Advanced Regression Assignment

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 have performed the Ridge and Lasso regression using both Negative Mean Absolute error and R2 score.
From the both the methods, We got the following alpha optimal values:

Alpha = 10 (Ridge)
Alpha = 0.001(Lasso)

From the above, the following features are identified:(using lasso)

Features	Coefficient
YearRemodAdd	0.131012
GrLivArea	0.123107
MasVnrArea	0.078319
CentralAir	0.054719
1stFlrSF	0.052679
2ndFlrSF	0.049002
YearBuilt	0.046462
LotFrontage	0.032802
TotalBsmtSF	0.006307
LotArea	-0.014317

As per the statement in the given question, now we are Implementing double the value of alpha for both models, hence the alpha values will be the following for the Ridge and Lasso and the predictions will be as follows:

Alpha = 20 (Ridge)
Alpha = 0.002 (Lasso)

The code implementations are done in the python notebook. Please refer there.

Considering the above, we are proceeding with alpha = 20 and continued with Ridge regression

```
# rebuilding the Ridge Regression, with alpha = 20
ridge_new = Ridge(alpha = 20)
ridge_new.fit(X_train,y_train)
# r2 score for train set
y_pred_train_new = ridge_new.predict(X_train)
print('r2 Score for Train is ',r2_score(y_train,y_pred_train_new))
#r2 score for test set
y_pred_test_new = ridge_new.predict(X_test)
print('r2 Score for test is ',r2_score(y_test,y_pred_test_new))
```

```
r2 Score for Train is  0.9012229283354803
r2 Score for test is  0.8632344435896999
```

```
model_parameter_new = list(ridge_new.coef_)
model_parameter_new.insert(0,ridge_new.intercept_)
cols = housePrice_df.columns
cols.insert(0,'constant')
ridge_coef_new = pd.DataFrame(list(zip(cols,model_parameter_new)))
ridge_coef_new.columns = ['Features','Coefficient']
```

```
ridge_coef_new[:10].sort_values(by='Coefficient',ascending=False)#.head(
```

	Features	Coefficient
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

After implementing the double the values of alpha, the features are changed to the following:(lasso)

Features	Coefficient
GrLivArea	0.135826
LotFrontage	0.104112
2ndFlrSF	0.077311
YearRemodAdd	0.069702
CentralAir	0.068973
1stFlrSF	0.068100
YearBuilt	0.050753
MasVnrArea	0.046765
TotalBsmtSF	0.012624
LotArea	-0.010894

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:

From the model built using Ridge and lasso regression we got the following optimal values of alpha:

Alpha = 10 (Ridge)

Alpha = 0.001 (Lasso)

Now, I will apply the Lasso Regression model with alpha value of 0.001.

The main intention of using Lasso Regression over the Ridge regression is, Lasso Regression performs the Feature selection by reducing some of the feature coefficients to zero.

Lasso Regression is mainly used to determine the features which will impact the target variable.

In this present model, we have more than 180 features are present in the data set, where Lasso Regression will reduce the number of features which are significant in determining the target variable(Sale Price)

```
model_param = list(lasso.coef_)
model_param.insert(0,lasso.intercept_)
cols = housePrice_df.columns
cols.insert(0,'const')
lasso_coef = pd.DataFrame(list(zip(cols,model_param)))
lasso_coef.columns = ['Features','Coef']
lasso_coef[:10].sort_values(by='Coef',ascending=False)
```

	Features	Coef
3	YearRemodAdd	0.131012
9	GrLivArea	0.123107
4	MasVnrArea	0.078319
6	CentralAir	0.054719
7	1stFlrSF	0.052679
8	2ndFlrSF	0.049002
2	YearBuilt	0.046462
0	LotFrontage	0.032802
5	TotalBsmtSF	0.006307
1	LotArea	-0.014317

Question - 3

After building the model, you realised 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?

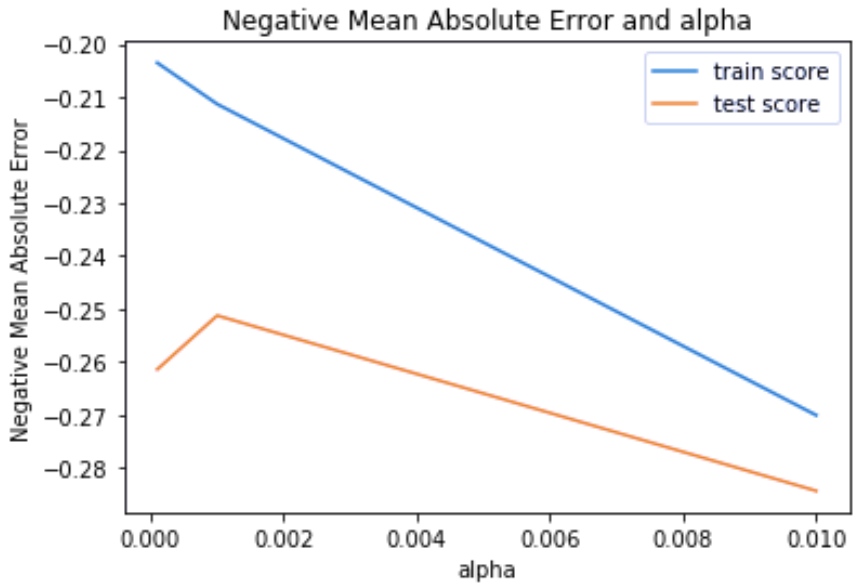
Answer:

First, we will drop the variables which are top 5 variables and we will create a new data frame.

Steps:

1. Splitting the data into train and test sets at 70 - 30 ratio
2. Scaling the variables using Standard scalar
3. Defining X,y Test and train sets
4. Performing the Lasso Regression using negative absolute mean error
5. Finding out the best score, r2 score and other important parameters
6. Finding out the modified top variables

All the steps performed in python notebook. Please refer.




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

```
Lasso(alpha=0.001, copy_X=True, fit_intercept=True, max_iter=1000,
      normalize=False, positive=False, precompute=False, random_state=None,
      selection='cyclic', tol=0.0001, warm_start=False)
Best alpha value: {'alpha': 0.001}
Best score: -0.2512201869102882
```

```
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_train_new_pred))
print('r2 Score for Test:',r2_score(y_true=y_test_new,y_pred=y_test_new_pred))
```

```
r2 Score for Train: 0.9091718231774004
r2 Score for Test: 0.8616221073272513
```

```
model_param = list(lasso.coef_)
model_param.insert(0,lasso.intercept_)
cols = housePrice_df_new.columns
cols.insert(0,'const')
lasso_coef = pd.DataFrame(list(zip(cols,model_param)))
lasso_coef.columns = ['Features','Coefficient']
```

```
lasso_coef[:10].sort_values(by='Coefficient',ascending=False)
```

	Features	Coefficient
3	TotalBsmtSF	0.131012

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:
Usually, when we run the algorithm against test data if there the algorithm does not deteriorate too much between the test and train data it shows the robustness of the algorithm.
In the current model we built, there is slight difference between the test and train data hence this model will be called robustness and generalised version of the model.
Further when we compare the error terms , r2 score and negative mean absolute error there is slight difference between the test and train data.

If the algorithm has too much difference / zero difference between the test and train data, hence it prone to over fitting / under fitting of the model. When the model prone to over fitting / under fitting the accuracy will be either less or 100% which makes an algorithm not generalised and not useful to the different data sets.
When the algorithm is performing well at both test and train data sets, it will be having good accuracy over the model and the accuracy contained to the other data also.

By:
Venkata Nagarjuna T H