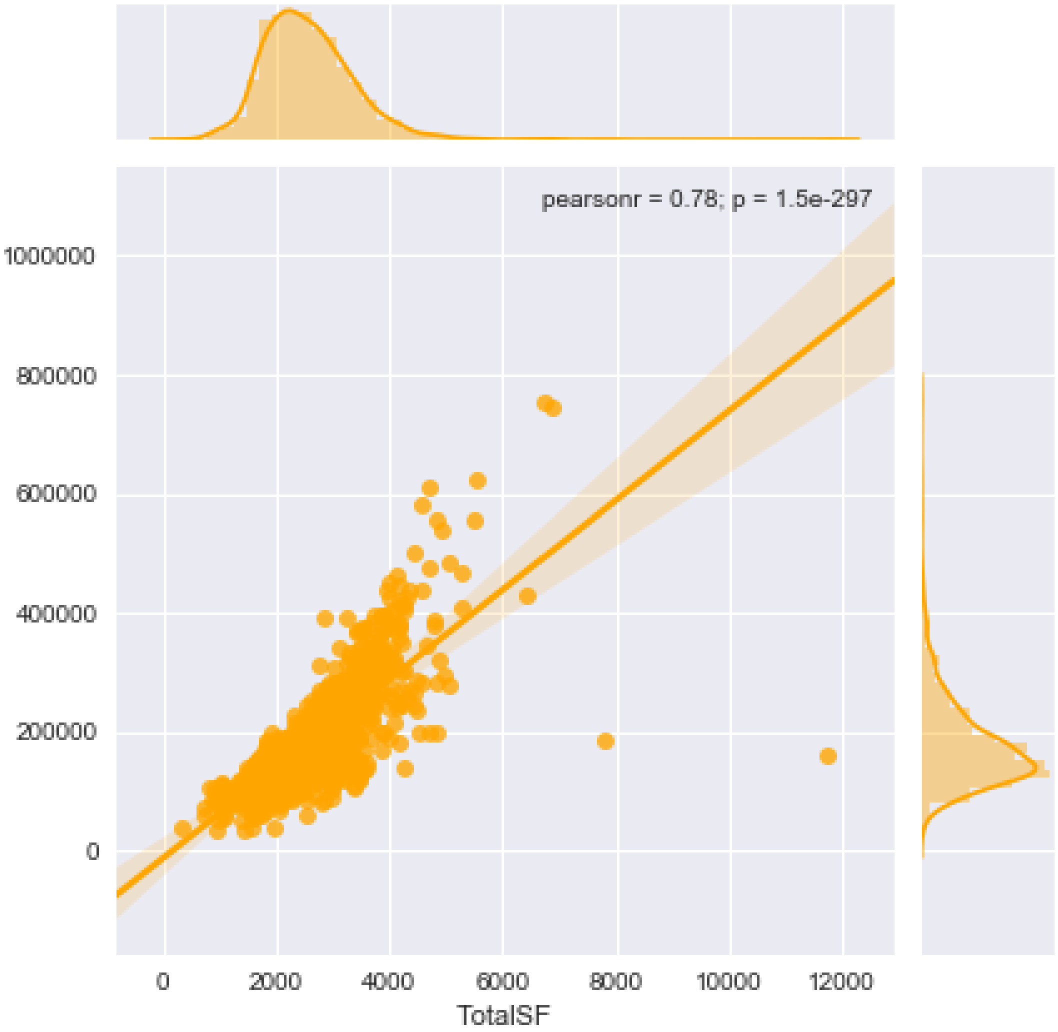


1) Construct 2 new variables:

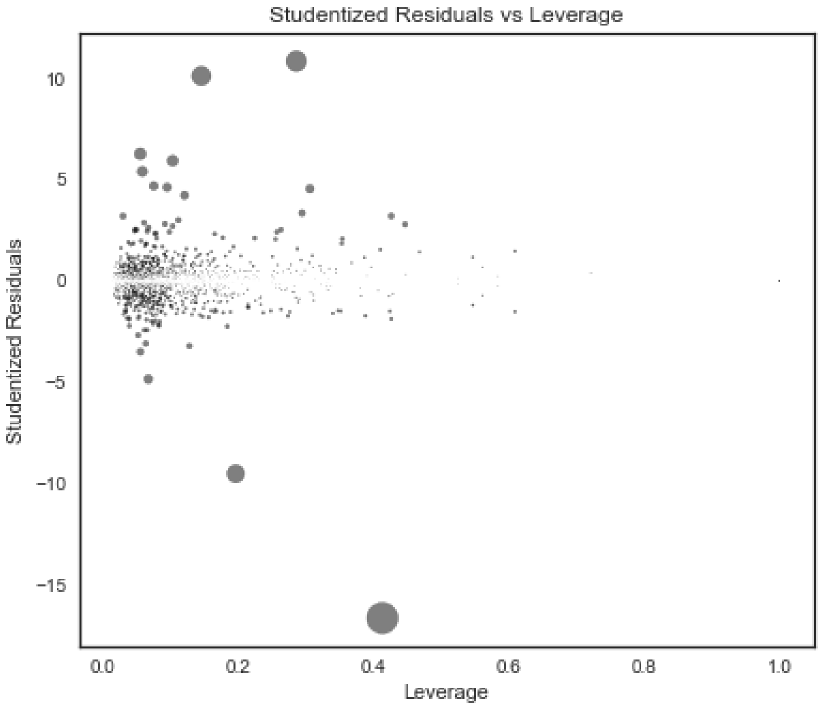
Total Square Feet =
GrLivArea + TotalBsmtArea

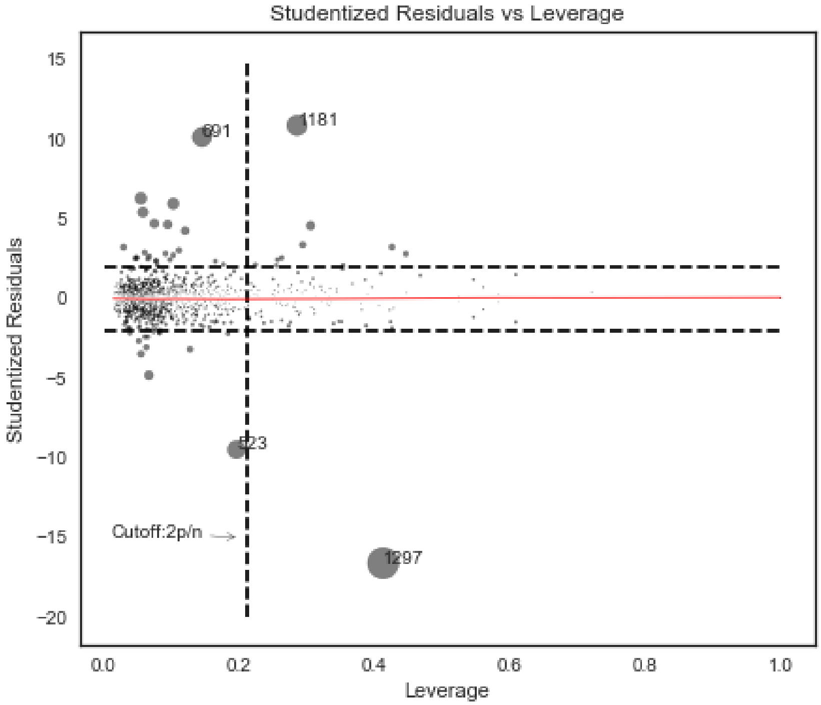
Porch =
OpenPorchSF+EnclosedPorch+3SsnPorch+ScreenPorch

