Coding for Question 1

jupyter HousingAssignment Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [90]:
```
#Coding for 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?
#best value for ridge regression is 10
#best value for lasso regression is .001

#If we want to double it let us actually do it on ridge
ridge = Ridge(alpha=20)

# Fit the model on training data
ridge.fit(X_train, y_train)

y_train_pred = ridge.predict(X_train)
y_pred = ridge.predict(X_test)
ridge_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R^2 (Train) = 0.92
R^2 (Test) = 0.89
RSS (Train) = 15.04
RSS (Test) = 6.05
MSE (Train) = 0.01
MSE (Test) = 0.02
RMSE (Train) = 0.11
RMSE (Test) = 0.14
```

In [91]:
```
#Let us do it on lasso now
lasso = Lasso(alpha=0.002)

# Fit the model on training data
lasso.fit(X_train, y_train)
y_train_pred = lasso.predict(X_train)
y_pred = lasso.predict(X_test)
lasso_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R^2 (Train) = 0.88
R^2 (Test) = 0.87
RSS (Train) = 20.85
RSS (Test) = 7.09
MSE (Train) = 0.02
MSE (Test) = 0.02
RMSE (Train) = 0.13
```

jupyter HousingAssignment Last Checkpoint: a minute ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [91]:
```
#Let us do it on Lasso now
lasso = Lasso(alpha=0.002)

# Fit the model on training data
lasso.fit(X_train, y_train)
y_train_pred = lasso.predict(X_train)
y_pred = lasso.predict(X_test)
lasso_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R^2 (Train) = 0.88
R^2 (Test) = 0.87
RSS (Train) = 20.85
RSS (Test) = 7.09
MSE (Train) = 0.02
MSE (Test) = 0.02
RMSE (Train) = 0.13
RMSE (Test) = 0.16
```

In [93]:
```
lr_table = {'Metric': ['R^2 Score (Train)','R^2 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')
```

Out[93]:

Metric	Ridge Regression	Lasso Regression
R^2 Score (Train)	0.915551	0.882902
R^2 Score (Test)	0.888895	0.869974
RSS (Train)	15.036588	20.849919
RSS (Test)	6.054215	7.085255
MSE (Train)	0.012874	0.017851
MSE (Test)	0.020734	0.024265
RMSE (Train)	0.113463	0.133607
RMSE (Test)	0.143992	0.155771

jupyter HousingAssignment Last Checkpoint: 2 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [94]:
```
#so the R2 score did not change as it went from .92 to .92 with doubling the lambda value for train ridge regression
#the R2 score did not really change for the test ridge regression as it went from .89 to .88
#the R2 score changed from .90 to .88  with doubling the lambda value for train lasso regression
#the R2 score changed from 0.88 to .86 an for the test lasso regression
```

In [96]:
```
Bcoeff = pd.DataFrame(index=X.columns)
Bcoeff.rows = X.columns
Bcoeff['Ridge'] = ridge.coef_
Bcoeff['Lasso'] = lasso.coef_
Bcoeff
```

Out[96]:

	Ridge	Lasso
LotFrontage	-1.046520e-02	-0.004135
LotArea	1.617495e-02	0.012602
YearRemodAdd	3.007310e-02	0.037721
MasVnrArea	3.371510e-03	0.000000
BsmtFinSF1	-9.020999e-04	0.000000
BsmtFinSF2	7.244376e-03	0.000000
BsmtUnfSF	2.880308e-03	0.000000
TotalBsmtSF	4.578704e-03	0.025243
1stFlrSF	3.431187e-02	0.005681
2ndFlrSF	4.250041e-02	0.000000
LowQualFinSF	4.341575e-03	-0.000000

In [97]:
```
Bcoeff['Ridge'].sort_values(ascending=False)[:10]
```

Out[97]:
```
OverallQual_9           0.100129
OverallQual_8           0.071467
Neighborhood_Crawfor    0.070411
Neighborhood_NridgHt    0.067788
CentralAir_Y            0.062891
Neighborhood_StoneBr    0.062816
Functional_Typ          0.061949
GrLivArea               0.061318
Exterior1st_BrkFace     0.060103
```

jupyter HousingAssignment Last Checkpoint: 2 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [97]:
```
Bcoeff['Ridge'].sort_values(ascending=False)[:10]
```

