Assignment Part -II - Subjective Questions and Answers

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?

According to the problem statement and based on all the analyses and calculations that we have performed, following are the optimal alpha values are:

Ridge → 10, Lasso → 0.001

This model also has the following values as obtained from evaluation in our notebook:

The optimal value of LAMBDA we got in case of Ridge and Lasso is :

- Ridge - **10.0**
- Lasso - **0.001**

The r2 value we got in case of Ridge and Lasso is:

- Ridge - Train = **0.94** , Test = **0.91** (Delta of 0.03 i.e 3%)
- Lasso - Train = **0.92** , Test = **0.90** (Delta of 0.02 i.e 2%)

The Mean Squared error on Test data in case of Ridge and Lasso is:

- Ridge - **0.0061**
- Lasso - **0.0068**

Doubling the values would imply that, the new values for alpha would be following:

Ridge → 20, Lasso → 0.002

We could observe the following changes in the parameters after doubling.

For Ridge, we can see the following metrics and significant features:

```
The output when alpha is 20:
The mean squared error value is  0.006161580736990166
The r2 value of train data is  0.9347436540641076
The r2 value of test data is  0.9100417062054396
Top correlated features when alpha is 20 are :

                     Coefficient
GrLivArea               1.111111
TotalBsmtSF             1.082719
1stFlrSF                1.078620
OverallQual_8           1.062785
OverallQual_9           1.059446
Neighborhood_Crawfor    1.059086
2ndFlrSF                1.058998
LotArea                 1.058272
GarageArea              1.055761
GarageCars              1.046347
```

For Lasso, we can see the following metrics and significant features:

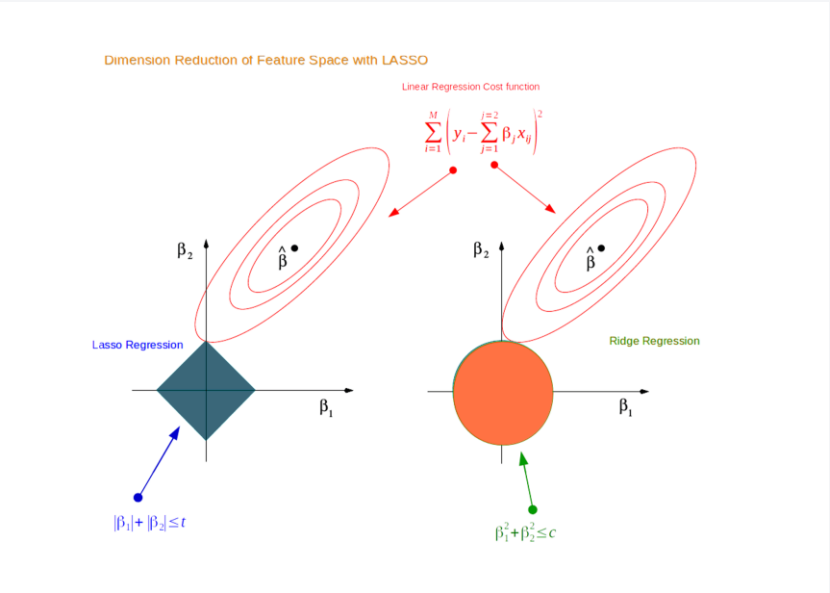
```
The output when alpha is 0.002:
The mean squared error value is  0.008006879793044326
The r2 value of train data is  0.9347436540641076
The r2 value of test data is  0.9100417062054396
Top correlated features when alpha is 0.002 are:

                     Coefficient
GrLivArea               1.344891
TotalBsmtSF             1.143791
YearRemodAdd            1.070822
GarageArea              1.069845
LotArea                 1.065220
OverallQual_8           1.060908
Neighborhood_Crawfor    1.056657
GarageCars              1.053171
BsmtFinSF1              1.049954
OverallQual_9           1.049831
```

Thus, we can see, that by doubling the parameters the model accuracy parameters and its significant features change accordingly. For example, for both Ridge and Lasso there is a very slightly marginal increase in the MSE values on test data.

