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 :

1. The optimal value of alpha for ridge and lasso regression Ridge Alpha 1

lasso Alpha 10

<u>Ridge Regression :</u>
#Change the alpha value from 1 to 2 alpha = 3

ridge2=Ridge(alpha=alpha)

ridge2.fit(X_train1,y_train)

<u>output:</u>
Ridge(alpha=3)

# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = ridge2.predict(X_train1)

y_pred_test = ridge2.predict(X_test1)

metric2 = []
r2_train_lr = r2_score(y_train, y_pred_train) print(r2_train_lr)
metric2.append(r2_train_lr)

rss1_lr = np.sum(np.square(y_train - y_pred_train)) print(rss1_lr)

metric2.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test - y_pred_test)) print(rss2_lr) metric2.append(rss2_lr)

mse_train_lr = mean_squared_error(y_train,
y_pred_train) print(mse_train_lr)

metric2.append(mse_train_lr**0.5)

mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)

metric2.append(mse_test_lr**0.5)

#Alpha 1
#R2score(train) 0.884340040460635
#R2score(test) 0.869613280468847

<u>output :</u>
0.87973158109324
56

0.8710282148272899
607995142958.1411
320928407278.46216
680845624.8131479
729382743.8146868

2. R2 score on training data has decreased but it has increased on testing data Lasso

#Changed alpha 10 to 20
alpha =20
lasso20 = Lasso(alpha=alpha)
lasso20.fit(X_train1, y_train)

output :

Lasso(alpha=20)

```
# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = lasso20.predict(X_train1)

y_pred_test = lasso20.predict(X_test1)

metric3 = []
r2_train_lr = r2_score(y_train, y_pred_train)
print(r2_train_lr)
metric3.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)

print(r2_test_lr)
metric3.append(r2_test_lr)

rss1_lr = np.sum(np.square(y_train -
y_pred_train)) print(rss1_lr)

metric3.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test -
y_pred_test)) print(rss2_lr)

metric3.append(rss2_lr)
```

mse_train_lr = mean_squared_error(y_train,
y_pred_train) print(mse_train_lr)

metric3.append(mse_train_lr**0.5)

mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)

metric3.append(mse_test_lr**0.5)

#R2score at alpha-10
#0.885922240089900
5

#0.8646666084570094

<u>output :</u>

0.8854019697956436
0.8670105921065014
579329522996.7144
330925704432.26794
648745266.5136778
752103873.7096999

R2 score of training data has decrease and it has increase on testing data
<u>output :</u>

#important predictor variables
betas =
pd.DataFrame(index=X_train1.columns)
betas.rows = X_train1.columns

betas['Ridge2'] =

ridge2.coef_ betas['Ridge'] =

ridge.coef_ betas['Lasso'] =

lasso.coef_

betas['Lasso20'] = lasso20.coef_

pd.set_option('display.max_rows', None)

betas.head(68)

OverallQual	106429.293471	115599.252408	119957.483345	121719.072148
OverallCond	30969.119664	35638.745398	37354.981812	36948.765235
YearBuilt	53872.884932	54545.692314	53864.332906	53764.548095
BsmtFinSF1	53388.964692	51586.657410	50216.539701	50458.153814
TotalBsmtSF	71811.348552	76674.754264	78348.099735	78209.333502
1stFlrSF	70196.443400	73061.086063	8832.898863	8244.958141
2ndFlrSF	33666.888170	37149.879346	0.000000	0.000000
GrLivArea	83295.309506	87839.676484	163982.920640	162804.680303
BedroomAbvGr	-38094.981167	-52962.603870	-62831.358381	-61134.170375
TotRmsAbvGrd	54102.652478	52937.952456	51280.023696	50757.774874
Street_Pave	34001.153057	49959.412426	63045.460825	59515.001052
LandSlope_Sev	-17857.132747	-27846.862924	-37188.510825	-29661.614776
Condition2_PosN	-3031.699352	-11908.785655	-21920.323877	-11645.855795
RoofStyle_Shed	5474.383816	11641.731102	17801.452620	1966.058339
RoofMatl_Metal	8130.068994	18201.049929	32845.684073	16580.031007
Exterior1st_Stone	-17057.383837	-37132.047065	-69633.615929	-59674.587283
Exterior2nd_CBlock	-15569.072249	-32941.699298	-60463.906721	-49678.514531
ExterQual_Gd	-49400.503457	-54900.543840	-58459.152105	-57016.336034
ExterQual_TA	-59179.903853	-62317.508218	-64902.622534	-63508.829030
BsmtCond_Po	-4343.870481	-2488.039788	0.000000	-0.000000
KitchenQual_TA	-7060.140437	-5437.664855	-4495.491440	-4450.468043
Functional_Maj2	-10968.231950	-23574.925049	-40743.007254	-31654.783158
SaleType_CWD	-16897.367011	-27224.575631	-35460.118834	-30830.830798
SaleType_Con	13636.660731	21036.193759	25659.755739	21222.403113

