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 Optimal alpha value for ridge and lasso regression:-

1.) Ridge Alpha 2
2.) lasso Alpha 50

The changes of model with the double the value of alpha for both ridge and lasso:-

Ridge Regression

In [100]:
```
#Change the alpha value from 2 to 4
alpha = 4
ridge2_house = Ridge(alpha=alpha)
ridge2_house.fit(X_house_price_prediction_train1, y_house_price_prediction_train)
```

Out[100]:
```
Ridge(alpha=4)
```

In [101]:
```
# Lets calculate some metrics such as R2 score, RSS and RMSE
y_house_price_prediction_pred_train = ridge2_house.predict(X_house_price_prediction_train1)
y_house_price_prediction_pred_test = ridge2_house.predict(X_house_price_prediction_test1)

metric2_house = []
r2_house_train_lr = r2_score(y_house_price_prediction_train, y_house_price_prediction_pred_train)
print("R2score of train data: ",r2_house_train_lr)
metric2_house.append(r2_house_train_lr)

r2_house_test_lr = r2_score(y_house_price_prediction_test, y_house_price_prediction_pred_test)
print("R2score of test data: ",r2_house_test_lr)
metric2_house.append(r2_house_test_lr)

rss1_house_lr = np.sum(np.square(y_house_price_prediction_train - y_house_price_prediction_pred_train))
print("RSS of train data: ",rss1_house_lr)
metric2_house.append(rss1_house_lr)

rss2_house_lr = np.sum(np.square(y_house_price_prediction_test - y_house_price_prediction_pred_test))
print("RSS of test data: ",rss2_house_lr)
metric2_house.append(rss2_house_lr)

mse_house_train_lr = mean_squared_error(y_house_price_prediction_train, y_house_price_prediction_pred_train)
print("RMSE of train data: ",mse_house_train_lr)
metric2_house.append(mse_house_train_lr**0.5)

mse_house_test_lr = mean_squared_error(y_house_price_prediction_test, y_house_price_prediction_pred_test)
print("RMSE of test data: ",mse_house_test_lr)
metric2_house.append(mse_house_test_lr**0.5)
```

```
R2score of train data:  0.8765838882231998
R2score of test data:  0.8686950819959689
RSS of train data:  649906268457.3325
RSS of test data:  299021843214.3202
RMSE of train data:  696576922.2479448
RMSE of test data:  747554608.0358005
```

The R2 score decreased on the training data but increased on the test data

Lasso Regression

In [102]:
```
#Change the alpha value from 50 to 100
alpha =100

lasso2_house = Lasso(alpha=alpha)

lasso2_house.fit(X_house_price_prediction_train1, y_house_price_prediction_train)
```

Out[102]:
```
Lasso(alpha=100)
```

```
In [103]: # Lets calculate some metrics such as R2 score, RSS and RMSE

y_house_price_prediction_pred_train = lasso2_house.predict(X_house_price_prediction_train1)
y_house_price_prediction_pred_test = lasso2_house.predict(X_house_price_prediction_test1)

metric3_house = []
r2_house_train_lr = r2_score(y_house_price_prediction_train, y_house_price_prediction_pred_train)
print("R2score of train data: ",r2_house_train_lr)
metric3_house.append(r2_house_train_lr)

r2_house_test_lr = r2_score(y_house_price_prediction_test, y_house_price_prediction_pred_test)
print("R2score of test data: ",r2_house_test_lr)
metric3_house.append(r2_house_test_lr)

rss1_house_lr = np.sum(np.square(y_house_price_prediction_train - y_house_price_prediction_pred_train))
print("RSS of train data: ",rss1_house_lr)
metric3_house.append(rss1_house_lr)

rss2_house_lr = np.sum(np.square(y_house_price_prediction_test - y_house_price_prediction_pred_test))
print("RSS of test data: ",rss2_house_lr)
metric3_house.append(rss2_house_lr)

mse_house_train_lr = mean_squared_error(y_house_price_prediction_train, y_house_price_prediction_pred_train)
print("RMSE of train data: ",mse_house_train_lr)
metric3_house.append(mse_house_train_lr**0.5)

mse_house_test_lr = mean_squared_error(y_house_price_prediction_test, y_house_price_prediction_pred_test)
print("RMSE of test data: ",mse_house_test_lr)
metric3_house.append(mse_house_test_lr**0.5)

R2score of train data:  0.8792337708599172
R2score of test data:  0.872220907888763
RSS of train data:  635952050393.8643
RSS of test data:  290992448936.1584
RMSE of train data:  681620632.7908512
```