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?

The choice between Ridge regression and Lasso regression depends on the specific characteristics of the data and the problem at hand. Since we have already determined the optimal value of lambda for both Ridge (Lambda ➔10) and Lasso (Lambda ➔0.001) regression during the assignment, we can simply choose the method that gave us the better performance in terms of prediction accuracy or interpretability, depending on the specific goals of our analysis which is to choose only those features that are significant in terms of prediction.



Ridge regression is a good choice when we have a large number of predictors that are potentially correlated with each other, and we want to shrink the coefficients towards zero without completely eliminating them. Ridge regression is also useful when the coefficients are expected to be of similar magnitude, as it applies a penalty to the sum of squared coefficients.

Lasso regression, on the other hand, is a good choice when we have a large number of predictors but only a few of them are expected to be important. Lasso regression tends to set some of the coefficients to exactly zero, effectively performing variable selection and reducing the number of predictors in the model.

Therefore, for this specific problem statement we can choose Lasso regression. Also, the difference between R2 for Training and Test Set, are slightly lower for Lasso (0.92 – 0.9 = 0.02) than those of Ridge (0.94 – 0.91 = 0.03). Both have similar MSE of 0.01 on the Test set.

In general, if we are unsure about the importance of different predictors, we may want to try both ridge and lasso regression and compare their performance in terms of prediction accuracy and interpretability. If we have prior knowledge or strong reasons to believe that only a few predictors are important, we may choose lasso regression. If we have many predictors and want to avoid overfitting, we may choose 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?

From the Lasso Regression performed in the notebook, the five most important predictor variables were as follows:

1. **GrLivArea** - With an increase of 1 square foot of house area above ground, the price will increase by 1.34 times
2. **TotalBsmtSF** - With an increase of 1 square feet of basement area, the price will increase by 1.14 times
3. **OverallQual_9** - If the overall material and finish of the house is Excellent, the price will increase by 1.09 times.
4. **Neighborhood_Crawfor** - if Crawford is a nearby location, then the price of house will increase by 1.08 times
5. **OverallQual_8** - If the overall material and finish of the house is Very Good, the price will increase by 1.08 times.

Now, if we have to build a new model after eliminating these 5 features it would look somewhat like this:

In [78]:
```
#As per the question, dropping the top 5 most important predictor variables in the lasso model
drop_cols = ['GrLivArea','OverallQual_8','OverallQual_9','TotalBsmtSF','Neighborhood_Crawfor']
X_train.drop(labels = drop_cols, axis = 1, inplace=True)
X_test.drop(labels = drop_cols, axis = 1, inplace=True)

#Let's create a lasso model with alpha  = 0.001
lasso2 = Lasso(alpha=0.001)
lasso2.fit(X_train,y_train)

lasso_coef_df = pd.DataFrame(np.exp(lasso2.coef_) , columns = ['Coefficient'], index =  X_train.columns)
print("Top 5 correlated features when alpha is 0.001 are:\n")
print(lasso_coef_df.sort_values(by = 'Coefficient', ascending = False).head(5))
```

```
Top 5 correlated features when alpha is 0.001 are:

                     Coefficient
1stFlrSF                1.310560
2ndFlrSF                1.191727
BsmtFinSF1              1.081075
LotArea                 1.079323
Neighborhood_Somerst    1.073410
```

The model would have evaluation params and metrics such as the one below:

```
In [79]: ## Making predictions
         y_train_pred = lasso2.predict(X_train)
         y_pred = lasso2.predict(X_test)

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

