Ridge-Lasso Regression

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 to 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 regression: 8

```
alpha = model_cv.best_params_.get('alpha')
ridge = Ridge(alpha=alpha)

ridge.fit(X_train, y_train)
```

```
Ridge(alpha=8.0)
```

```
#predict y values using ridge model
y_pred_train = ridge.predict(X_train)
y_pred_test = ridge.predict(X_test)
```

```
ridge_metric_scores = metric_scores(y_train, y_pred_train,y_test, y_pred_test)
```

```
r2_train score:  0.8976526798659606
r2_test score:  0.8319975928856693
rss_train score:  1.357070463391783
rss_test score:  0.7341294037936948
mse_train score:  0.0013317668924355084
mse_test score:  0.0016799299857979287
rmse_train score:  0.036493381488093266
rmse_test score:  0.040986948969128315
```

When the alpha is doubled for ridge model i.e., changed to alpha=16,

R2 score reduces slightly. Whereas RMSE score increases.

R2_train changes from 0.8976 to 0.8852
R2_test changes from 0.8319 to 0.8255

```
alpha = 16 #model_cv.best_params_.get('alpha')
ridge = Ridge(alpha=alpha)

ridge.fit(X_train, y_train)
```

```
Ridge(alpha=16)
```

```
#predict y values using ridge model
y_pred_train = ridge.predict(X_train)
y_pred_test = ridge.predict(X_test)
```

```
ridge_metric_scores = metric_scores(y_train, y_pred_train,y_test, y_pred_test)
```

```
r2_train score:  0.8852707497080105
r2_test score:  0.825518249158449
rss_train score:  1.521248203220514
rss_test score:  0.7624425501892798
mse_train score:  0.0014928834182733321
mse_test score:  0.0017447197944834777
rmse_train score:  0.03863784955549831
rmse_test score:  0.041769843122562454
```

➔ Optimal value of alpha for lasso regression: 0.0001

```
[829]
alpha = model_cv.best_params_.get('alpha')

lasso = Lasso(alpha=alpha)
lasso.fit(X_train, y_train)
```

```
Lasso(alpha=0.0001)
```

```
[830]
#predict y values using lasso model
y_pred_train = lasso.predict(X_train)
y_pred_test = lasso.predict(X_test)
```

```
lasso_metric_scores = metric_scores(y_train, y_pred_train,y_test, y_pred_test)
```

```
r2_train score:  0.9067832745100263
r2_test score:  0.8310716983393773
rss_train score:  1.2360036852051441
rss_test score:  0.73817533636645
mse_train score:  0.0012129574928411622
mse_test score:  0.001689188412737872
rmse_train score:  0.03482753928776999
rmse_test score:  0.04109973738040028
```

When the alpha is doubled for lasso i.e., changed to alpha=0.0002,

R2 score reduces slightly. Whereas RMSE score increases.

R2_train changes from 0.9067 to 0.8989
R2_test changes from 0.8310 to 0.8292

```
[826]
alpha = 0.0002 #model_cv.best_params_.get('alpha')

lasso = Lasso(alpha=alpha)
lasso.fit(X_train, y_train)
```

```
Lasso(alpha=0.0002)
```

```
[827]
#predict y values using lasso model
y_pred_train = lasso.predict(X_train)
y_pred_test = lasso.predict(X_test)
```

```
lasso_metric_scores = metric_scores(y_train, y_pred_train,y_test, y_pred_test)
```

```
r2_train score:  0.8989059797862033
r2_test score:  0.8292532511338078
rss_train score:  1.3404523799744041
rss_test score:  0.7461215056254811
mse_train score:  0.0013154586653330758
mse_test score:  0.0017073718664198652
rmse_train score:  0.036269252340420194
rmse_test score:  0.04132035656210949
```

This implies that model performed better with optimal alpha value.

The important predictor variables before and after doubling the alpha values are same as change in r2 was very less.

GrLivArea
RoofMatl
OverallQual
Neighborhood
GarageCars
2ndFlrSF
BsmtExposure
OverallCond
FullBath
MasVnrArea
BsmtQual
LotShape
KitchenQual
MSSubClass

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:

➔ Optimal value of alpha for ridge regression: 8
Optimal value of alpha for lasso regression: 0.0001

Using these optimal values, when models were respectively fit and as captured in the above screenshots from Ridge and Lasso model scores, Lasso has slightly higher R2 score and almost same RMSE score. Moreover, Lasso model shrinks the size of the predictor variables keeping only the important predictors giving the edge of feature selection.

Choice of regression model: LASSO

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:

➔ 5 most important predictor variables:

GrLivArea
RoofMatl
OverallQual
Neighborhood
GarageCars

5 most important predictor variables after excluding above variables and rebuilding lasso model:

1stFlrSF
2ndFlrSF
MasVnrArea
GarageArea
BsmtExposure

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:

➔ When a model performs well on the data that is used to train it, but does not perform well with unseen data, we know we have a problem: overfitting.

Such a model will perform very well with training data and, hence, will have very low bias; but since it does not perform well with unseen data, it will show high variance. Hence, these overfitted models are less robust and since they perform well only on train dataset, they are not generalized models.

➔ More extreme the magnitude of the coefficients and/or higher the number of coefficients may be the **reasons for a model overfitting**.

➔ **So, to avoid overfitting and thus make a model robust and generalizable, we need to use Regularization.**

When we use regularization, we add a penalty term to the model's cost function.

Here, the cost function would be Cost = RSS + Penalty.

1. Ridge Regression:

   Performs L2 regularization, i.e., adds penalty equivalent to square of the magnitude of coefficients
   Minimization objective = RSS + α * (sum of square of coefficients)

2. Lasso Regression:

   Performs L1 regularization, i.e., adds penalty equivalent to absolute value of the magnitude of coefficients
   Minimization objective = RSS + α * (sum of absolute value of coefficients)

Adding this penalty term in the cost function helps suppress or shrink the extreme magnitude of the model coefficients towards 0. Thereby discouraging the creation of a more complex model and preventing the risk of overfitting.

➔ **Implications of Regularization on Accuracy of the model:**

When we add this penalty and try to get the model parameters that optimize this updated cost function (RSS + Penalty), the coefficients that we get given the training data may not be the best (**maybe more biased**).

So, on adding a penalty term to the cost function, we **compromise a bit on the bias to get a significant reduction in the variance.**

→ we use regularization because we want our models to work well with unseen data, without missing out on identifying underlying patterns in the data. (Nothing but bias-variance trade-off)