ADVANCED REGRESSION SUBJECTIVE QUESTIONS

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:

We created our model using Ridge and Lasso Regression Methods with the help of Negative Mean Absolute Error and R2 square and arrived at the following best alpha for both the regression techniques.

Ridge Alpha: 10

Lasso Alpha: 0.001

From the above best alpha we had derived the following best features for both the methods.

SL.NO	Ridge Features	Coefficients
9	GrLivArea	0.147724
0	LotFrontAge	0.091727
8	2ndFlrSF	0.073740
3	YearRemodAdd	0.069124
6	1stFlrSF	0.061069
7	CentralAir	0.058608
2	YearBuilt	0.050184
4	MasVnrArea	0.042278
5	TotalBsmtSF	0.009447
1	LotArea	- 0.010729

SL.NO	Lasso Features	Coefficients
3	YearRemodAdd	0.135826
9	GrLivArea	0.104112
4	MasVnrArea	0.077311
6	CentralAir	0.069702
7	1stFlrSF	0.068973
8	2ndFlrSF	0.068100
2	YearBuilt	0.050753
0	LotFrontAge	0.046765
5	TotalBsmtSF	0.012624
1	LotArea	-0.010894

As per the question when we double the alpha values, we get the following changes.

Ridge Alpha: 20

Lasso Alpha: 0.002

(The code implementations are done in the Jupyter Notebook.)

After doubling the alpha values for both Ridge and Lasso we can see there is not much a difference in the R2 square values.

```
: # ridge regression model with alpha 20

new_ridge = Ridge(alpha=20)
new_ridge.fit(X_train,y_train)
# r2 score for train set
y_pred_train = new_ridge.predict(X_train)
print('r2 Score for Train is ',r2_score(y_train,y_pre
#r2 score for test set
y_pred_test = new_ridge.predict(X_test)
print('r2 Score for test is ',r2_score(y_test,y_pred_
```

```
#rebuilding lasso regression model with alpha 0.002
lasso_new = Lasso(alpha=0.002)
lasso_new.fit(X_train,y_train)
y_train_pred_new1 = lasso_new.predict(X_train)
y_test_pred_new1 = lasso_new.predict(X_test)
print('r2 Score for Train set',r2_score(y_true=y_train,y_pred=y_tr
print('r2 score for Test set',r2_score(y_true=y_test,y_pred=y_test
```

```
r2 Score for Train set 0.90234690911789
```

The top features we got after doubling the ridge and lasso alpha values are as follows:

SL.NO	New Ridge Features	Coefficients
9	GrLivArea	0.135826
0	LotFrontAge	0.104112
8	2ndFlrSF	0.077311
3	YearRemodAdd	0.069702
6	CentralAir	0.068973
7	1stFlrSF	0.068100
2	YearBuilt	0.050753
4	MasVnrArea	0.046765
5	TotalBsmtSF	0.012624
1	LotArea	- 0.010894

SL.NO	New Lasso Features	Coefficients
3	YearRemodAdd	0.121152
9	MasVnrArea	0.091519
4	CentralAir	0.074965
6	1stFlrSF	0.061901
7	GrLivArea	0.059103
8	YearBuilt	0.045717
2	2ndFlrSF	0.011501
0	TotalBsmtSF	0.009684
5	LotArea	-0.014352
1	LotFrontage	-0.046562

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?

As per my model the optimal value of lambda for Ridge and Lasso Regression are:

Ridge – 10

Lasso – 0.001

I shall choose Lasso lambda value to apply for value for the following reasons:

- The most important advantage of using the Lasso Regression instead of the Ridge regression is that Lasso Regression trims down the coefficients of redundant variables to zero, which in turn performs Feature selection whereas Ridge just reduces the coefficients value to a very low value and not zero.
- In our current data, we have more than 190 features and since we want to pick up only those features which the most useful here Lasso Regression is helpful as it produces sparse solutions.
- Lasso Regression also helps to determine the features which have a direct impact to the target variable.

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?

Answer:

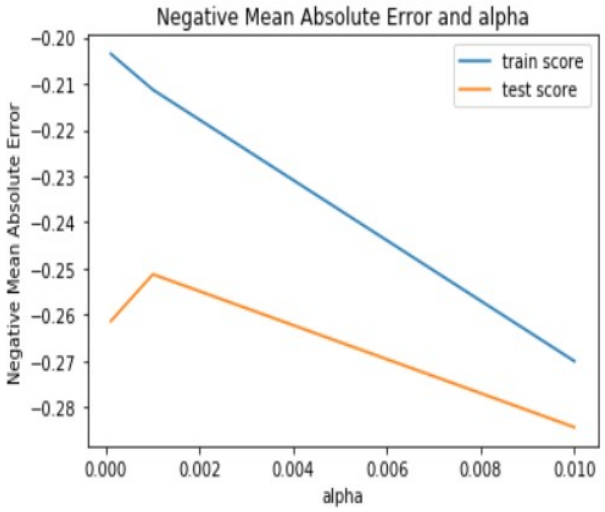
The steps we follow to build a model with 5 different predictor variables are:

- First we drop the 5 top variables from the previous model and create a new data frame
- Next we split the new data frame into test and train set (0.3, 0.7 respectively).
- After splitting the data we scale both the train and test data using Standard Scaler.

- Defining the X and y train and test sets.
- As the question asks for top 5 predictor variable we shall proceed with Lasso Regression Technique at arrive at the same.
- In Lasso Regression we use Negative Mean Absolute Error
- Next we find out the best score, R2 Score and other important parameters.
- Lastly we find the top most predictor variables.

Please find all the above mentioned steps in the Jupyter Notebook.

```
<matplotlib.legend.Legend at 0x18ce70e31c8>
```




```
lasso_new_modified = Lasso(alpha=0.001)
lasso_new_modified.fit(X_train_new,y_train_new)

y_train_new_pred = lasso_new_modified.predict(X_train_new)
y_test_new_pred = lasso_new_modified.predict(X_test_new)

print('r2 Score for Train:',r2_score(y_true=y_train_new,y_pred=y_tr
print('r2 Score for Test:',r2_score(y_true=y_test_new,y_pred=y_test
```

```
# Printing the best score and optimum value of alpha
print(lasso_cv_model_modified.best_estimator_)
print('Best alpha value:',lasso_cv_model_modified.be
```

The Top features derived by the modified changes are:

```
lasso_coef[:10].sort_values(by='Coefficient',ascendi
```

	Features	Coefficient
3	TotalBsmtSF	0.131012
9	SalePrice	0.123107
4	2ndFlrSF	0.078319
6	WoodDeckSF	0.054719
7	OpenPorchSF	0.052679
8	YrSold	0.049002
2	YearBuilt	0.046462
0	LotFrontage	0.032802

Question 4:

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

Answer:

A model is said to be more robust and generalisable when the model is Simple. The benefits of simple models are:

- Simple models are generic in nature when compared with complex models, and we can say that generic models usually perform better on the unseen data than the complex models.
- As in real world problems, most of the time there isn't enough data to create our model so such cases simple models works best as it requires less training data.
- If the training data undergoes any small changes, Simple models perform better and are more robust and do not change significantly.
- A simple model always outperforms on the new data and always makes more error while training the data.

Implications of Accuracy:

- Simple model has low variance and high bias.
- Bias quantifies how accurate is the model likely to be on the test data set.
- While on the other hand, if in case there in enough quantity of training data, a complex model outperforms and accurately predicts.

To make our models more robust and generalisable we need to use Regularisation Method to strike a balance between accuracy and model complexity.