2) Pick between Quality and Condition variables

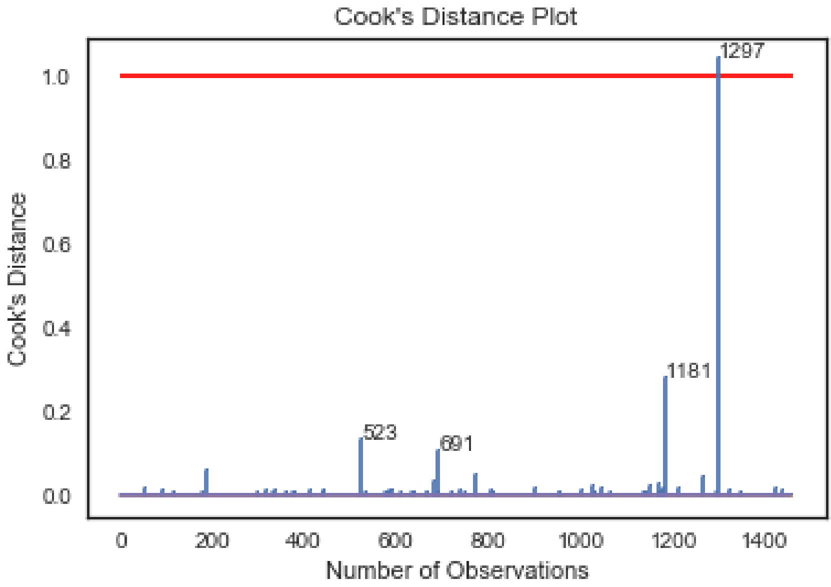


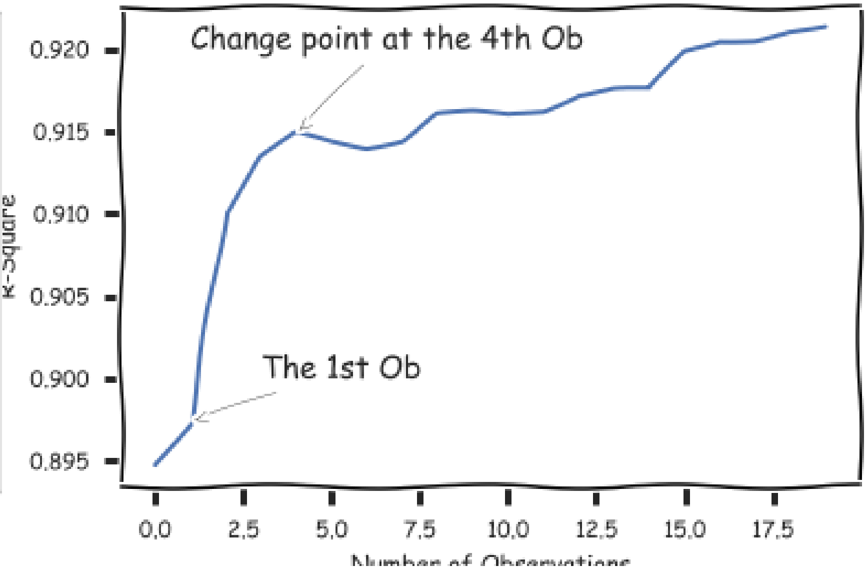
OUTLIERS – STEP4



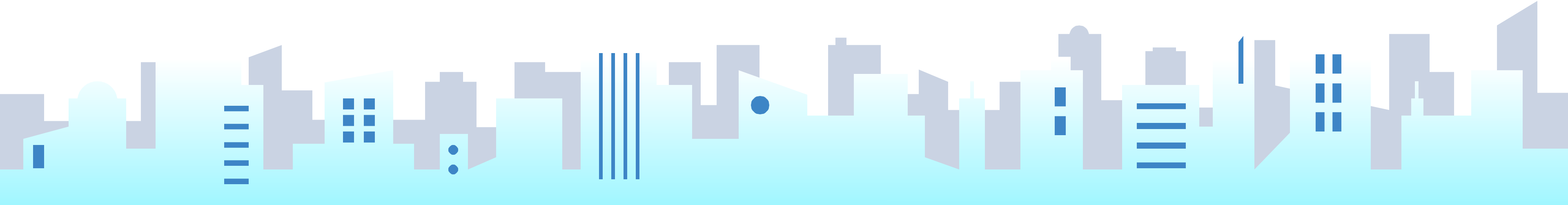


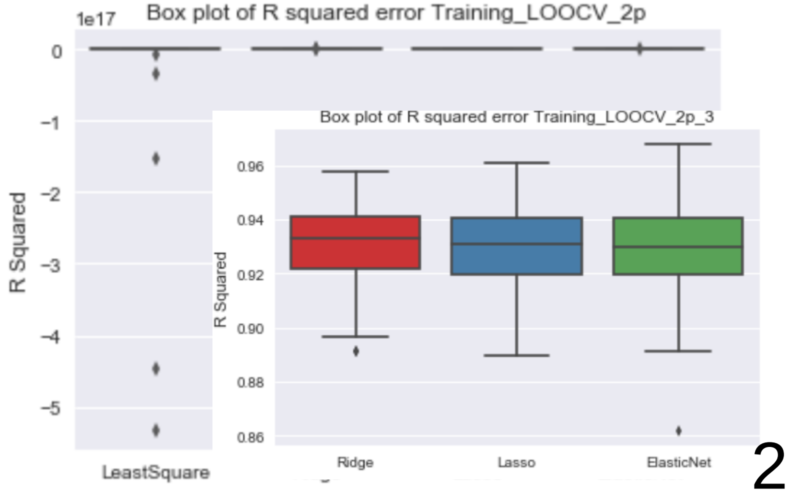
OUTLIERS – STEP4





REGRESSION AND MODEL SELECTION

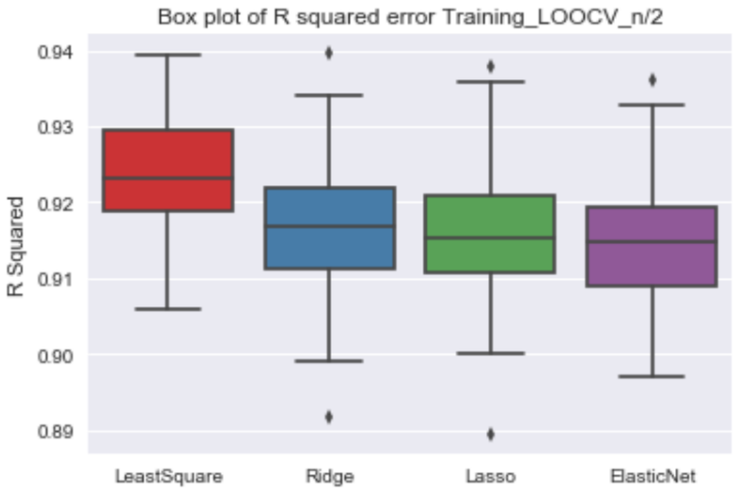




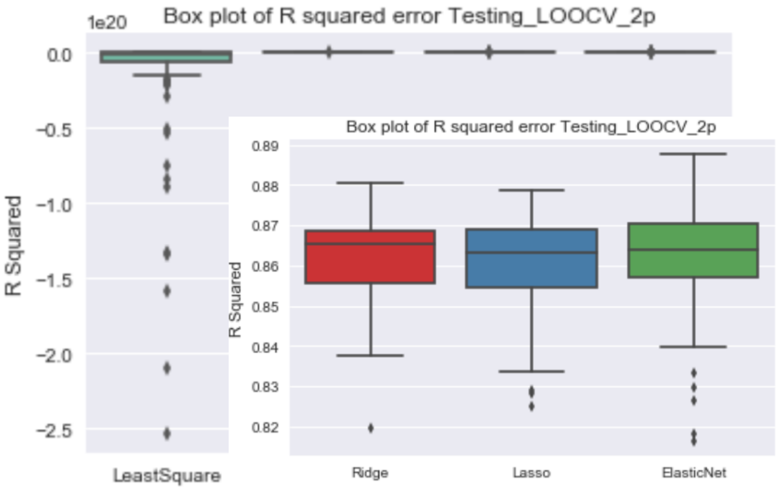
Box plot of R squared error Training_LOOCV_2p

Box plot of R squared error Training_LOOCV_2p_3

