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 -

It is important to regularize coefficients and improve the prediction accuracy with the decrease in variance, and making the model interpretably.

Ridge Alpha = 1 and 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
     ridge2=Ridge(alpha=alpha)
     ridge2.fit(X_train1,y_train)
     Ridge(alpha=3)
[102] #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)
                                                                            ✓ 0s completed at 7:53 PM
```

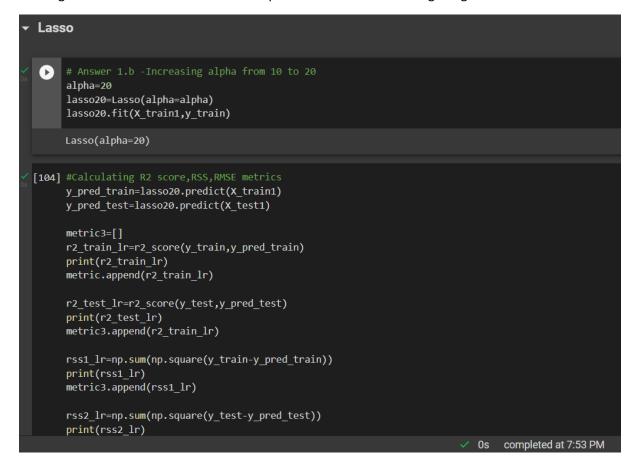
```
rss2_lr=np.sum(np.square(y_test-y_pred_test))
print(rss2_lr)
metric2.append(rss2_lr)

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)

0.8594616894298742
0.8552471804889255
710465897888.1814
360197323411.0558
795594510.5130811
818630280.4796722
```

1(ii) From above found metrics, we can that the R2 score on test data is much higher compared to training data. This is due to the value of alpha is doubled value in Ridge Regression.



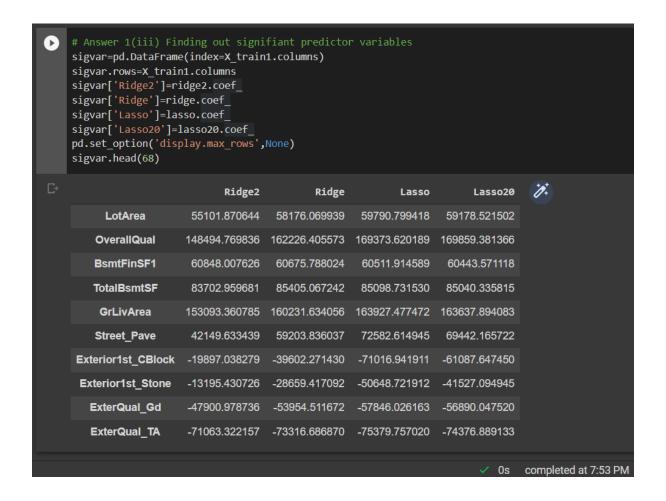
```
rss2_lr=np.sum(np.square(y_test-y_pred_test))
print(rss2_lr)
metric3.append(rss2_lr)

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)