Out[97]:
```
OverallQual_9           0.100129
OverallQual_8           0.071467
Neighborhood_Crawfor    0.070411
Neighborhood_NridgHt    0.067788
CentralAir_Y            0.062891
Neighborhood_StoneBr    0.062816
Functional_Typ          0.061949
GrLivArea               0.061318
Exterior1st_BrkFace     0.060103
BsmtCond_TA             0.056340
Name: Ridge, dtype: float64
```

In [98]:
```
ridge_coeffs = np.exp(Bcoeff['Ridge'])
ridge_coeffs.sort_values(ascending=False)[:10]
Bcoeff['Lasso'].sort_values(ascending=False)[:10]
```

Out[98]:
```
OverallQual_9           0.159031
GrLivArea               0.107007
OverallQual_8           0.098249
Neighborhood_Crawfor    0.074540
Functional_Typ          0.072462
CentralAir_Y            0.070772
GarageCars              0.057090
Exterior1st_BrkFace     0.048741
Neighborhood_NridgHt    0.044696
Condition1_Norm         0.041073
Name: Lasso, dtype: float64
```

In [99]:
```
lasso_coeffs = np.exp(Bcoeff['Lasso'])
lasso_coeffs.sort_values(ascending=False)[:10]
```

Out[99]:
```
OverallQual_9           1.172375
GrLivArea               1.112942
OverallQual_8           1.103237
Neighborhood_Crawfor    1.077388
Functional_Typ          1.075152
CentralAir_Y            1.073336
GarageCars              1.058751
Exterior1st_BrkFace     1.049948
Neighborhood_NridgHt    1.045710
```


```
GrLivArea               0.107007
OverallQual_8           0.098249
Neighborhood_Crawfor    0.074540
Functional_Typ          0.072462
CentralAir_Y            0.070772
GarageCars              0.057090
Exterior1st_BrkFace     0.048741
Neighborhood_NridgHt    0.044696
Condition1_Norm         0.041073
Name: Lasso, dtype: float64
```

In [99]:
```
lasso_coeffs = np.exp(Bcoeff['Lasso'])
lasso_coeffs.sort_values(ascending=False)[:10]
```

Out[99]:
```
OverallQual_9           1.172375
GrLivArea               1.112942
OverallQual_8           1.103237
Neighborhood_Crawfor    1.077388
Functional_Typ          1.075152
CentralAir_Y            1.073336
GarageCars              1.058751
Exterior1st_BrkFace     1.049948
Neighborhood_NridgHt    1.045710
```


```
Condition1_Norm         1.041928
Name: Lasso, dtype: float64
```

In [ ]:
```
#so the most important predictor values after you double the value is
#OverallQual_9           1.172375
#GrLivArea               1.112942
#OverallQual_8           1.103237
#Neighborhood_Crawfor    1.077388
#Functional_Typ          1.075152
#CentralAir_Y            1.073336
#GarageCars              1.058751
#Exterior1st_BrkFace     1.049948
#Neighborhood_NridgHt    1.045710
#Condition1_Norm         1.041928
```

Q1 Answer: So the R2 score did not change as it went from 0.92 to 0.92 with doubling the lambda value for train ridge regression. The R2 score did not really change for the test ridge regression as it went from 0.89 to 0.88. The R2 score changed from 0.90 to 0.88 with doubling the lambda value for train lasso regression. The R2 score changed from 0.88 to 0.86 an for the test lasso regression. The most important predictor variables now are

OverallQual_9	1.172375
GrLivArea	1.112942
OverallQual_8	1.103237
Neighborhood_Crawfor	1.077388
Functional_Typ	1.075152
CentralAir_Y	1.073336
GarageCars	1.058751
Exterior1st_BrkFace	1.049948
Neighborhood_NridgHt	1.045710
Condition1_Norm	1.041928

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?
I would choose to apply lasso regression as it finds the important predictor variables which is what this case was asking. Additionally, it reduces the amount of important predictor variables by taking out the non important ones.