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
RoofMatl_Metal Roof material_Metal
Predictors are same but the coefficient of these predictor has changed

Question 2

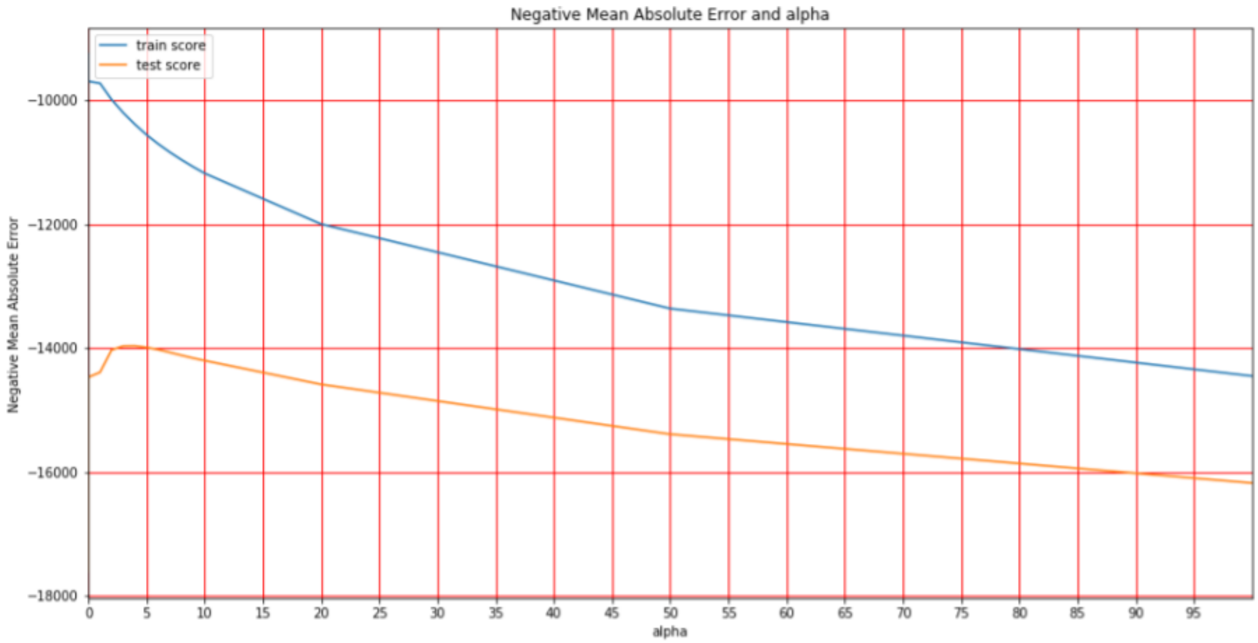
You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answer :

We would decide that on the basis of plots and choose a value of alpha where we have good training as well as the test score.

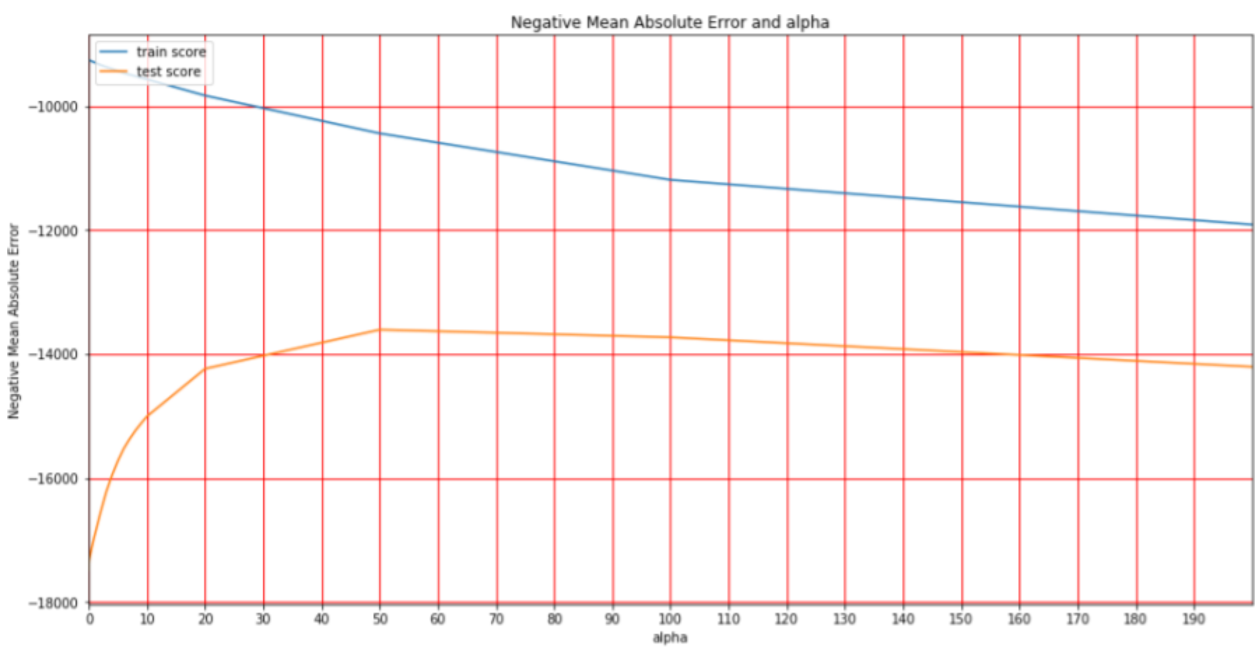
Ridge regression plot:

