Assignment-based Subjective Questions

(Completed by Prayag Sanjay)

Question 1

What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer

Following are output from the notebook of the optimal value for ridge and lasso regression.

Alpha for Ridge : 73.88

```
In [95]: # Print the best hyperparameter alpha

         print(model_cv.best_params_)

         {'alpha': 73.87755102040816}
```

Alpha For Lasso : 1000

```
In [152]: # Printing the best hyperparameter alpha
          print(model_cv.best_params_)

          {'alpha': 1000}
```

When alpha is doubled for Ridge to approx. to 146:

Following is change in values of various metrics for Ridge

	Metric	Alpha = 73	Alpha = 146
0	R2 Score (Train)	8.800000e-01	8.700000e-01
1	R2 Score (Test)	8.400000e-01	8.400000e-01
2	RSS (Train)	6.424967e+11	6.651588e+11
3	RSS (Test)	3.857951e+11	4.026306e+11
4	MSE (Train)	2.539837e+04	2.584241e+04
5	MSE (Test)	3.005830e+04	3.070715e+04

Following is change in the coefficients for Ridge:

	Alpha = 73	Alpha = 146
LotArea	8465.27	8517.30
BsmtFinSF1	10055.42	9597.65
TotalBsmtSF	11282.11	11752.58
1stFlrSF	16236.42	15317.33
2ndFlrSF	22601.86	21160.72
AgeHouse	-13339.15	-13466.02
OverallQual_Good	9718.89	8957.19
OverallQual_VeryGood	18241.77	17127.70
OverallQual_Excellent	18289.10	17141.16
OverallQual_10	12655.55	11824.67
Exterior1st_CemntBd	2886.99	2110.72
Exterior2nd_CmentBd	-1923.05	-800.55

When alpha is doubled for Lasso to approx. to 2000:

Following is change in values of various metrics for Lasso

	Metric	Alpha = 1000	Alpha = 2000
0	R2 Score (Train)	8.800000e-01	8.700000e-01
1	R2 Score (Test)	8.500000e-01	8.400000e-01
2	RSS (Train)	6.418111e+11	6.651588e+11
3	RSS (Test)	3.785877e+11	4.026306e+11
4	MSE (Train)	2.538481e+04	2.584241e+04
5	MSE (Test)	2.977620e+04	3.070715e+04

Following is change in the coefficients for Lasso

	Alpha = 1000	Alpha = 2000
LotArea	7806.07	8517.30
BsmtFinSF1	9953.11	9597.65
TotalBsmtSF	10440.66	11752.58
1stFlrSF	17625.87	15317.33
2ndFlrSF	23604.03	21160.72
AgeHouse	-13366.27	-13466.02
OverallQual_Good	9263.32	8957.19
OverallQual_VeryGood	18436.40	17127.70
OverallQual_Excellent	18586.74	17141.16
OverallQual_10	12650.29	11824.67
Exterior1st_CemntBd	107.80	2110.72
Exterior2nd_CmentBd	0.00	-800.55

Observations (Ridge):

1. R^2 score on train decreased from 0.88 to 0.87 while the R^2 score on test largely remained same when alpha was doubled.

2. MSE for train and test increased slightly when alpha was doubled.

3. RSS for train and test increased slightly when alpha was doubled.

4. Coefficient of some significant predictor such as **second floor area** decreased.

Observations (Lasso):

1. R^2 score on train and test decreased from 0.88 to 0.87 and 0.85 to 0.84 when alpha was doubled.

2. MSE for train and test increased slightly when alpha was doubled.

3. RSS for train and test increased slightly when alpha was doubled.

4. Coefficient of some significant predictor such as **second floor area** decreased.

After the change the most important predictor continue to remain the **second floor area** as it has the highest absolute value of the coefficient.

Question 2
You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answer:

Following are output from the notebook of the optimal value for ridge and lasso regression.