The R2 score decreased on the training data but increased on the test data

The change implemented:-

```
In [104]: #important predictor variables
betas_house = pd.DataFrame(index=X_house_price_prediction_train1.columns)
betas_house.rows = X_house_price_prediction_train1.columns
betas_house['Ridge2'] = ridge2_house.coef_
betas_house['Ridge'] = ridge_house.coef_
betas_house['Lasso'] = lasso_house.coef_
betas_house['Lasso2'] = lasso2_house.coef_
pd.set_option('display.max_rows', None)
betas_house.head(70)
```

Out[104]:

	Ridge2	Ridge	Lasso	Lasso2
LotArea	48481.175823	54085.526418	59127.110707	53866.642717
OverallQual	104306.752074	112466.016096	125870.184347	129702.973957
OverallCond	32225.792221	36170.845304	38489.048215	35394.015121
YearBuilt	56374.768728	57414.556294	56404.875297	54944.783023
BsmtFinSF1	52600.896041	51502.344099	49800.510434	50352.777698
TotalBsmtSF	70332.440052	74585.173649	77939.130586	76878.128775
1stFlrSF	69831.795159	72524.540318	10484.387768	11645.130416
2ndFlrSF	32592.859719	35064.821005	0.000000	0.000000
GrLivArea	82374.408529	86210.604460	158477.106233	154254.322638
BedroomAbvGr	-34247.713202	-46172.064700	-54311.410282	-43188.949191
TotRmsAbvGrd	54338.462601	53660.357612	49056.082490	45763.357075
Street_Pave	30023.467457	41019.720864	49410.571082	33928.832694
LandSlope_Sev	-14835.847836	-21340.780496	-17590.848111	-6644.343166
RoofStyle_Gable	-3892.800285	-3876.908683	-0.000000	-3.102487
RoofStyle_Hip	-1454.749165	-2955.421214	-0.000000	0.000000
RoofStyle_Shed	3858.188196	6174.287591	0.000000	0.000000
RoofMatl_Metal	5380.747869	9490.536669	0.000000	0.000000
Exterior1st_CBlock	-10740.817847	-16635.884356	-18639.095861	-0.000000
Exterior1st_Stone	-13589.574954	-24084.220103	-28272.316287	-0.000000
Exterior2nd_CBlock	-10740.817847	-16635.884356	-0.000000	-0.000000
ExterQual_Gd	-45132.640889	-50488.575634	-52963.322936	-49293.210333
ExterQual_TA	-60402.417221	-62600.009475	-62174.818369	-59009.028182
Foundation_Wood	-4862.078071	-9457.310779	-0.000000	-0.000000
Heating_OthW	-8351.847428	-14373.635842	-0.000000	-0.000000
Functional_Maj2	-9258.392483	-15799.821816	-2854.327419	-0.000000

The most important predictor variables:-

- LotArea (Lot size in square feet)
- OverallQual (Rates the overall material and finish of the house)
- OverallCond (Rates the overall condition of the house)
- YearBuilt (Original construction date)
- BsmtFinSF1 (Type 1 finished square feet)
- TotalBsmtSF (Total square feet of basement area)
- GrLivArea (Above grade (ground) living area square feet)
- TotRmsAbvGrd (Total rooms above grade (does not include bathrooms))
- Street_Pave (Pave road access to property)

The predictors are the same, but the coefficient of these predictors 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:

The lasso r2_score is slightly higher than the test dataset lasso, so we choose lasso regression to solve this problem.

