: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

The optimum value of alphs for ridge is 2.0 and lasso is 50.
(Note: Implementation is done HousePricePrediction python notebook)

Doubling the value of alpha

```
]: #1. Optimum value alpha for ridge is 2.0 and lasso is 50 .Doubling this value
   # Ridge
   alpha = 4
   ridge2 = Ridge(alpha = alpha)
   ridge2.fit(X_train_rfe,y_train)
   y_pred_train = ridge2.predict(X_train_rfe)
   y_pred_test = ridge2.predict(X_test_rfe)
   r2_train = r2_score(y_train,y_pred_train)
   r2_test = r2_score(y_test,y_pred_test)
   print("r2_train:",r2_train ,"r2_test:",r2_test)
   # For Alpha = 2
   # r2_train: 0.8763726463385658 r2_test: 0.8642383653545764
```

R2 score for testing and training data has decreased

```
: #Lasso
  aplha =100
  lasso2 = Lasso(alpha=alpha)
  lasso2.fit(X_train_rfe,y_train)
  y_pred_train = lasso2.predict(X_train_rfe)
  y_pred_test = lasso2.predict(X_test_rfe)
  r2_train = r2_score(y_train,y_pred_train)
  r2_test = r2_score(y_test,y_pred_test)
  print("r2_train:",r2_train ,"r2_test:",r2_test)
  # For Alpha = 50
  # r2_train: 0.8772967512726669 r2_test: 0.8652796218220864

  r2_train: 0.8797600618352953 r2_test: 0.8592549924821122
```

R2 score for training data has increased slightly while test data has decreased

Important Predictor variables:

[97]:
```
betas = pd.DataFrame(index=X_train_rfe.columns)
betas.rows = X_train_rfe.columns
betas['Ridge'] = ridge.coef_
betas['Ridge2'] = ridge2.coef_
betas['Lasso'] = lasso.coef_
betas['Lasso2'] = lasso2.coef_
betas.sort_values(by = 'Lasso2' , ascending =False)
```

[97]:

	Ridge	Ridge2	Lasso	Lasso2
OverallQual	116725.809938	106730.618178	133814.008365	127823.591344
GrLivArea	84416.192776	80810.190439	156045.785707	86330.208347
1stFlrSF	70154.505214	67546.215064	10740.639065	81587.431675
TotalBsmtSF	58290.861498	58140.335360	73282.332386	71933.071046
Street_Pave	44293.248231	32019.189373	54717.397312	70743.980435
LotArea	53429.428819	48159.785087	58545.631363	61240.112023
TotRmsAbvGrd	54422.168936	54801.898254	50009.635860	53679.085388
BsmtFinSF1	66983.519630	66691.446461	53503.571842	53559.183703
YearBuilt	47027.620734	47704.968388	44707.317811	44712.735021

Predictors are Same but the coefficients of predictor has changed

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:

We will choose Lasso regression as R2 score for Lasso is slightly higher than Ridge on test dataset.Also RMSE value for Lasso is slightly lower than ridge.

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:

(Note: Implementation is done HousePricePrediction python notebook)

Removing top 5 variables OverallQual,GrLivArea,TotalBsmtSF,LotArea,BsmtFinSF1 and performing Lasso Regression

```
#2.Removing top 5 variables and performing Lasso regression OverallQual,GrLivArea,TotalBsmtSF,LotArea,BsmtFinSF1
X_train2 = X_train_rfe.drop(['OverallQual','GrLivArea','TotalBsmtSF','LotArea','BsmtFinSF1'],axis=1)
X_test2 = X_test_rfe.drop(['OverallQual','GrLivArea','TotalBsmtSF','LotArea','BsmtFinSF1'],axis=1)
```

```
aplha = 50
lasso3 = Lasso(alpha=alpha)
lasso3.fit(X_train2,y_train)
y_pred_train = lasso3.predict(X_train2)
y_pred_test = lasso3.predict(X_test2)
r2_train = r2_score(y_train,y_pred_train)
r2_test = r2_score(y_test,y_pred_test)
print("r2_train:",r2_train ,"r2_test:",r2_test)
# For Alpha = 50
# r2_train: 0.8772967512726669 r2_test: 0.8652796218220864
```

```
r2_train: 0.8355266040212833 r2_test: 0.7873992905646137
```

R2 score of Training and testing data is decreased

R2 score of training and testing data is decreased.

Important predictor variables

```
[107]: #Important preedictor variable
betas = pd.DataFrame(index = X_train2.columns)
betas.rows = X_train2.columns
betas['Lasso3'] = lasso3.coef_
betas.sort_values(by = 'Lasso3' , ascending = False )
```

[107]:

	Lasso3
1stFlrSF	339190.321066
2ndFlrSF	111179.485427
RoofMatl_Metal	91045.205672
Street_Pave	84072.627847
YearBuilt	83703.569902
RoofStyle_Shed	79342.872806

1. **1stFlrSF**
2. **2ndFlrSF**
3. **RoofMatl_Metal**
4. **Street_Pave**
5. **YearBuilt**

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:

Model Should be generalisable so that testing accuracy is not less that training accuracy.
This can be done by adding more data,treating missing and outlier values.Only those outliers which makes more sense has to be retained else can be removed.Feature selection also plays important role.All these factors helps improving the model accurracy which inturn performs well on unseen data.Model has to be robust so that we can trust the model for its predictive analysis.