Question 3

jupyter HousingAssignment Last Checkpoint: 12 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [110]:
```
#Coding for question 3
bad5 = ['OverallQual_9', 'GrLivArea', 'OverallQual_8', 'Neighborhood_Crawfor', 'Neighborhood_NridgHt']
X_train_dropped = X_train.drop(bad5, axis=1)
X_test_dropped = X_test.drop(bad5, axis=1)
params = {'alpha': [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, 500, 1000]}

lasso = Lasso()

# cross validation

lassoCross = GridSearchCV(estimator = lasso,
                          param_grid = params,
                          scoring= 'neg_mean_absolute_error',
                          cv = 5,
                          return_train_score=True,
                          verbose = 1, n_jobs=-1)
lassoCross.fit(X_train_dropped, y_train)
lassoCross.best_params_
```

```
Fitting 5 folds for each of 28 candidates, totalling 140 fits
```

Out[110]:
```
{'alpha': 0.0001}
```

In [ ]:

In [114]:
```
lasso = Lasso(alpha=0.001)
lasso.fit(X_train_dropped, y_train)
y_train_pred = lasso.predict(X_train_dropped)
y_pred = lasso.predict(X_test_dropped)
lasso_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R^2 (Train) = 0.90
R^2 (Test) = 0.87
RSS (Train) = 18.49
RSS (Test) = 6.95
MSE (Train) = 0.02
MSE (Test) = 0.02
RMSE (Train) = 0.13
RMSE (Test) = 0.15
```

In [115]:
```
lr_table = {'Metric': ['R2 Score (Train)','R2 Score (Test)','RSS (Train)','RSS (Test)',
```

jupyter HousingAssignment Last Checkpoint: 13 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [115]:
```
lr_table = {'Metric': ['R2 Score (Train)','R2 Score (Test)','RSS (Train)','RSS (Test)',
                       'MSE (Train)','MSE (Test)', 'RMSE (Train)', 'RMSE (Test)'],
            'Lasso Regression' : lasso_metrics
        }

final_metric = pd.DataFrame(lr_table, columns = ['Metric', 'Lasso Regression'] )
final_metric.set_index('Metric')
```

Out[115]:

Metric	Lasso Regression
R2 Score (Train)	0.896183
R2 Score (Test)	0.872421
RSS (Train)	18.485186
RSS (Test)	6.951929
MSE (Train)	0.015826
MSE (Test)	0.023808
RMSE (Train)	0.125803
RMSE (Test)	0.154298

In [116]:
```
BCoeff = pd.DataFrame(index=X_train_dropped.columns)
BCoeff.rows = X_train_dropped.columns
BCoeff['Lasso'] = lasso.coef_
BCoeff['Lasso'].sort_values(ascending=False)[:5]
```

Out[116]:
```
Exterior1st_BrkFace    0.091095
2ndFlrSF               0.090993
1stFlrSF               0.081430
Functional_Typ         0.080675
BsmtCond_TA            0.075369
Name: Lasso, dtype: float64
```

So the most important predictor variables after taking out the top 5 are:
Exterior1st_BrkFace 0.091095
2ndFlrSF 0.090993
1stFlrSF 0.081430
Functional_Typ 0.080675
BsmtCond_TA 0.075369

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?

A model can be ensured that it is robust and generalisable by making sure there is thorough data preparation. Making sure there are no NA values or missing values that could affect the analysis is very important. Additionally, finding the right lambda value for lasso and ridge regression is important to control regularization and to stop overfitting. Finally, cross validation is important to make sure that the training data split into folds and evaluated to prevent overfitting.

To ensure accuracy, make sure the model is not overfit as it will not work on new test data. Additionally, not making the model too complex is helpful.