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:

Another model excluding the five most important predictor variables:-

```
In [105]: X_house_price_prediction_train1
```

417	0.597818	0.555556	0.625	0.320896	0.515539	0.427324	0.323872	0.778399	0.722615	0.666667	0.777778	1.0
323	0.123654	0.222222	0.875	0.597015	0.117002	0.362445	0.248544	0.000000	0.230199	0.500000	0.333333	1.0
1358	0.017881	0.555556	0.500	0.932836	0.172761	0.235808	0.105167	0.499069	0.368386	0.500000	0.222222	1.0
1319	0.249193	0.333333	0.500	0.589552	0.224863	0.269495	0.170306	0.000000	0.157735	0.500000	0.222222	1.0
157	0.300266	0.777778	0.500	1.000000	0.000000	0.241422	0.106987	0.741155	0.501517	0.666667	0.555556	1.0
978	0.227342	0.333333	0.500	0.589552	0.252285	0.278852	0.150655	0.000000	0.139535	0.500000	0.222222	1.0
937	0.233768	0.666667	0.500	0.970149	0.155850	0.347162	0.230349	0.532588	0.502528	0.500000	0.555556	1.0
976	0.125939	0.333333	0.750	0.358209	0.000000	0.137243	0.141557	0.000000	0.131109	0.333333	0.111111	1.0
155	0.231626	0.555556	0.500	0.365672	0.000000	0.178415	0.033479	0.325264	0.207617	0.333333	0.222222	1.0
1278	0.227999	0.777778	0.500	0.947761	0.367459	0.351840	0.235808	0.560521	0.522750	0.500000	0.444444	1.0
1132	0.240195	0.555556	0.375	0.037313	0.000000	0.314410	0.254003	0.640596	0.583081	0.833333	0.555556	1.0
248	0.280671	0.666667	0.500	0.955224	0.000000	0.262009	0.131004	0.513966	0.400404	0.500000	0.555556	1.0
846	0.223543	0.666667	0.500	0.880597	0.234461	0.230817	0.191412	0.477343	0.436468	0.500000	0.444444	1.0

In [106]:
```
y_house_price_prediction_train
```

Out[106]:
```
366     159000
390     119000
154     125000
417     239000
323     126175
1358    177500
1319    111000
157     269500
978     110000
937     253000
976      85500
155      79000
1278    237000
1132    117500
248     180000
846     213000
303     149900
357     134000
474     251000
```

In [107]:
```
X_house_price_prediction_train1.columns
```

Out[107]:
```
Index(['LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'BsmtFinSF1', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea',
'BedroomAbvGr', 'TotRmsAbvGrd', 'Street_Pave', 'LandSlope_Sev', 'RoofStyle_Gable', 'RoofStyle_Hip', 'RoofStyle_Shed', 'RoofMatl
_Metal', 'Exterior1st_CBlock', 'Exterior1st_Stone', 'Exterior2nd_CBlock', 'ExterQual_Gd', 'ExterQual_TA', 'Foundation_Wood', 'H
eating_OthW', 'Functional_Maj2'], dtype='object')
```

In [108]:
```
e_price_prediction_train2 = X_house_price_prediction_train1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF']
e_price_prediction_test2 = X_house_price_prediction_test1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],a
```

In [109]:
```
X_house_price_prediction_train2.head()
```

Out[109]:

	OverallCond	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd	Street_Pave	LandSlope_Sev	RoofStyle_Gable	RoofStyle_Hip	RoofStyle_She
366	0.500	0.332606	0.000000	0.308055	0.500000	0.333333	1.0	0	1	0	
390	0.875	0.175036	0.252017	0.298955	0.666667	0.444444	1.0	0	1	0	
154	0.500	0.262009	0.000000	0.242669	0.666667	0.444444	1.0	0	1	0	
417	0.625	0.323872	0.778399	0.722615	0.666667	0.777778	1.0	0	0	1	
323	0.875	0.248544	0.000000	0.230199	0.500000	0.333333	1.0	0	1	0	

In [110]:
```
X_house_price_prediction_test2.head()
```

Out[110]:

	OverallCond	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd	Street_Pave	LandSlope_Sev	RoofStyle_Gable	RoofStyle_Hip	RoofStyle_Sh
990	0.50	0.337336	0.611421	0.644422	0.5	0.444444	1.0	0	1	0	
1161	0.75	0.422125	0.000000	0.390967	0.5	0.444444	1.0	0	0	1	
1369	0.50	0.432314	0.000000	0.400404	0.5	0.555556	1.0	0	0	1	
329	0.50	0.042213	0.369957	0.239973	0.5	0.333333	1.0	0	1	0	
262	0.75	0.266376	0.000000	0.246714	0.5	0.333333	1.0	0	1	0	

In [111]:
```
alpha =50

lasso21_house = Lasso(alpha=alpha)

lasso21_house.fit(X_house_price_prediction_train2, y_house_price_prediction_train)
lasso21_house.coef_
```