```
R-Squared (Train) = 0.911180651541823
R-Squared (Test) = 0.8916728937668655
RSS (Train) = 185540.990347002
RSS (Test) = 10465.65959099323
MSE (Train) = 0.006524393833594493
MSE (Test) = 0.007419729553611535
RMSE (Train) = 0.08077371994401703
RMSE (Test) = 0.08613785203736819
```

Therefore, we can conclude from above that the top 5 features after eliminating the previous ones would be: ***1stFlrSF, 2ndFlrSF, BsmtFinSF1, LotArea*** and ***Neighborhood_Somrst***

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?

Regularization techniques like Ridge(L2) and Lasso(L1) regression are often used to improve the generalization performance of machine learning models, i.e., their ability to perform well on new, unseen data. Regularization achieves this by reducing overfitting, which is the tendency of a model to fit the noise in the training data rather than the underlying patterns.

To make sure that a regularized model is robust and generalisable, we can use various techniques such as cross-validation, hold-out validation, and bootstrapping.

Cross-validation involves dividing the data into multiple subsets, using each subset as a validation set and the rest as a training set. This process is repeated multiple times, and the results are averaged to obtain a more reliable estimate of the model's performance. We have used cross validation in our assignment for the same, in order to determine the best value of alpha/lambda i.e., hyperparameter tuning

9.1 Ridge - L2 Regularization

In [55]:
```
ridge = Ridge()

# cross validation
ridgeCV = GridSearchCV(estimator = ridge,
                       param_grid = params,
                       scoring= 'r2',
                       cv = 5,
                       return_train_score=True,
                       verbose = 1)
ridgeCV.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits
```

Out[55]:

GridSearchCV

```
GridSearchCV(cv=5, estimator=Ridge(),
             param_grid={'alpha': [1e-08, 1e-07, 1e-06, 1e-05, 0.0001, 0.001,
                                   0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
                                   0.7, 0.8, 0.9, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0,
                                   7.0, 8.0, 9.0, 10.0, 20, 50, 100, ...]},
             return_train_score=True, scoring='r2', verbose=1)
```

estimator: Ridge

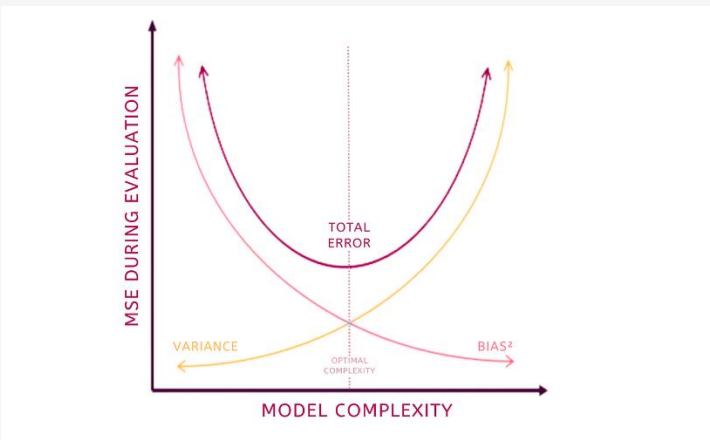
```
Ridge()
```

▸ Ridge

In [56]:
```
## View the optimal value of alpha
ridgeCV.best_params_
```

Out[56]:
```
{'alpha': 10.0}
```

The implications of regularization for the accuracy of the model depend on the specific regularization technique and the data at hand. In general, regularization tends to improve the generalization performance of the model by reducing overfitting, but this may come at the cost of reduced accuracy on the training data. Regularization achieves a balance between bias and variance by reducing the variance of the model at the cost of slightly increasing its bias. This bias-variance trade-off may be more or less pronounced depending on the amount of regularization, the complexity of the model, and the amount and quality of the data.



In summary, to ensure that a regularized model is robust and generalizable, we can use various techniques such as cross-validation, hold-out validation, and bootstrapping. Regularization tends to improve the generalization performance of the model by reducing overfitting, but this may come at the cost of reduced accuracy on the training data due to the bias-variance trade-off.