## 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:

(i) Optimal value of Alpha:

- The most optimal value of alpha for ridge is 0.9
- The most optimal value of alpha for lasso is 0.001

(ii) If we double the value of alpha, then the R2 score for train & test would be:

- Ridge:

```
In [60]: ridge = Ridge(alpha = 1.8)
         ridge.fit(X_train,y_train)

         y_pred_train = ridge.predict(X_train)
         print(r2_score(y_train,y_pred_train))

         y_pred_test = ridge.predict(X_test)
         print(r2_score(y_test,y_pred_test))

         0.8918242986811323
         0.8567137136837922
```

- Lasso:

```
In [65]: lm = Lasso(alpha=0.002)
         lm.fit(X_train,y_train)

         y_train_pred = lm.predict(X_train)
         print(r2_score(y_true=y_train,y_pred=y_train_pred))

         y_test_pred  = lm.predict(X_test)
         print(r2_score(y_true=y_test,y_pred=y_test_pred))

         0.8809959852234268
         0.8527005626784095
```

(iii) The most important predictor variables after the change is implemented:

- Ridge

```
In [64]: ridge_coef.sort_values(by='Coef',ascending=False).head(10)
```

Out[64]:

| | Feaure | Coef |
|---|---|---|
| 48 | YrSold_Old | 1.200680 |
| 67 | Neighborhood_Gilbert | 0.404231 |
| 28 | BedroomAbvGr | 0.377547 |
| 39 | OpenPorchSF | 0.344513 |
| 42 | ScreenPorch | 0.234592 |
| 11 | BsmtExposure | 0.230828 |
| 43 | PoolArea | 0.224708 |
| 2 | LotShape | 0.211108 |
| 58 | LotConfig_FR3 | 0.203819 |
| 45 | YearBuilt_Old | 0.198965 |

- Lasso

```
In [68]: lasso_coef.sort_values(by='Coef',ascending=False).head(10)
```

Out[68]:

| | Feaure | Coef |
|---|---|---|
| 48 | YrSold_Old | 1.217012 |
| 28 | BedroomAbvGr | 0.414466 |
| 11 | BsmtExposure | 0.393817 |
| 67 | Neighborhood_Gilbert | 0.269223 |
| 2 | LotShape | 0.235403 |
| 3 | LandSlope | 0.194057 |
| 18 | HeatingQC | 0.181639 |
| 4 | OverallQual | 0.175523 |
| 62 | Neighborhood_BrkSide | 0.173649 |
| 12 | BsmtFinType1 | 0.149158 |

## 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:

(i) The R2 test score on the Lasso Regression Model is slightly better than that of Ridge Regression Model. Moreover, the training accuracy is slightly reduced; hence, making the model an optimal choice as it seems to perform better on the unseen data.

(ii) The MSE for Test set (Lasso Regression) is slightly lower than that of the Ridge Regression Model; implies Lasso Regression performs better on the unseen test data. Also, since Lasso helps in feature selection (the coefficient values of some of the insignificant predictor variables became 0), implies Lasso Regression has a better edge over Ridge Regression. Therefore, the variables predicted by Lasso

can be applied in order to choose significant variables for predicting the price of a house in this analysis.

Moreover, while choosing a type of regression in the real world, an analyst has to deal with the lurking and confounding dangers of outliers, non-normality of errors and overfitting especially in sparse datasets among others. Using L2 norm (Ridge) results in exposing the analyst to such risks. Hence, use of L1 norm (Lasso) could be quite beneficial as it is quite robust to fend off such risks to a large extent, thereby resulting in better and robust regression models.

## 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:

Those 5 most important predictor variables that will be excluded are :-

1. GrLivArea
2. OverallQual
3. OverallCond
4. TotalBsmtSF
5. GarageArea

## 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?**

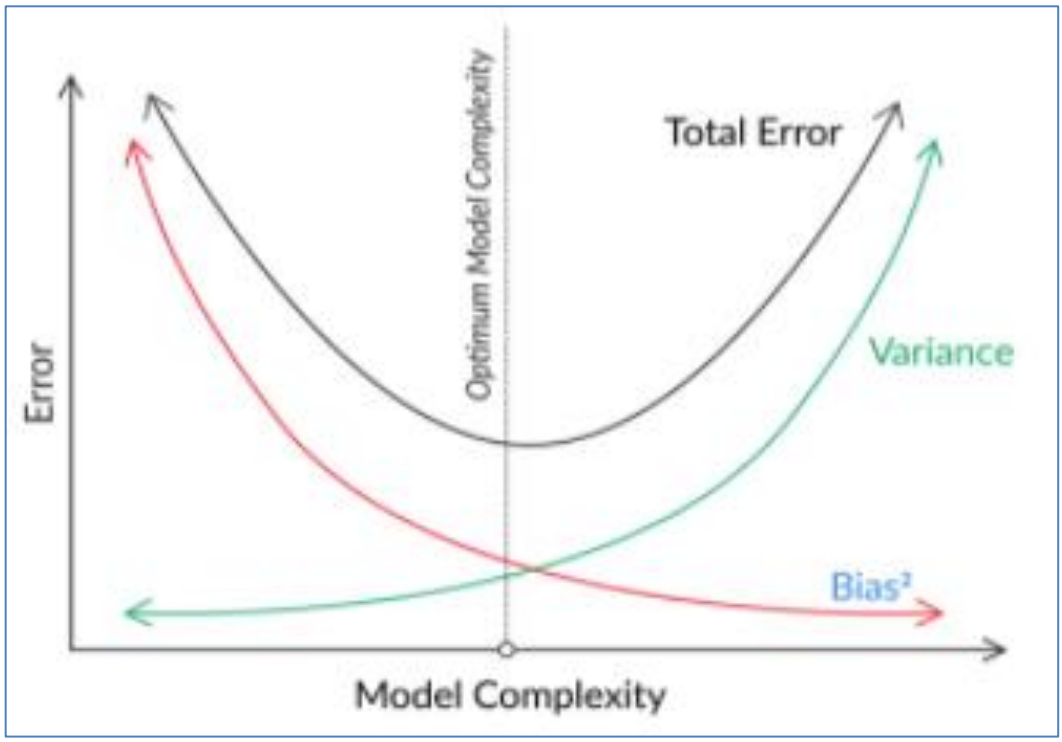
## Answer:

Robustness of a model implies, either the testing error of the model is consistent with the training error, the model performs well with enough stability even after adding some noise to the dataset. Thus, the robustness (or generalizability) of a model is a measure of its successful application to data sets other than the one used for training and testing.

By the implementing regularization techniques, we can control the trade-off between model complexity and bias which is directly connected the robustness of the model. Regularization, helps in penalizing the coefficients for making the model too complex; thereby allowing only the optimal amount of complexity to the model. It helps in controlling the robustness of the model by making the model optimal simpler. Therefore, in order to make the model more robust and generalizable, one need to make sure that there is a delicate balance between keeping the model simple and not making it too naive to be of any use. Also, making a model simple leads to Bias-Variance Trade-off:

- A complex model will need to change for every little change in the dataset and hence is very unstable and extremely sensitive to any changes in the training data.

- A simpler model that abstracts out some pattern followed by the data points given is unlikely to change wildly even if more points are added or removed.




Thus, accuracy of the model can be maintained by keeping the balance between Bias and Variance as it minimizes the total error as shown in the below graph. Thus, accuracy and robustness may be at the odds to each other as too much accurate model can be prey to over fitting hence it can be too much accurate on train data but fails when it faces the actual data or vice versa.