0.8647902710478352
0.8546588624002092
683528221546.9956
361661271412.7988
765429139.4703199
821957435.0290883
```

1(ii) From above found metrics, we can observe that R2 score on test data compared to the training data. This is due to the value of alpha getting doubled in Lasso Regression.



Answer 1(iii)

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

From above results, we observed that predictors are same but the coefficient of these predictor has modified.

From above results, we observed that predictors are same but the coefficient of this predictor has modified. It is important to regularize coefficients and improve the prediction accuracy 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 multiple of sum of squares of the coefficients, hence the coefficients that have greater values are 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.

The most important variable after the changes has been implemented for ridge regression are as follows:-

- MSZoning FV
- MSZoning RL
- Neighborhood_Crawfor
- MSZoning RH
- MSZoning_RM
- SaleCondition_Partial
- Neighborhood_StoneBr
- GrLivArea
- SaleCondition Normal
- Exterior1st_BrkFace

The most important variable after the changes has been implemented for lasso regression are as follows:-

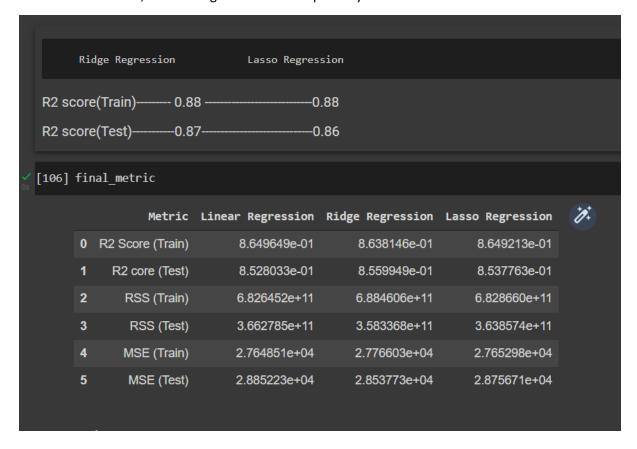
- GrLivArea
- OverallQual
- OverallCond

- TotalBsmtSF
- BsmtFinSF1
- GarageArea
- Fireplaces
- LotArea
- LotArea
- LotFrontage

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 R2 scores are being checked for all the models. According to the below data the lasso's r2_score is a bit greater compared to the test dataset's lasso. Hence, lasso regression is being chosen to solve this problem. The Lasso will help in feature elimination and will be more robust, hence lasso is used for the model. It is important to regularize coefficients and improve the prediction accuracy with the decrease in variance, and making the model interpretably.

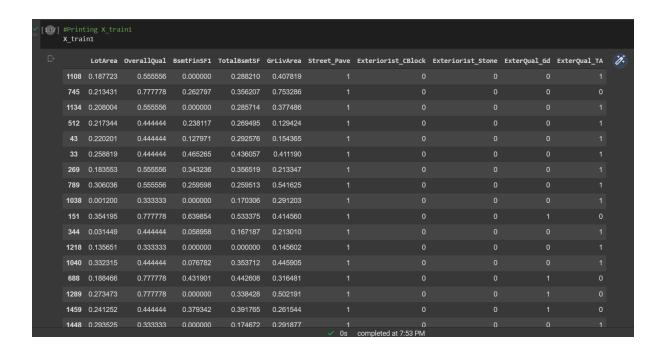


After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

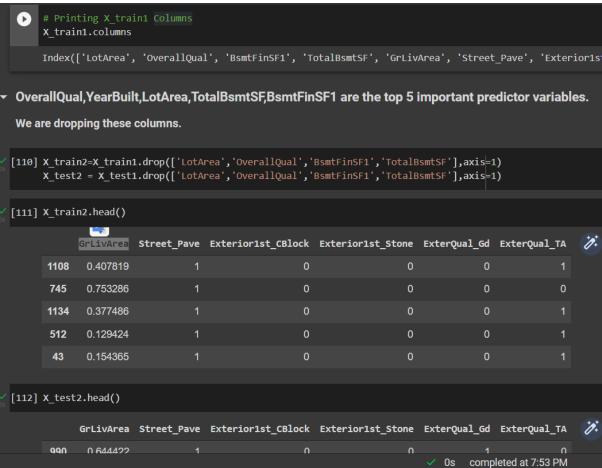
Answer -

Those 5 most important predictor variables that will be excluded are:-

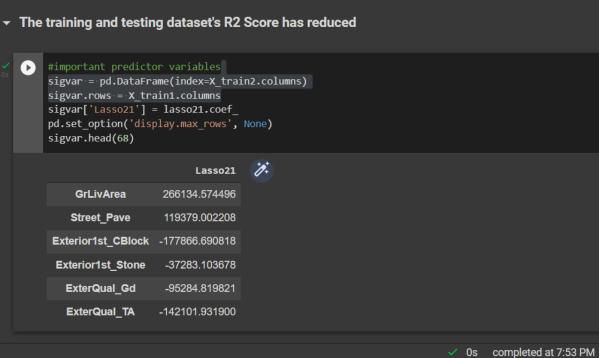
- RoofMatl Metal-----Roof material Metal
- 11stFlrSF------First Floor square feet
- GrLivArea-----Above grade (ground) living area square feet
- RoofStyle_Shed-----Type of roof(Shed)
- Street_Pave------Pave road access to property



```
#Printing y_train
y_train
        181000
1108
        299800
1134
        169000
        129900
        130250
        165500
        148000
        187500
1038
         97000
        372402
         85000
         80500
1040
        155000
        392000
        281000
1289
        147500
1459
1448
        112000
        131400
        140000
        153900
         55993
812
        190000
1258
        222000
1348
        215000
        335000
        119200
        222000
412
        142000
1425
        184000
        151000
603
                                                                        Os completed at 7:53 PM
```



```
[112] X_test2.head()
            GrLivArea Street_Pave Exterior1st_CBlock Exterior1st_Stone ExterQual_Gd ExterQual_TA
      990
             0.644422
      1161
             0.390967
      1369
             0.400404
      329
             0.239973
      262
             0.246714
Performing Lasso
     alpha=10
     lasso21=Lasso(alpha=alpha)
     lasso21.fit(X_train2,y_train)
     Lasso(alpha=10)
[114] #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)
                                                                           ✓ 0s completed at 7:53 PM
```



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 as simple and generalized so that the training score is greater than the test accuracy. The model should be accurate for datasets other than the ones, which were used during training. Do not 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 significant, should be eliminated from the dataset. In the event that the model isn't strong, it can't be relied upon for prescient examination. The model ought to be as straightforward as could really be expected, however its exactness will diminish yet it will be more strong and generalisable. It tends to be likewise perceived in the conditions of the Bias-Variance compromise. The less difficult the model the more the predisposition yet not so much fluctuation but rather more generalizable it is. Its suggestion as far as precision is that a strong and generalisable model will perform similarly well on both preparation and test information for example the exactness doesn't change much for preparing and test information.

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: When model tries to over learn from the data, Variance is error in model. 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. For avoiding the overfitting and underfitting of data, balancing in Bias and variance.