Assignment Part – II

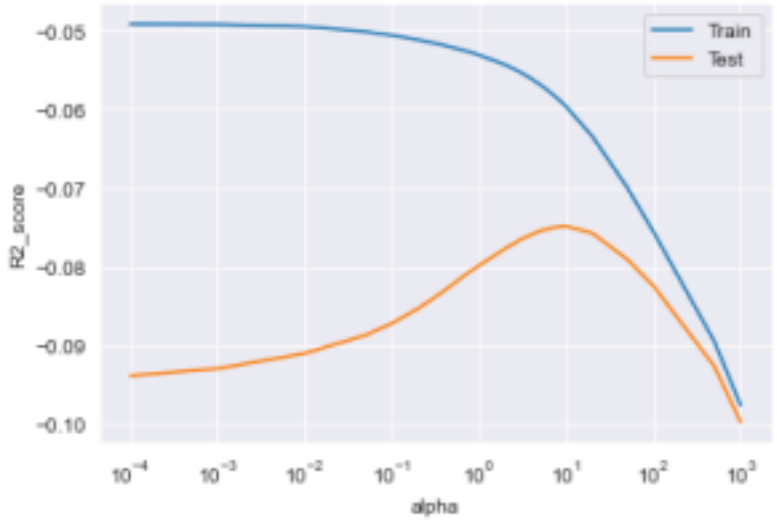
Question 1: What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer: -

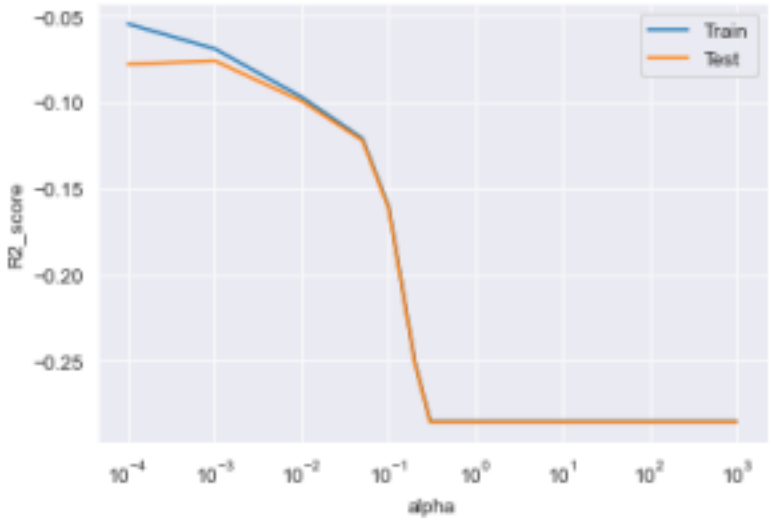
For ridge and lasso regression, the optimal alpha value is as follows:

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

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



b) **For lasso regression:** We witness a drop in both train and test error as the value of alpha grows.



Changes in the model if the alpha value for both the ridge and the lasso is doubled:

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
```

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')
```

Metric	Ridge Regression	Lasso Regression
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

Ridge metrics changes:

- R2 score of the train set has fallen from 0.94 to 0.93,
- R2 score of the test set has remained constant at 0.93.

Lasso metrics changes:

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

After the adjustments/changes have been implemented, the most crucial variable is:

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
```

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 employ will be determined by the use case.

Case 1 - If we have too many variables and feature selection is one of our key aims, we will employ Lasso.

Case 2 - If we don't want to gain too many huge coefficients and one of our primary aims is to reduce the magnitude of the coefficients, we'll employ 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 remove the top five characteristics from the Lasso model and rebuild it.

To obtain new top 5 predictors, we will do the following steps: -

a) Make a list of the top five lasso predictors to be deleted.

b) Remove them from the train and test data.

c)To develop a Lasso model, we will perform cross validation on a set of alphas to determine the best alpha value.

d)We will determine the optimal alpha value and use it to design a lasso regression model.

e) Now, we'll look at the top five qualities that are important in forecasting the value of a house, according to the fresh lasso design

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 considered resilient when any variation in the data has little effect on its performance. A generalizable model is capable of adapting to new, previously unseen data collected from the same distribution as the one used to generate the model. To ensure that a model is resilient and generalizable, we must ensure that it does not overfit. This is because an overfitting model has a very high variance, and even the smallest change in data has a large impact on model prediction. Such a model will recognize all of the patterns in training data but will miss patterns in unknown test data.

In other words, the model should not be overly complicated in order to be robust and generalizable.

From the standpoint of accuracy, an overly complex model will have a very high accuracy. As a result, in order to make our model more robust and generalizable, we must reduce variance, which will result in some bias. When bias is introduced, accuracy suffers.

In general, we must find a happy medium between model correctness and complexity. Regularization techniques such as Ridge Regression and Lasso can help achieve this.