Out[111]:
```
array([ 1.29560726e+04,  1.40078519e+05,  0.00000000e+00,  2.14245892e+05,
       -7.17768784e+04,  4.15422840e+04,  8.99688286e+04, -3.29897519e+04,
        1.00886633e+04,  1.33187216e+04,  1.83167008e+04,  4.78789950e+04,
       -1.02045716e+05, -9.05034005e+04, -9.71066528e+01, -6.62886172e+04,
       -1.06028782e+05, -0.00000000e+00, -3.16435747e+04, -1.43098807e+04])
```

```
In [112]: # Lets calculate some metrics such as R2 score, RSS and RMSE
y_house_price_prediction_pred_train = lasso21_house.predict(X_house_price_prediction_train2)
y_house_price_prediction_pred_test = lasso21_house.predict(X_house_price_prediction_test2)

metric3_house = []
r2_house_train_lr = r2_score(y_house_price_prediction_train, y_house_price_prediction_pred_train)
print("R2score of train data: ",r2_house_train_lr)
metric3_house.append(r2_house_train_lr)

r2_house_test_lr = r2_score(y_house_price_prediction_test, y_house_price_prediction_pred_test)
print("R2score of test data: ",r2_house_test_lr)
metric3_house.append(r2_house_test_lr)

rss1_house_lr = np.sum(np.square(y_house_price_prediction_train - y_house_price_prediction_pred_train))
print("RSS of train data: ",rss1_house_lr)
metric3_house.append(rss1_house_lr)

rss2_house_lr = np.sum(np.square(y_house_price_prediction_test - y_house_price_prediction_pred_test))
print("RSS of test data: ",rss2_house_lr)
metric3_house.append(rss2_house_lr)

mse_house_train_lr = mean_squared_error(y_house_price_prediction_train, y_house_price_prediction_pred_train)
print("RMSE of train data: ",mse_house_train_lr)
metric3_house.append(mse_house_train_lr**0.5)

mse_house_test_lr = mean_squared_error(y_house_price_prediction_test, y_house_price_prediction_pred_test)
print("RMSE of test data: ",mse_house_test_lr)
metric3_house.append(mse_house_test_lr**0.5)

R2score of train data:  0.7908672063150358
R2score of test data:  0.7743103977250472
RSS of train data:  1101288248341.996
RSS of test data:  513964913823.64966
RMSE of train data:  1180373256.5294707
RMSE of test data:  1284912284.5591242
```

R2score of training and testing data decreased

```
In [113]: #important predictor variables
betas_house = pd.DataFrame(index=X_house_price_prediction_train2.columns)
betas_house.rows = X_house_price_prediction_train2.columns
betas_house['Lasso21'] = lasso21_house.coef_
pd.set_option('display.max_rows', None)
betas_house.head(68)
```

Out[113]:

	Lasso21
OverallCond	12956.072572
1stFlrSF	140078.518743
2ndFlrSF	0.000000
GrLivArea	214245.892084
BedroomAbvGr	-71776.878424
TotRmsAbvGrd	41542.283956
Street_Pave	89968.828611
LandSlope_Sev	-32989.751930
RoofStyle_Gable	10088.663267
RoofStyle_Hip	13318.721578
RoofStyle_Shed	18316.700817
RoofMatl_Metal	47878.995039
Exterior1st_CBlock	-102045.716237
Exterior1st_Stone	-90503.400452
Exterior2nd_CBlock	-97.106653
ExterQual_Gd	-66288.617184
ExterQual_TA	-106028.782335
Foundation_Wood	-0.000000
Heating_OthW	-31643.574655
Functional_Maj2	-14309.880682

The 5 most important predictor variables are:-

- 1stFlrSF (First Floor square feet)
- GrLivArea (Above grade (ground) living area square feet)
- Street_Pave (Pave road access to property)
- TotRmsAbvGrd (Total rooms above grade (does not include bathrooms))
- RoofMatl_Metal (Roof material_Metal)

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:

The model should be generalized so that the test accuracy is not less than the training score. The model should be accurate for datasets other than those used during training. Outliers should not be given too much importance so that the accuracy predicted by the model is high. To ensure that this is not the case, outlier analysis should be performed and only those relevant to the data set retained. Those outliers that do not make sense to keep must be removed from the dataset. If the model is not robust, it cannot be trusted for predictive analytics.