2



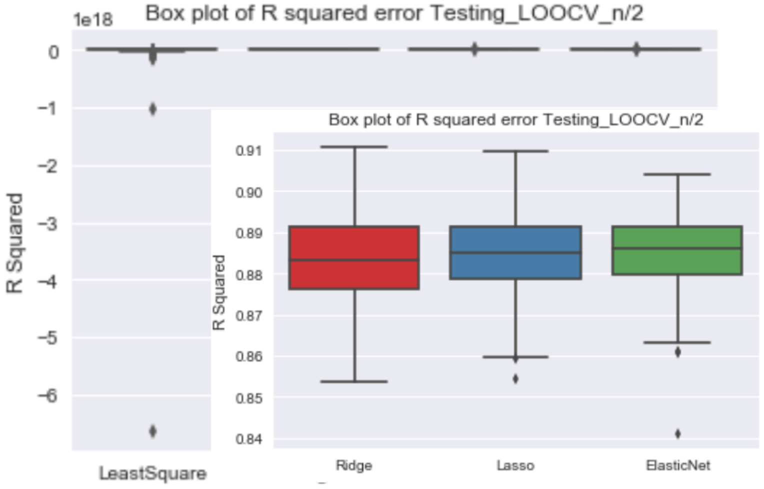
Box plot of R squared error Training_LOOCV_n/2



Box plot of R squared error Testing_LOOCV_2p

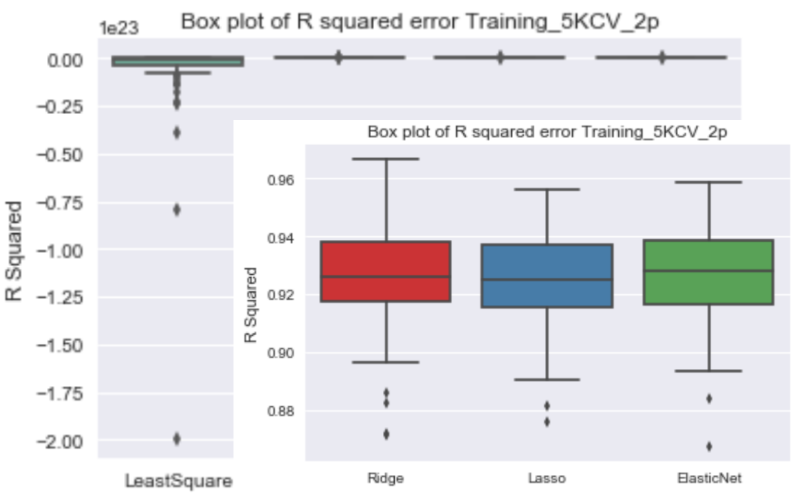
Box plot of R squared error Testing_LOOCV_2p

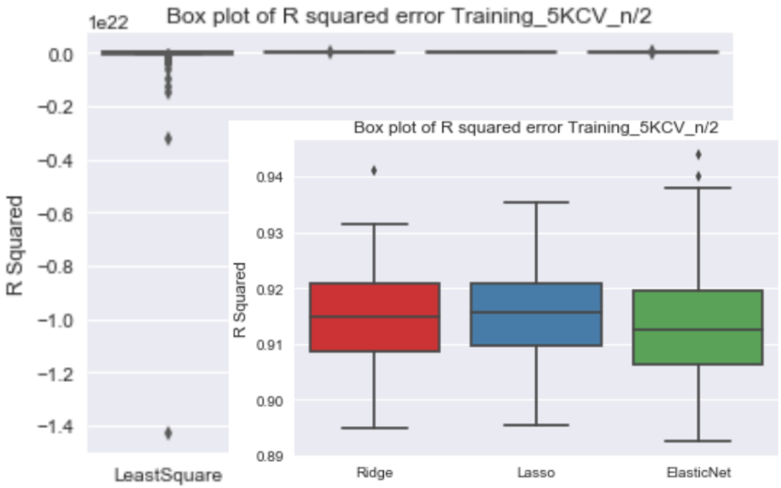
P



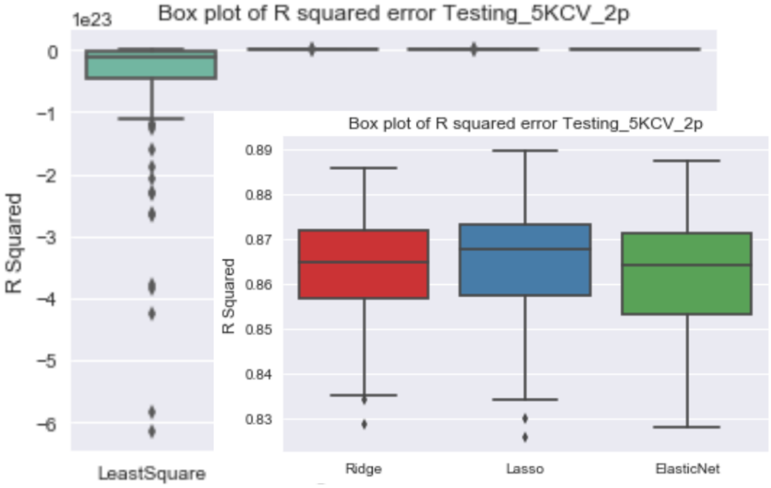
Box plot of R squared error Testing_LOOCV_n/2

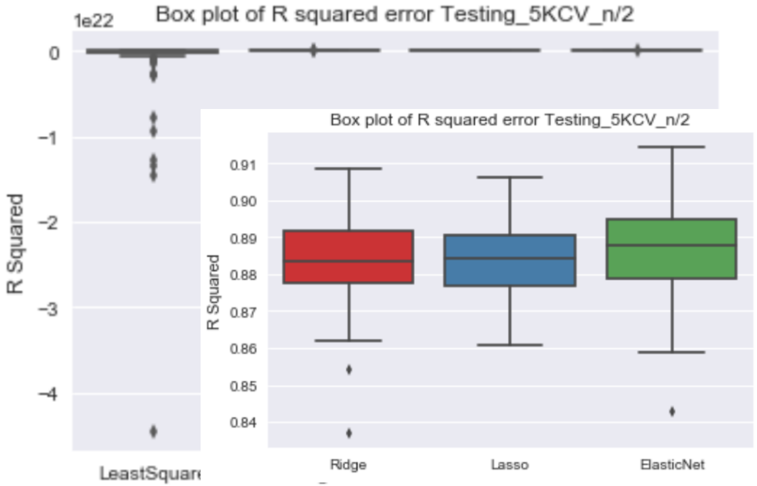
Box plot of R squared error Testing_LOOCV_n/2