Alpha for Ridge : 73.88

In [95]:
```
# Print the best hyperparameter alpha

print(model_cv.best_params_)
```
```
{'alpha': 73.87755102040816}
```

Coefficients for Ridge

	Ridge
LotArea	8465.27
BsmtFinSF1	10055.42
TotalBsmtSF	11282.11
1stFlrSF	16236.42
2ndFlrSF	22601.86
AgeHouse	-13339.15
OverallQual_Good	9718.89
OverallQual_VeryGood	18241.77
OverallQual_Excellent	18289.10
OverallQual_10	12655.55
Exterior1st_CemntBd	2886.99
Exterior2nd_CmentBd	-1923.05

Alpha For Lasso : 1000

In [152]:
```
# Printing the best hyperparameter alpha
print(model_cv.best_params_)
```
```
{'alpha': 1000}
```

Coefficients for Lasso

	Lasso
LotArea	7806.07
BsmtFinSF1	9953.11
TotalBsmtSF	10440.66
1stFlrSF	17625.87
2ndFlrSF	23604.03
AgeHouse	-13366.27
OverallQual_Good	9263.32
OverallQual_VeryGood	18436.40
OverallQual_Excellent	18586.74
OverallQual_10	12650.29
Exterior1st_CemntBd	107.80
Exterior2nd_CmentBd	0.00

We can observe that Lasso has reduced the number features from 12 to 11 as compared to Ridge.

Since the R2 score for both of them are same as 0.84 on test data, using **Occam's Razor** directive of choosing a simpler model when all other things are equal, **we will choose Lasso model as it is simpler.**

Question 3

After building the model, you realized that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Answers : We created the model as shown below.

14. Excercise Question 3:

Dropping 5 important features of lasso and then building model again

[122]:
```
# get 5 top features of lasso by sorting on absolute value of coefficient

# add a abosLute value column
lasso_betas['abs'] = lasso_betas['Alpha = 1000'].abs()
```

[123]:
```
# sort the value by absolute value of the coefficient values (in this case 'Alpha' = 1000)

sorted_lasso = lasso_betas.sort_values(by='abs', ascending=False).head(5)
sorted_lasso
```

[123]:

	Alpha = 1000	Alpha = 2000	abs
2ndFlrSF	23604.03	21160.72	23604.03
OverallQual_Excellent	18586.74	17141.16	18586.74
OverallQual_VeryGood	18436.40	17127.70	18436.40
1stFlrSF	17625.87	15317.33	17625.87
AgeHouse	-13366.27	-13468.02	13366.27

[124]:
```
# we will drop the top five features

X_train_rfe = X_train_rfe.drop(sorted_lasso.index, axis=1)
```

In [124]:
```
# we will drop the top five features

X_train_rfe = X_train_rfe.drop(sorted_lasso.index, axis=1)
```


In [125]:
```
# list of alphas to tune -
# first we will tune apiha following approximately Logarithimc scale

params = {'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1,
    0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0,
    4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 500, 1000 ]}

lasso = Lasso()

model_cv = GridSearchCV(estimator = lasso,
                        param_grid = params,
                        scoring= 'neg_mean_absolute_error',
                        cv = folds,
                        return_train_score=True,
                        verbose = 1)


model_cv.fit(X_train_rfe, y_train)

print(model_cv.best_params_)
```

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
{'alpha': 500}
```

In [126]:
```
#Fitting Lasso model for alpha = 500 and printing coefficients

alpha = 500

lasso = Lasso(alpha=alpha)

# fit the model with alpha 500

lasso.fit(X_train_rfe, y_train)

lasso_betas = pd.DataFrame(index=X_train_rfe.columns)

# we will get top 5 features
lasso_betas['Lasso'] = np.round(lasso.coef_,2)
lasso_betas.sort_values(by='Lasso', ascending=False)
```

Top 5 features now are :

Out[126]:

	Lasso
TotalBsmtSF	34037.08
LotArea	13668.31
OverallQual_Good	12117.09
OverallQual_10	10757.28
BsmtFinSF1	10015.99
Exterior1st_CemntBd	6249.84
Exterior2nd_CmentBd	0.00

i.e., Total Basement Area, Lot Area, Overall condition (Good), Overall Quality (Excellent), Basement Type I finish area

Question 4

How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

Answer :

A model is said to be **generalizable** when it performs better on unseen data. **We can make a model generalizable by making it simple.**

A model is said to be more robust when it does not change significantly if the training data points undergo small changes. In the other words, the robust model is the one, the testing error of which is close to the training error.

We can make robust and generalizable by making it simple but no simpler than it need to be. i.e., find that tradeoff between bias and variance. Implication of this is that bias will increase but that is small price to pay for large benefit in robustness and generalization.