Advanced Regression-Subjective Questions-

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.a. The optimal value of alpha for ridge and lasso regression

Ridge Alpha 1

Lasso Alpha 10

When we double the value of alpha for our ridge regression number we will take the value of alpha equal to 3 the model will apply more penalty on the curve and try to make the model more generalized that is making model more simpler and not thinking to fit every data of the data set.We can see below that when alpha is 3 we get more error for both test and train. Similarly when we increase the value of alpha for lasso we try to penalize more our model and more coefficient of the variable will reduced to zero, when we increase the value of our r2 square also decreases

Ridge Regression

```
#Taking the alpha as 3 for second part of the question
alpha=3
ridge2=Ridge(alpha=alpha)
ridge2.fit(X_train1,y_train)
```

Out[262]:

```
Ridge(alpha=3)
```

```
#Calculating R2 score,RSS,RMSE metrics
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)

r2_test_lr=r2_score(y_test,y_pred_test)
print(r2_test_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)
```

```
0.8797315810932455
0.87102821482729
607995142958.1414
320928407278.462
680845624.8131483
729382743.8146863
```

Answer 1.b. From above metrics you'll observe that R2 Score is high on test data but low on training data.This is the effect of doubing the value of alpha in Ridge Regression.

Lasso

```
# Answer 1.b -Increasing alpha from 10 to 20
alpha=20
lasso20=Lasso(alpha=alpha)
lasso20.fit(X_train1,y_train)
```

Out[264]:

```
Lasso(alpha=20)
```

```
#Calculating R2 score,RSS,RMSE metrics
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)
metric.append(r2_train_lr)

r2_test_lr=r2_score(y_test,y_pred_test)
print(r2_test_lr)
metric3.append(r2_train_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)
```

```
0.8854019697956436
0.8670105921065014
579329522996.7144
330925704432.26794
648745266.5136778
752103873.7096999
```

Answer 1.b. From above metrics you'll observe that R2 Score is high on test data but low on training data.This is the effect of doubing the value of alpha in Lasso Regression.

```
#Answer 1.c. Finding out signifiant predictor variables
sigvar=pd.DataFrame(index=X_train1.columns)
sigvar.rows=X_train1.columns
sigvar['Ridge2']=ridge2.coef_
sigvar['Ridge']=ridge.coef_
sigvar['Lasso']=lasso.coef_
sigvar['Lasso20']=lasso20.coef_
pd.set_option('display.max_rows',None)
sigvar.head(68)
```

Out[266]:

	Ridge2	Ridge	Lasso	Lasso20
LotArea	52892.418502	59778.431939	63955.064210	63617.887669
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

Answer 1.c.-

- 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

From above results, we observed that predictors are same but the coefficent of these predictor has modified.
Notes- It is important to regularize coefficients and improve the prediction accuracy also with the decrease in variance, and making the model interpretably.

Ridge regression, uses a tuning parameter called lambda as the penalty is square of magnitude of coefficients which is identified by cross validation. Residual sum or squares should be small by using the penalty. The penalty is lambda times sum of squares of the coefficients, hence the coefficients that have greater values gets penalized. As we increase the value of lambda the variance in model is dropped and bias remains constant. Ridge regression includes all variables in final model unlike Lasso Regression.

Lasso regression, uses a tuning parameter called lambda as the penalty is absolute value of magnitude of coefficients which is identified by cross validation. As the lambda value increases Lasso shrinks the coefficient towards zero and it make the variables exactly equal to 0. Lasso also does variable selection. When lambda value is small it performs simple linear regression and as lambda value increases, shrinkage takes place and variables with 0 value are neglected by the model.

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?

We will check R2 Score of all the models.From the following data The r2_score of lasso is slightly greater than the lasso for the test dataset so we will choose lasso regression to solve this problem.Secondly,Lasso will be a better model as it will help in feature elimintation and will be more robust.

```
         Ridge Regression              Lasso Regression
```

R2 score(Train)--------- 0.88 ---------------------------0.88

R2 score(Test)-----------0.87----------------------------0.86

final_metric

```
In [267]: final_metric
Out[267]:
```

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	R2 Score (Train)	8.861162e-01	8.843400e-01	8.859222e-01
1	R2 core (Test)	8.621985e-01	8.696133e-01	8.646666e-01
2	RSS (Train)	5.757188e+11	5.846979e+11	5.766994e+11
3	RSS (Test)	3.429000e+11	3.244493e+11	3.367584e+11
4	MSE (Train)	2.539098e+04	2.558822e+04	2.541260e+04
5	MSE (Test)	2.791627e+04	2.715483e+04	2.766514e+04

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-Top 5 most important predictor variables are as below-

- 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)

In [268]:
```python
#Printing X_train1
X_train1
```

Out[268]:

	LotArea	OverallQual	OverallCond	YearBuilt	BsmtFinSF1	TotalBsmtSF	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr
1108	0.187723	0.555556	0.500	0.932836	0.000000	0.288210	0.170306	0.460583	0.407819	0.500000
746	0.213431	0.777778	1.000	0.753731	0.262797	0.356207	0.252911	0.955928	0.753286	0.666667
1134	0.208004	0.555556	0.500	0.910448	0.000000	0.285714	0.158661	0.424581	0.377486	0.500000
512	0.217344	0.444444	0.500	0.619403	0.238117	0.269495	0.139738	0.000000	0.129424	0.500000
43	0.220201	0.444444	0.625	0.746269	0.127971	0.292576	0.166667	0.000000	0.154365	0.500000
33	0.258819	0.444444	0.500	0.626866	0.465265	0.436057	0.443959	0.000000	0.411190	0.666667
269	0.183553	0.555556	0.750	0.753731	0.343236	0.356519	0.230349	0.000000	0.213347	0.500000
789	0.306036	0.555556	0.875	0.679104	0.259598	0.259513	0.180495	0.689634	0.541625	0.833333
1038	0.001200	0.333333	0.625	0.708955	0.000000	0.170306	0.115721	0.338920	0.291203	0.500000
151	0.354195	0.777778	0.500	0.985075	0.639854	0.533375	0.447598	0.000000	0.414560	0.333333

In [269]:
```python
#Printing y_train
y_train
```

Out[269]:
```
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
```

In [270]:
```
#Printing X_train1 Columns
X_train1.columns
```

Out[270]:
```
Index(['LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'BsmtFinSF1', 'TotalBsmtSF', '1stFlrSF', '
'BedroomAbvGr', 'TotRmsAbvGrd', 'Street_Pave', 'LandSlope_Sev', 'Condition2_PosN', 'RoofStyle_Shed', 'R
r1st_Stone', 'Exterior2nd_CBlock', 'ExterQual_Gd', 'ExterQual_TA', 'BsmtCond_Po', 'KitchenQual_TA', 'Fu
e_CWD', 'SaleType_Con'], dtype='object')
```

LotArea,OverallQual,YearBuilt,BsmtFinSF1,TotalBsmtSF are the top 5 important predictor variables.

Let's drop these columns.

In [271]:
```
X_train2=X_train1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],axis=1)
X_test2 = X_test1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],axis=1)
```

In [272]:
```
X_train2.head()
```

Out[272]:

	OverallCond	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd	Street_Pave	LandSlope_Sev	Condition2_Po
1108	0.500	0.170306	0.460583	0.407819	0.500000	0.444444	1	0	
746	1.000	0.252911	0.955928	0.753286	0.666667	0.888889	1	0	
1134	0.500	0.158661	0.424581	0.377486	0.500000	0.444444	1	0	
512	0.500	0.139738	0.000000	0.129424	0.500000	0.222222	1	0	
43	0.625	0.166667	0.000000	0.154365	0.500000	0.222222	1	0	

In [273]:
```
X_test2.head()
```

Out[273]:

	OverallCond	1stFlrSF	2ndFlrSF	GrLivArea	BedroomAbvGr	TotRmsAbvGrd	Street_Pave	LandSlope_Sev	Condition2_Po
990	0.50	0.337336	0.611421	0.644422	0.5	0.444444	1	0	
1161	0.75	0.422125	0.000000	0.390967	0.5	0.444444	1	0	
1369	0.50	0.432314	0.000000	0.400404	0.5	0.555556	1	0	

Performing Lasso

In [274]:
```
#Taking alpha as 10
alpha=10
lasso21=Lasso(alpha=alpha)
lasso21.fit(X_train2,y_train)
```

Out[274]:
```
Lasso(alpha=10)
```

In [275]:
```
#Calulating R2 Score,RSS,RMSE Metrics
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)
```

```
0.7988346707068132
0.758810320925813
```

```
0.7988346707068132
0.758810320925813
1016954777102.8657
600167078819.8159
1138807141.2126155
1364016088.2268543
```

As we can see that training and testing dataset's R2 Score has decreased

In [276]:

```
#important predictor variables
sigvar = pd.DataFrame(index=X_train2.columns)
sigvar.rows = X_train1.columns
sigvar['Lasso21'] = lasso21.coef_
pd.set_option('display.max_rows', None)
sigvar.head(68)
```

Out[276]:

	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

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

Answer-Top 5 most important predictor variables are as below-

- 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?

The model should be generalized so that the test accuracy is not lower than the training score. The model should be accurate for datasets other than the ones which were used during training. Don't give importance to the outliers so that the accuracy predicted by the model is high. That is why, the outliers analysis needs to be done and we need to retain only those values which are relevant to the dataset. The outliers which are not relevant must be removed from the dataset. If the model is not robust, it cannot be trusted for predictive analysis.

The model should be as simple as possible, though its accuracy will decrease but it will be more

robust and generalisable. It can be also understood in the terms of the Bias-Variance trade-off. The simpler the

model the more the bias but less variance and more generalizable it is. 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.

Bias: Bias is error in model, when the model is weak to learn from the data. High bias means model is

unable to learn details in the data. Model performs poor on training and testing data.

Variance: Variance is error in model, when model tries to over learn from the data. High variance means

model performs exceptionally well on training data as it has very well trained on this of data but

performs very poor on testing data as it was unseen data for the model.

It is important to have balance in Bias and Variance to avoid overfitting and under-fitting of data.