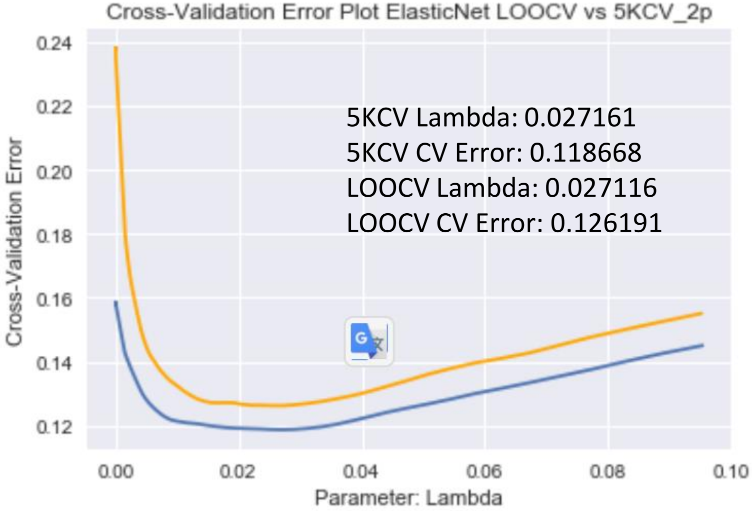

Box plot of R squared error Training_5KCV_2p
1e23
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00
R Squared
LeastSquare
Box plot of R squared error Training_5KCV_2p
0.96
0.94
0.92
0.90
0.88
R Squared
Ridge
Lasso
ElasticNet


Box plot of R squared error Training_5KCV_n/2
1e22
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
−1.4
R Squared
LeastSquare
Box plot of R squared error Training_5KCV_n/2
0.94
0.93
0.92
0.91
0.90
0.89
R Squared
Ridge
Lasso
ElasticNet

N/2

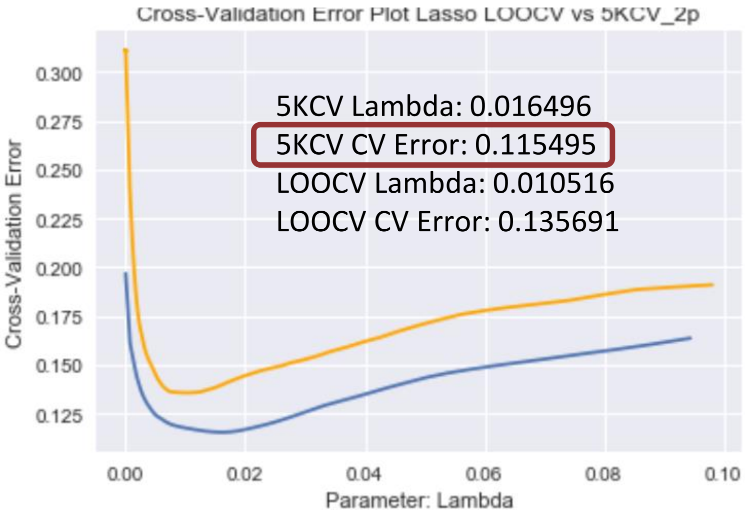

Box plot of R squared error Testing_5KCV_2p
1e23
0
−1
−2
−3
−4
−5
−6
R Squared
LeastSquare
Box plot of R squared error Testing_5KCV_2p
0.89
0.88
0.87
0.86
0.85
0.84
0.83
R Squared
Ridge
Lasso
ElasticNet


Box plot of R squared error Testing_5KCV_n/2
1e22
0
−1
−2
−3
−4
R Squared
LeastSquare
Box plot of R squared error Testing_5KCV_n/2
0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
R Squared
Ridge
Lasso
ElasticNet



Cross-Validation Error Plot ElasticNet LOOCV vs 5KCV_2p

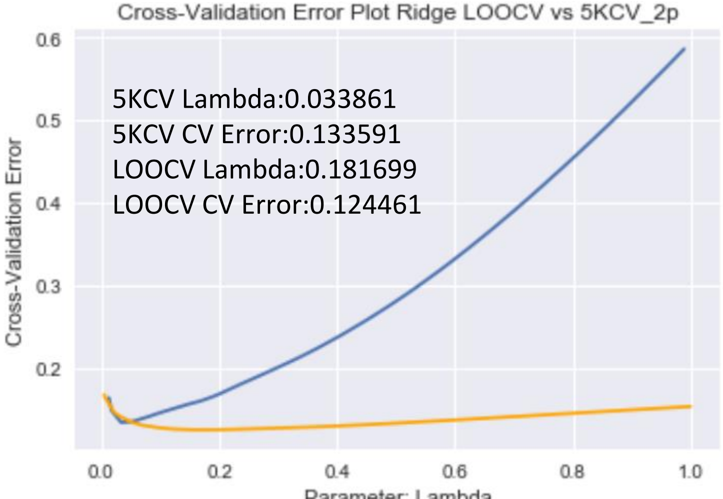
5KCV Lambda: 0.027161
5KCV CV Error: 0.118668
LOOCV Lambda: 0.027116
LOOCV CV Error: 0.126191



Cross-Validation Error Plot Lasso LOOCV vs 5KCV_2p

5KCV Lambda: 0.016496
5KCV CV Error: 0.115495
LOOCV Lambda: 0.010516
LOOCV CV Error: 0.135691

