Assignment Part – II

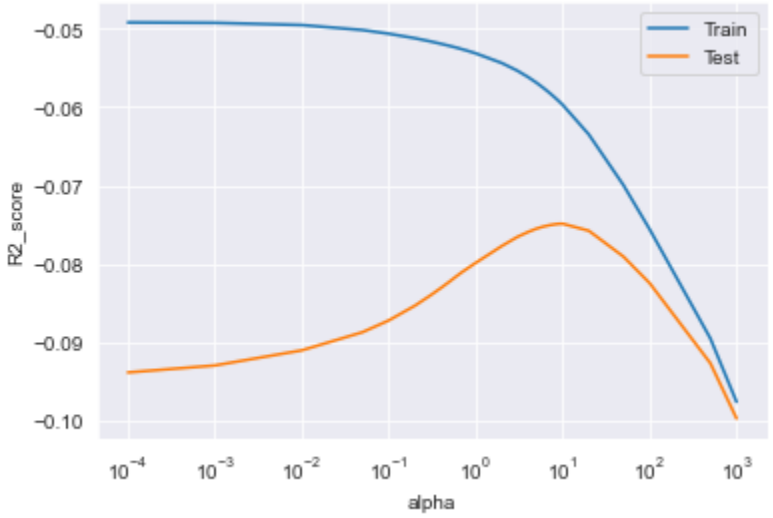
Question 1: What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer: -

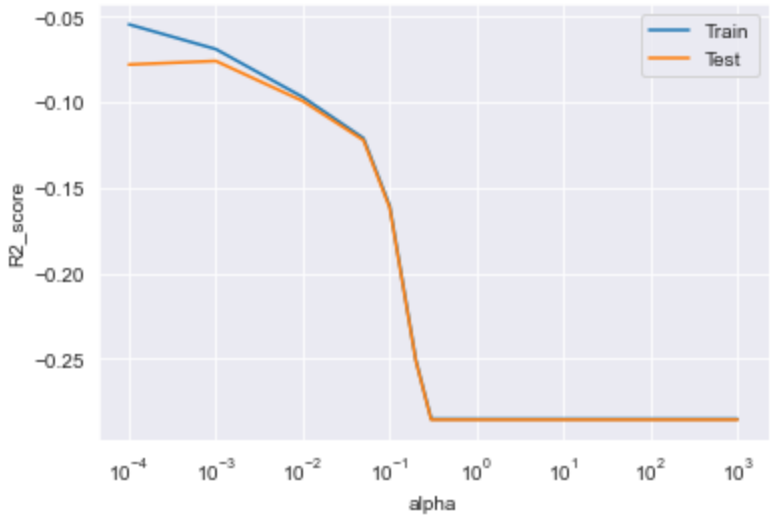
Optimal value of alpha for ridge and lasso regression are as follows: -

- Optimal value of lambda for Ridge Regression = 10
- Optimal value of lambda for Lasso = 0.001

a) **For ridge regression: -** As the value of alpha increases, we see a decrease in train error and an initial increase followed by decrease in test error.



b) **For lasso regression:** As the value of alpha increases, we see a decrease in both train and test error.



Changes in the model if you choose double the value of alpha for both ridge and lasso:-

For ridge regression:-

```
## Let us build the ridge regression model with double value of alpha i.e. 20
ridge = Ridge(alpha=20)

# Fit the model on training data
ridge.fit(X_train, y_train)
```

```
Ridge(alpha=20)
```

```
## Make predictions
y_train_pred = ridge.predict(X_train)
y_pred = ridge.predict(X_test)
```

```
## Check metrics
ridge_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R-Squared (Train) = 0.93
R-Squared (Test) = 0.93
RSS (Train) = 9.37
RSS (Test) = 2.82
MSE (Train) = 0.01
MSE (Test) = 0.01
RMSE (Train) = 0.09
RMSE (Test) = 0.10
```