Based on the plot, we choose 4 as the value for lambda for Ridge Regression, since it has the best train as well as the test score.



Lasso Regression Plot:

Based on the plot, we choose 50 as the value for lambda for Lasso Regression, since it has the best train as well as the test score.



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 :

X_train1

	LotArea	OverallQual	OverallCond	YearBuilt	BsmtFinSF1	TotalBsmtSF	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd
1108	0.187723	0.555556	0.500	0.932836	0.000000	0.288210	0.170306	0.460583	0.407819	0.500000	0.444444
745	0.213431	0.777778	1.000	0.753731	0.262797	0.356207	0.252911	0.955928	0.753286	0.666667	0.888889
1134	0.208004	0.555556	0.500	0.910448	0.000000	0.285714	0.158661	0.424581	0.377486	0.500000	0.444444
512	0.217344	0.444444	0.500	0.619403	0.238117	0.269495	0.139738	0.000000	0.129424	0.500000	0.222222
43	0.220201	0.444444	0.625	0.746269	0.127971	0.292576	0.166667	0.000000	0.154365	0.500000	0.222222
33	0.258819	0.444444	0.500	0.626866	0.465265	0.436057	0.443959	0.000000	0.411190	0.666667	0.333333
269	0.183553	0.555556	0.750	0.753731	0.343236	0.356519	0.230349	0.000000	0.213347	0.500000	0.333333
789	0.306036	0.555556	0.875	0.679104	0.259598	0.259513	0.180495	0.689634	0.541625	0.833333	0.666667
1038	0.001200	0.333333	0.625	0.708955	0.000000	0.170306	0.115721	0.338920	0.291203	0.500000	0.333333
151	0.354195	0.777778	0.500	0.985075	0.639854	0.533375	0.447598	0.000000	0.414560	0.333333	0.333333
344	0.031449	0.444444	0.250	0.753731	0.058958	0.167187	0.020378	0.357542	0.213010	0.500000	0.111111
1218	0.135651	0.333333	0.500	0.537313	0.000000	0.000000	0.069869	0.148976	0.145602	0.333333	0.000000
1040	0.332315	0.444444	0.375	0.611940	0.076782	0.353712	0.481441	0.000000	0.445905	0.500000	0.555556
688	0.188466	0.777778	0.625	0.985075	0.431901	0.442608	0.341703	0.000000	0.316481	0.333333	0.444444
1289	0.273473	0.777778	0.500	0.977612	0.000000	0.338428	0.232897	0.527623	0.502191	0.500000	0.555556
1459	0.241252	0.444444	0.625	0.671642	0.379342	0.391765	0.282387	0.000000	0.261544	0.500000	0.333333
1448	0.293525	0.333333	0.750	0.261194	0.000000	0.174672	0.114993	0.341403	0.291877	0.333333	0.333333
733	0.243052	0.444444	0.625	0.641791	0.271481	0.269495	0.241630	0.000000	0.223795	0.500000	0.333333
3	0.230198	0.666667	0.500	0.298507	0.098720	0.235808	0.175036	0.469274	0.416919	0.500000	0.444444
123	0.182839	0.555556	0.500	0.880597	0.137112	0.373986	0.261645	0.000000	0.242332	0.333333	0.222222
812	0.206261	0.444444	0.500	0.574627	0.000000	0.168434	0.205240	0.000000	0.190091	0.333333	0.111111
1258	0.231255	0.666667	0.500	0.970149	0.299360	0.266999	0.249636	0.000000	0.231210	0.333333	0.222222
929	0.328915	0.666667	0.500	0.910448	0.000000	0.300686	0.186681	0.771570	0.591844	0.666667	0.555556

Y_train

1108	181000
745	299800
1134	169000
512	129900
43	130250
33	165500
269	148000
789	187500
1038	97000
151	372402
344	85000
1218	80500
1040	155000
688	392000
1289	281000
1459	147500
1448	112000
733	131400
3	140000
123	153900
812	55993
1258	190000
929	222000
1348	215000
692	335000
1014	119200
412	222000
1425	142000
497	184000
603	151000