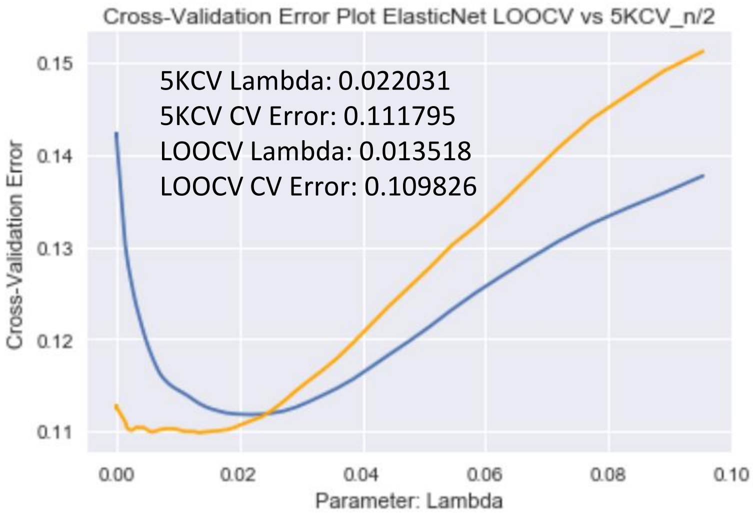
Blue: 5KCV
Orange: LOOCV



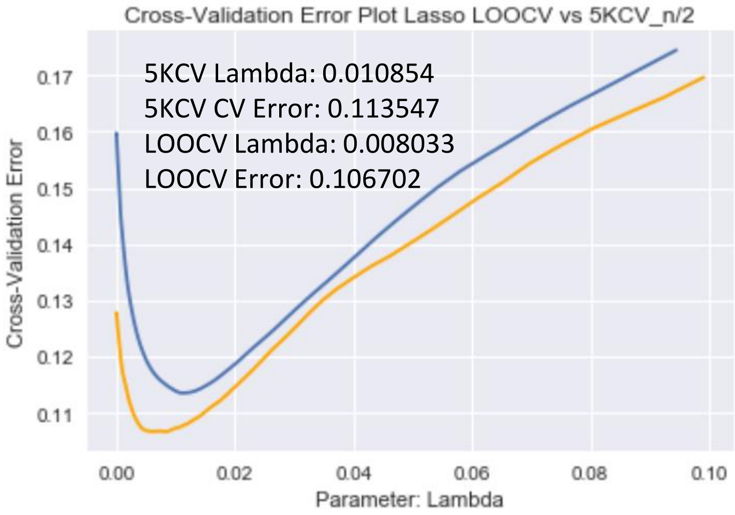
Cross-Validation Error Plot Ridge LOOCV vs 5KCV_2p

5KCV Lambda:0.033861
5KCV CV Error:0.133591
LOOCV Lambda:0.181699
LOOCV CV Error:0.124461

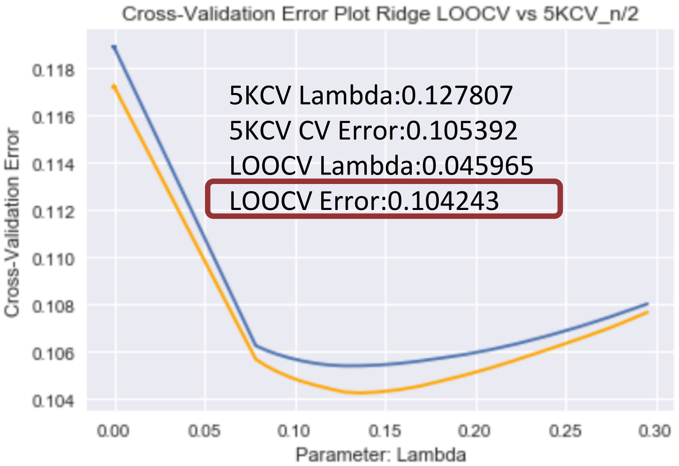
2
P



5KCV Lambda: 0.022031
5KCV CV Error: 0.111795
LOOCV Lambda: 0.013518
LOOCV CV Error: 0.109826



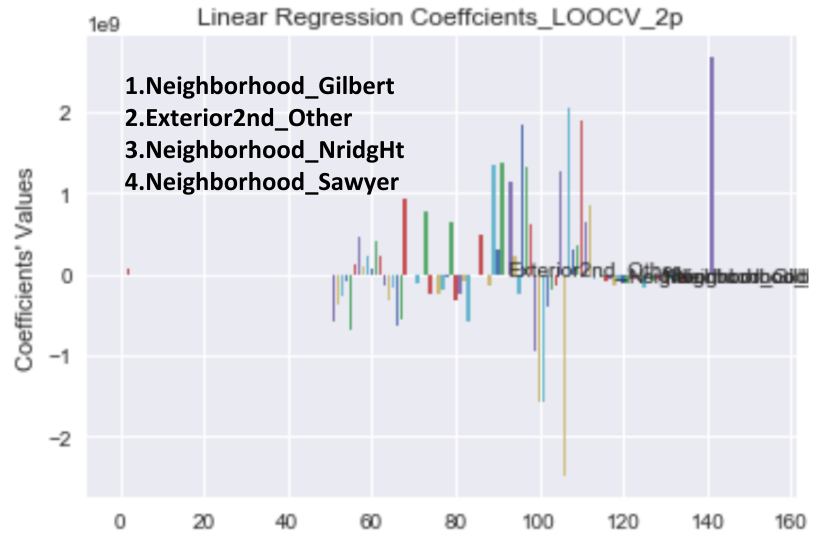
5KCV Lambda: 0.010854
5KCV CV Error: 0.113547
LOOCV Lambda: 0.008033
LOOCV Error: 0.106702



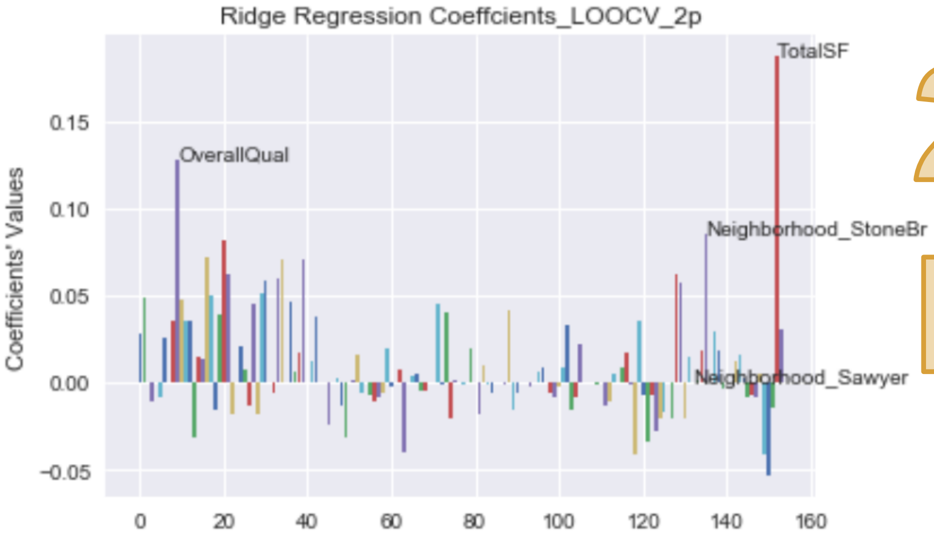
5KCV Lambda:0.127807
5KCV CV Error:0.105392
LOOCV Lambda:0.045965
LOOCV Error:0.104243