For Lasso regression: -

```
## Now we will build the lasso model with double value of alpha i.e. 0.002
lasso = Lasso(alpha=0.002)

# Fit the model on training data
lasso.fit(X_train, y_train)
```

```
Lasso(alpha=0.002)
```

```
## Make predictions
y_train_pred = lasso.predict(X_train)
y_pred = lasso.predict(X_test)
```

```
## Make predictions
y_train_pred = lasso.predict(X_train)
y_pred = lasso.predict(X_test)
```

```
## Check metrics
lasso_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R-Squared (Train) = 0.91
R-Squared (Test) = 0.91
RSS (Train) = 13.49
RSS (Test) = 3.45
MSE (Train) = 0.01
MSE (Test) = 0.01
RMSE (Train) = 0.11
RMSE (Test) = 0.11
```

```
# Again creating a table which contain all the metrics

lr_table = {'Metric': ['R2 Score (Train)','R2 Score (Test)','RSS (Train)','RSS (Test)',
                       'MSE (Train)','MSE (Test)', 'RMSE (Train)', 'RMSE (Test)'],
            'Ridge Regression' : ridge_metrics,
            'Lasso Regression' : lasso_metrics
        }

final_metric = pd.DataFrame(lr_table, columns = ['Metric', 'Ridge Regression', 'Lasso Regression'] )
final_metric.set_index('Metric')
```

	Ridge Regression	Lasso Regression
Metric		
R2 Score (Train)	0.93	0.91
R2 Score (Test)	0.93	0.91
RSS (Train)	9.37	13.49
RSS (Test)	2.82	3.45
MSE (Train)	0.01	0.01
MSE (Test)	0.01	0.01
RMSE (Train)	0.09	0.11
RMSE (Test)	0.10	0.11

Changes in Ridge Regression metrics:

- R2 score of train set decreased from 0.94 to 0.93
- R2 score of test set remained same at 0.93

Changes in Lasso metrics:

- R2 score of train set decreased from 0.92 to 0.91
- R2 score of test set decreased from 0.93 to 0.91

The most important variable after the changes has been implemented: -

For ridge regression: -

```
## View the top 10 coefficients of Ridge regression in descending order
betas['Ridge'].sort_values(ascending=False)[:10]
```

```
GrLivArea               0.08
OverallQual_8           0.07
OverallQual_9           0.06
Neighborhood_Crawfor    0.06
Functional_Typ          0.06
Exterior1st_BrkFace     0.06
OverallCond_9           0.05
TotalBsmtSF             0.05
CentralAir_Y            0.05
OverallCond_7           0.04
Name: Ridge, dtype: float64
```

For Lasso regression:-

```
## View the top 10 coefficients of Lasso in descending order
betas['Lasso'].sort_values(ascending=False)[:10]
```

```
GrLivArea               0.11
OverallQual_8           0.08
OverallQual_9           0.08
Functional_Typ          0.07
Neighborhood_Crawfor    0.07
TotalBsmtSF             0.05
Exterior1st_BrkFace     0.04
CentralAir_Y            0.04
YearRemodAdd            0.04
Condition1_Norm         0.03
Name: Lasso, dtype: float64
```

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: -

The model we will choose to apply will depend on the use case.

Use Case 1 - If we have too many variables and one of our primary goals is feature selection

Then we will use Lasso.

Use Case 2 - If we don't want to get too large coefficients and reduction of coefficient magnitude is one of our prime goals

then we will use Ridge Regression

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:-

Here, we will drop the top 5 features in Lasso model and build the model again.

We will follow below steps to get new top 5 predictors: -

a) Create a list of top 5 lasso predictors that are to be removed

b) Drop them from train and test data

c)Now to create a Lasso model, we will run a cross validation on a list of alphas to find the optimum value of alpha.

d)We will get the optimum value of alpha, and we will build a lasso regression model using this value

e) Now, we will look at the top 5 features significant in predicting the value of a house according to the new lasso model

After dropping our top 5 lasso predictors, we get the following new top 5 predictors: -

a)2ndFlrSF

b)Functional_Typ

c)1stFlrSF

d)MSSubClass_70

e)Neighborhood_Somerst

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 robust when any variation in the data does not affect its performance much. A generalizable model is able to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model. To ensure a model is robust and generalizable, we have to take care it doesn't overfit. This is because an overfitting model has very high variance and any smallest change in data affects the model prediction heavily. Such a model will identify all the patterns of a training data but fails to pick up the patterns in unseen test data.

In other words, the model should not be too complex to be robust and generalizable.

If we look at it from the perspective of Accuracy, a too complex model will have a very high accuracy. So, to make our model more robust and generalizable, we will have to decrease variance which will lead to some bias. Addition of bias means that accuracy will decrease.

In general, we must strike some balance between model accuracy and complexity. This can be achieved by Regularization techniques like Ridge Regression and Lasso.