X_train1.columns

Index(['LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'BsmtFinSF1', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea', 'BedroomAbvGr', 'TotRmsAbvGrd', 'Street_Pave', 'LandSlope_Sev', 'Condition2_PosN', 'RoofStyle_Shed', 'RoofMatl_Metal', 'Exterior1st_Stone', 'Exterior2nd_CBlock', 'ExterQual_Gd', 'ExterQual_TA', 'BsmtCond_Po', 'KitchenQual_TA', 'Functional_Maj2', 'SaleType_CWD', 'SaleType_Con'], dtype='object')
LotArea,OverallQual,YearBuilt,BsmtFinSF1,TotalBsmtSF are the top 5 important predictors.

Let's drop these columns

X_train2 =
X_train1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],axis=1) X_test2
= X_test1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],axis=1)

X_train2.head()

	OverallCond	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd	Street_Pave	LandSlope_Sev	Condition2_PosN	RoofStyle_She
1108	0.500	0.170306	0.460583	0.407819	0.500000	0.444444	1	0	0	
745	1.000	0.252911	0.955928	0.753286	0.666667	0.888889	1	0	0	
1134	0.500	0.158661	0.424581	0.377486	0.500000	0.444444	1	0	0	
512	0.500	0.139738	0.000000	0.129424	0.500000	0.222222	1	0	0	
43	0.625	0.166667	0.000000	0.154365	0.500000	0.222222	1	0	0	

X_test2.head()

	OverallCond	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd	Street_Pave	LandSlope_Sev	Condition2_PosN	RoofStyle_She
990	0.50	0.337336	0.611421	0.644422	0.5	0.444444	1	0	0	
1161	0.75	0.422125	0.000000	0.390967	0.5	0.444444	1	0	0	
1369	0.50	0.432314	0.000000	0.400404	0.5	0.555556	1	0	0	
329	0.50	0.042213	0.369957	0.239973	0.5	0.333333	1	0	0	
262	0.75	0.266376	0.000000	0.246714	0.5	0.333333	1	0	0	

Lasso

# alpha 10
alpha =10
lasso21 = Lasso(alpha=alpha)
lasso21.fit(X_train2, y_train)

<u>output :</u>
Lasso(alpha=10)

# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = lasso21.predict(X_train2)

```
y_pred_test = lasso21.predict(X_test2)


metric3 = []
r2_train_lr = r2_score(y_train,
y_pred_train) print(r2_train_lr)
metric3.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
metric3.append(r2_test_lr)

rss1_lr = np.sum(np.square(y_train - y_pred_train))
print(rss1_lr)

metric3.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test - y_pred_test))
print(rss2_lr)

metric3.append(rss2_lr)


mse_train_lr = mean_squared_error(y_train, y_pred_train)
print(mse_train_lr)

metric3.append(mse_train_lr**0.5)


mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)

metric3.append(mse_test_lr**0.5)
#R2 Score at alpha-10
#0.8859222400899005

#0.8646666084570094
```

output :

0.7988346707068132

0.758810320925813

1016954777102.8657

600167078819.8159

1138807141.2126155

1364016088.2268543

R2 score of training and testing data has decreased
#important predictor variables
betas =

pd.DataFrame(index=X_train2.columns)

betas.rows = X_train1.columns

betas['Lasso21'] = lasso21.coef_

pd.set_option('display.max_rows', None)

betas.head(68)

	Lasso21
OverallCond	7403.774043
1stFlrSF	163379.262938
2ndFlrSF	12227.759048
GrLivArea	186638.919740
BedroomAbvGr	-71218.036474
TotRmsAbvGrd	41610.305613
Street_Pave	101376.262107
LandSlope_Sev	-40205.679947
Condition2_PosN	0.000000
RoofStyle_Shed	53262.728685
RoofMatl_Metal	84219.173436
Exterior1st_Stone	-124162.644239
Exterior2nd_CBlock	-139534.253019
ExterQual_Gd	-77170.982079
ExterQual_TA	-108569.936019
BsmtCond_Po	-122646.594039
KitchenQual_TA	-11135.858324
Functional_Maj2	-48462.215856
SaleType_CWD	-64725.438438
SaleType_Con	52937.625483

Five most important predictor variables

- 11stFlrSF First Floor square feet
- GrLivArea Above grade (ground) living area square feet
- Street_Pave Pave road access to property
- RoofMatl_Metal Roof material_Metal
- RoofStyle_Shed Type of roof(Shed)

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 :

A model is considered to be robust if the model is stable, that does not change drastically upon changing the training set. The model is considered generalisable if it does not overfit the training data, and works well with new data. Its implication in terms of accuracy is that a robust and generalisable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data.