Blue: 5KCV
Orange: LOOCV

N
/
2

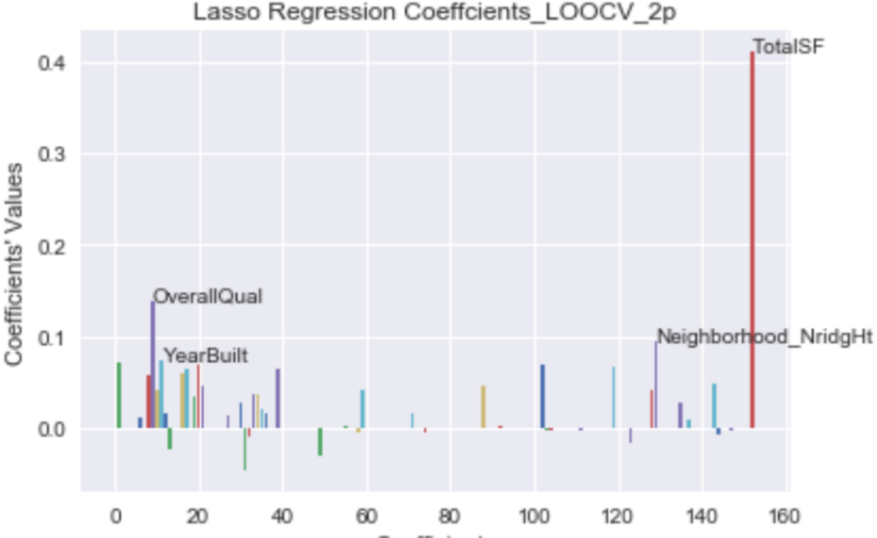


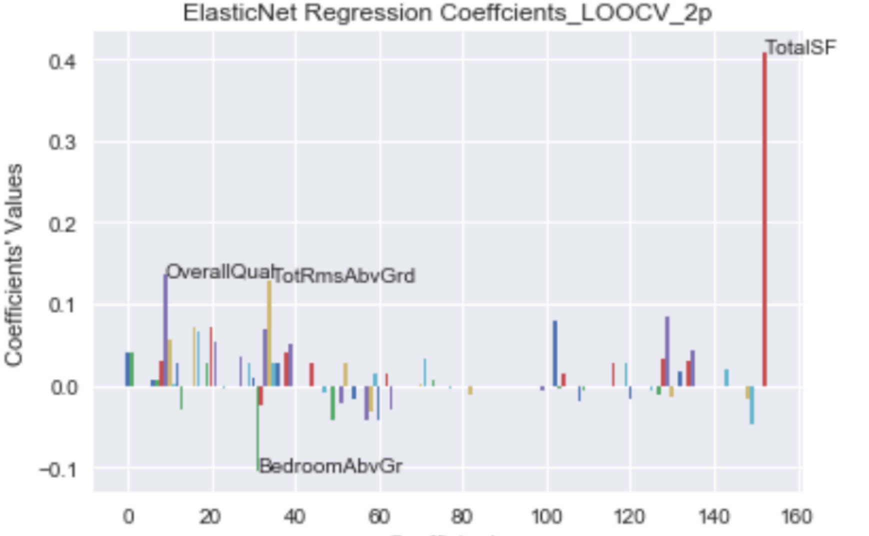
1.Neighborhood_Gilbert
2.Exterior2nd_Other
3.Neighborhood_NridgHt
4.Neighborhood_Sawyer

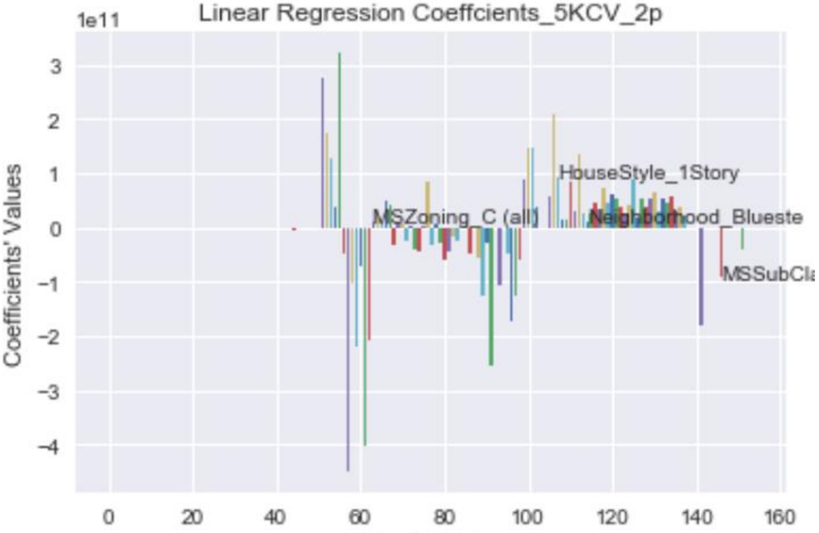


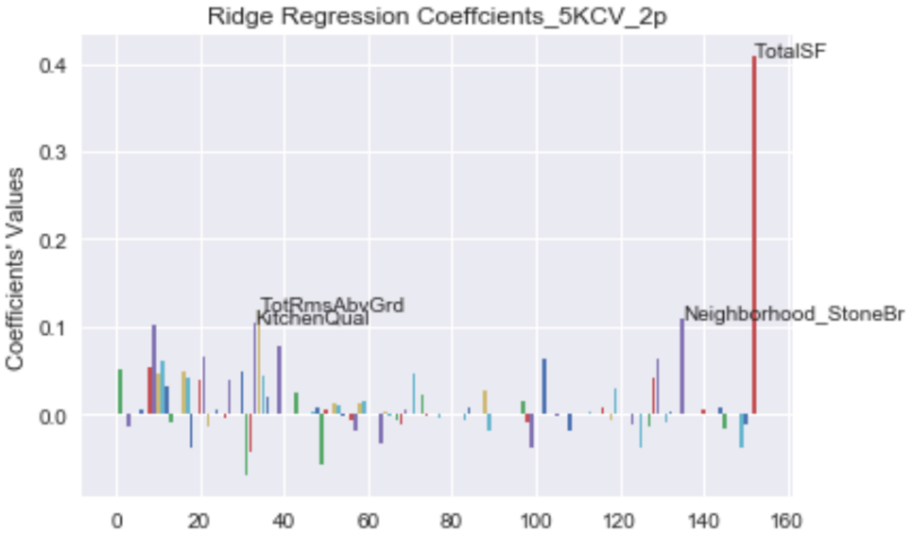
2P

LOOCV



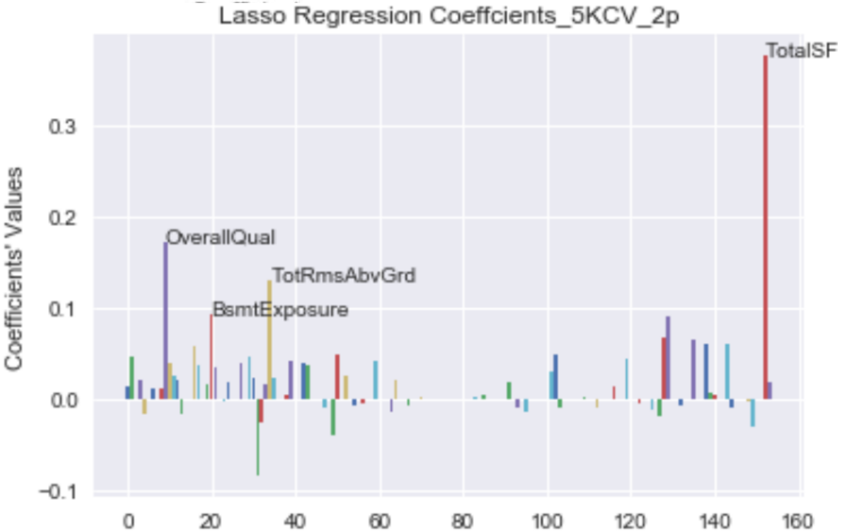


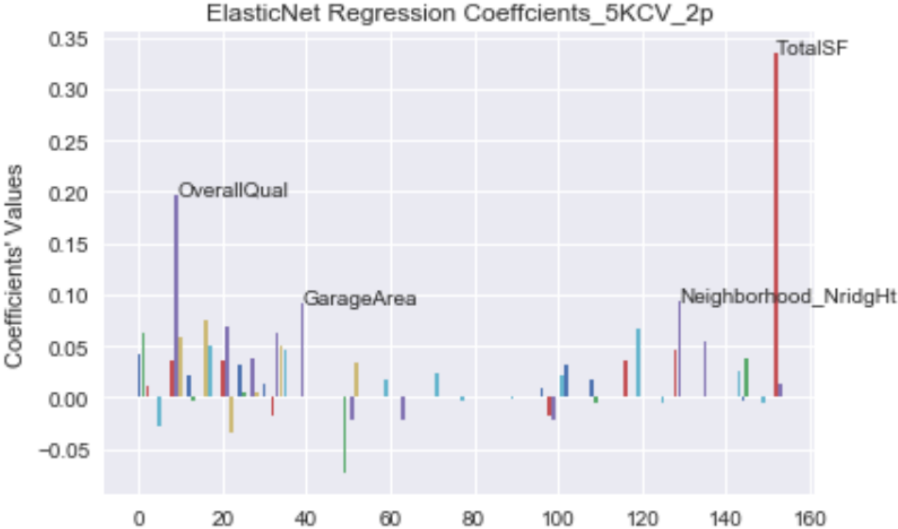

Linear Regression Coeffcients_5KCV_2p


Ridge Regression Coeffcients_5KCV_2p

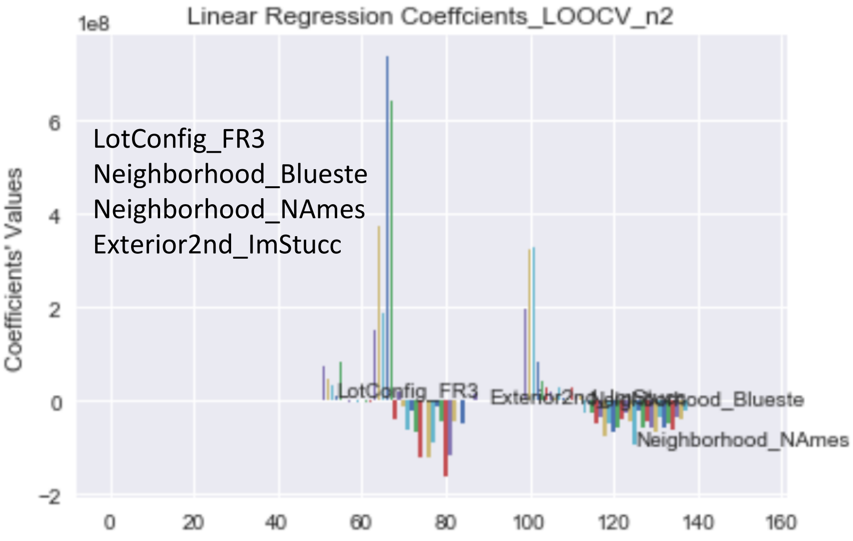
2P

5KCV

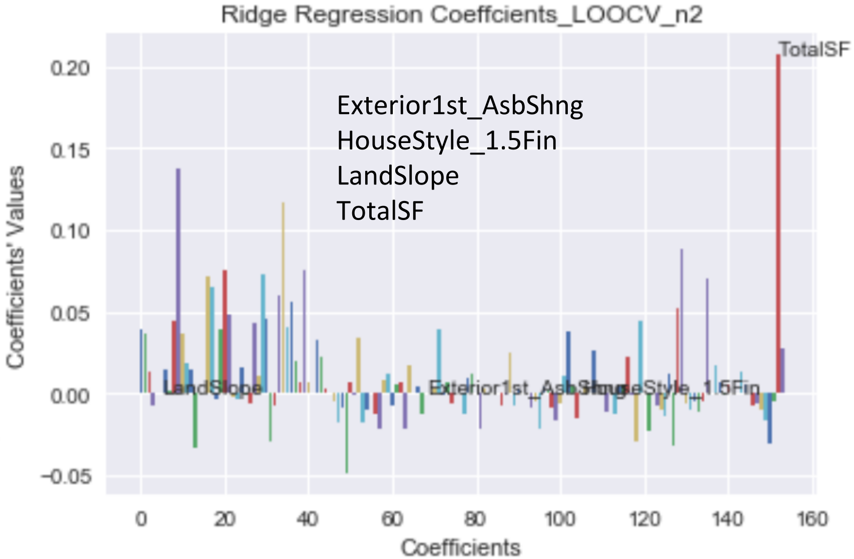

Lasso Regression Coeffcients_5KCV_2p


ElasticNet Regression Coeffcients_5KCV_2p

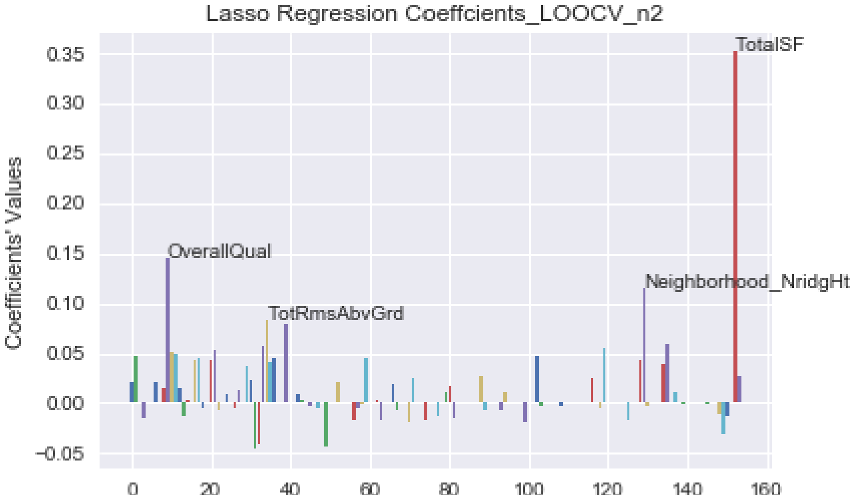
LOOCV

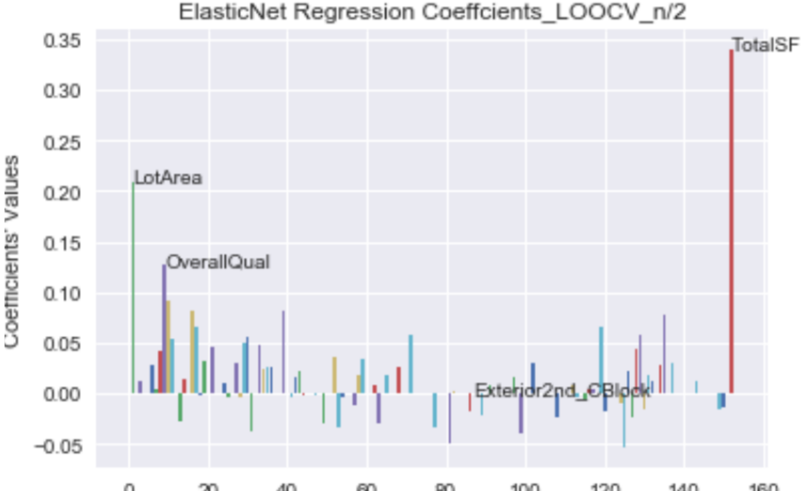

Linear Regression Coeffcients_LOOCV_n2

LotConfig_FR3
Neighborhood_Blueste
Neighborhood_NAmes
Exterior2nd_ImStucc


Ridge Regression Coeffcients_LOOCV_n2

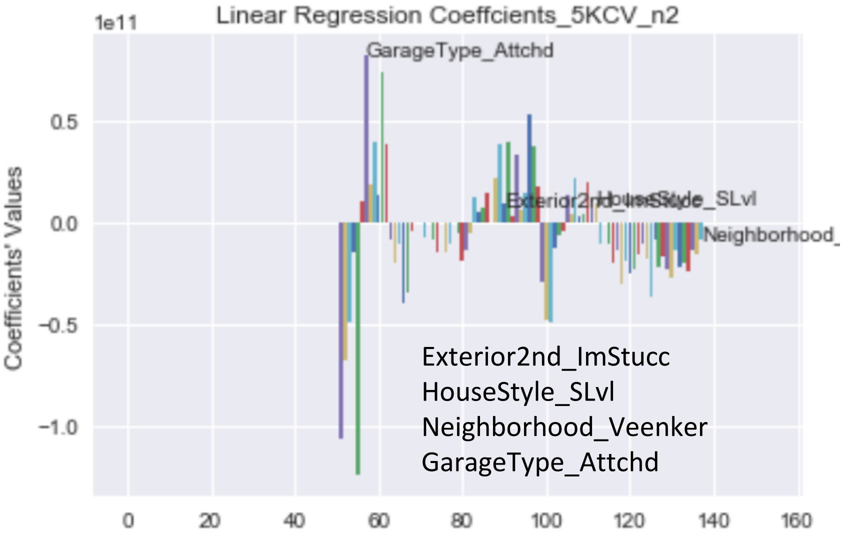
Exterior1st_AsbShng
HouseStyle_1.5Fin
LandSlope
TotalSF


Lasso Regression Coeffcients_LOOCV_n2

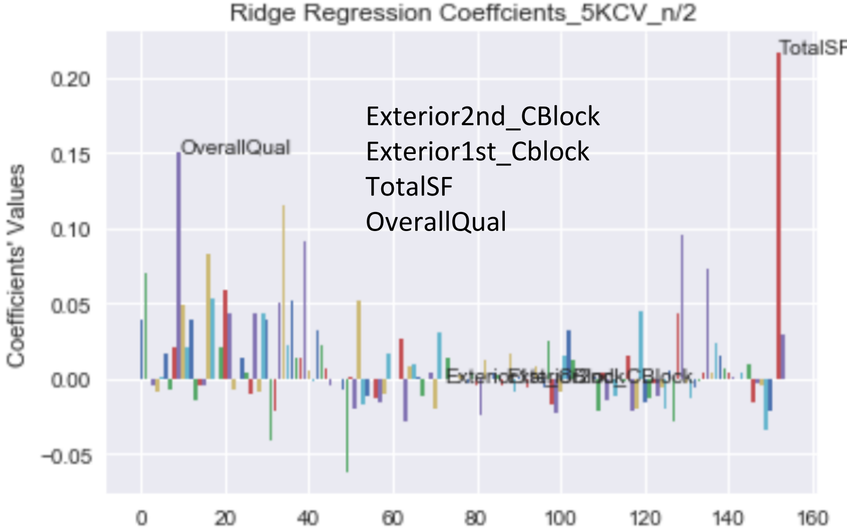

ElasticNet Regression Coeffcients_LOOCV_n/2

N/2

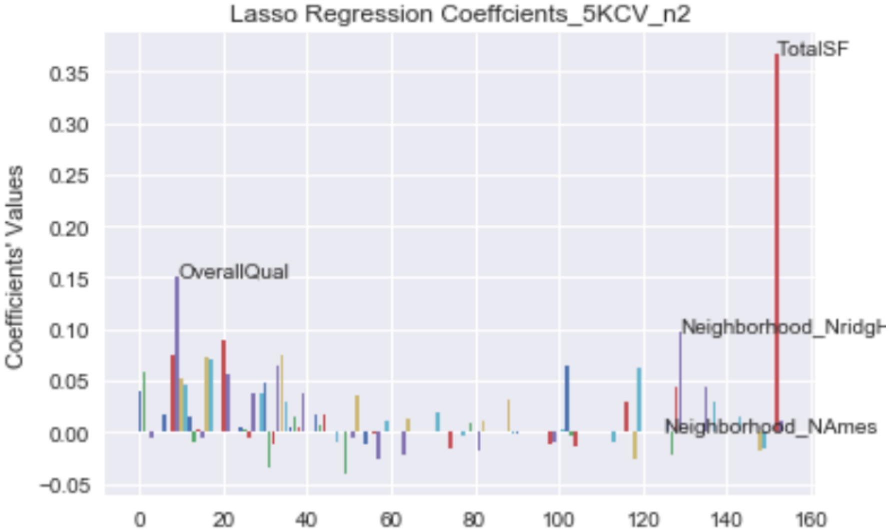
5KCV

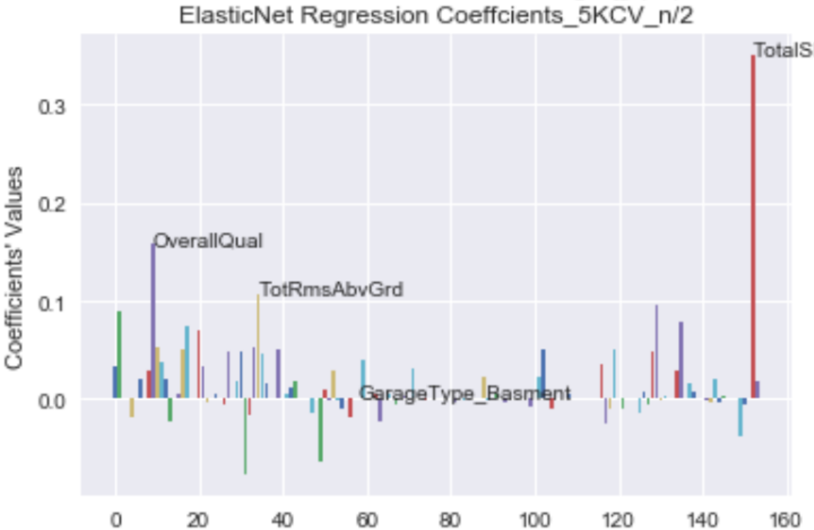

Linear Regression Coeffcients_5KCV_n2

Exterior2nd_ImStucc
HouseStyle_SLvl
Neighborhood_Veenker
GarageType_Attchd


Ridge Regression Coeffcients_5KCV_n/2

Exterior2nd_CBlock
Exterior1st_Cblock
TotalSF
OverallQual


Lasso Regression Coeffcients_5KCV_n2


ElasticNet Regression Coeffcients_5KCV_n/2

N
/
2

CONCLUSION:

R square:

1. LOOCV method has more variation than the 5 Fold method
2. Increasing sample size increases the testing R square but reduces training R square

Parameter Tuning:

1.LOOCV method leads to the lower CV error

2.Increasing sample size leads to lower CV error

Coefficients:

1. 5 Fold method shrinks more aggressively
2. Increasing sample size mitigate effect of regularizations

Top variables:

1.TotalSF : Total Square Feet

2.OverallQuality

3.Neighborhood_NridgHt (Northridge Heights)

4.Total Rooms Above Ground

5.GarageArea

THANKS FOR WATCHING

MICHELLE (QIN) PENG