# **Problem Statement - Part II**

# **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 value of alpha for ridge and lasso regression:

For Ridge regression, alpha is 1.0

For Lasso regression, alpha is 0.0001

For Ridge regression alpha is 1.0 and now doubling it and making it 2.0.

For Ridge regression alpha is 1.0 and now doubling it and making it 2.0 In [125]: # Model building using optimal alpha
ridge\_modified = Ridge(alpha=2.0) ridge\_modified.fit(X\_train, y\_train) Out[125]: Ridge(alpha=2.0, copy\_X=True, fit\_intercept=True, max\_iter=None, normalize=False, random\_state=None, solver='auto', tol=0.001) In [126]: #creating coeffcients for the ridge regression model\_parameter = list(ridge.coef\_) model\_parameter.insert(0,ridge.intercept\_) cols = house train.columns cols.insert(0,'const') ridge\_coef = pd.DataFrame(list(zip(cols,model\_parameter,(abs(ele) for ele in model\_parameter)))) ridge\_coef.columns = ['Features','Coefficient','Mod']
#selecting the top 10 variables ridge\_coef.sort\_values(by='Mod',ascending=False).head(10) Out[126]: Features Coefficient 0 LotFrontage 10.261566 10.261566 OverallCond 0.527282 0.527282 3 14 BsmtUnfSF 0.406426 0.406426 12 BsmtFinType2 0.355562 0.355562 2 OverallQual 0.338037 0.338037 11 BsmtFinSF1 0.322512 0.322512 9 BsmtExposure 0.316265 0.316265 33 GarageFinish 0.303286 0.303286 73 LotConfig\_CulDSac -0.272453 0.272453 ExterCond 0.264079 0.264079 In [127]: y\_train\_pred = ridge\_modified.predict(X\_train) y\_test\_pred = ridge\_modified.predict(X\_test) print("Ridge Regression train r2:",r2\_score(y\_true=y\_train,y\_pred=y\_train\_pred)) print("Ridge Regression test r2:",r2\_score(y\_true=y\_test,y\_pred=y\_test\_pred)) Ridge Regression train r2: 0.9238505663513401

For Lasso regression alpha is 0.0001 and not doubling it and making it 0.0002.

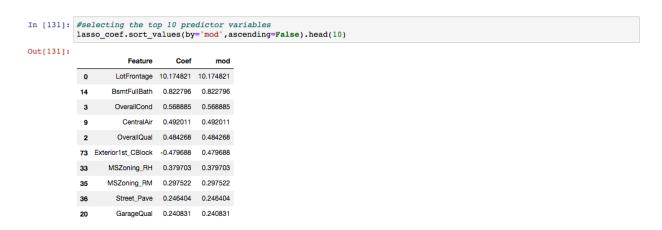
For Lasso regression alpha is 0.0001 and now doubling it and making it 0.0002

Ridge Regression test r2: 0.7526833276310241

For all the models, the training score has decreased slightly and the testing score has increased slightly.

Most important predictor variables after the change is implemented are:

- LotFrontage
- BsmtFullBath
- OverallCond
- CentralAir
- OverallQual
- Exterior1st CBlock
- MSZoning RH
- MSZoning\_RM
- Street Pave
- GarageQual



# **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:**

Even though Ridge has given good performance(R-Square value), I would choose the Lasso model for the following reasons:

- Lasso regression would help in feature elimination and the model will be more robust.
- Model is giving decent performance.
- Efficiently solved the high dimensionality problem by shrinking insignificant coefficients to zero.
- Simpler model and easy for maintenance.

## **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:

The five most important predictor variables now are:

- LotArea
- FullBath
- 1stFlrSF
- ExterCond
- MSZoning\_RH

```
: #selecting the top 5 predictor variables
  lasso_coef.sort_values(by='mod',ascending=False).head(5)
           Feature
                      Coef
                                mod
           LotArea 10.205140 10.205140
   0
   10
           FullBath
                  1.029837
                           1.029837
          1stFlrSF 0.606857
                           0.606857
   1
         ExterCond 0.561906
                           0.561906
   28 MSZoning_RH 0.512930 0.512930
```

# **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 demonstrates robustness when its performance remains largely unchanged in response to any variations in the data.
- A model with good generalizability can effectively adjust to new, previously unseen data that originates from the same distribution as the one used during model creation.
- To make sure a model is robust and generalizable, we have to take care it
  doesn't overfit. Overfitting occurs when a model becomes overly complex,
  leading to high variance and excessive sensitivity to minor fluctuations in the
  data. Consequently, an overfitted model may accurately capture patterns present
  in the training data but struggle to generalize to unseen test data.
- In other words, the model should not be too complex in order to be robust and generalizable.
- If we look at it from the perspective of Accuracy, a complex model will have a
  very high accuracy. So, to make our model more robust and generalizable, we
  will have to decrease variance which will lead to some bias. Addition of bias
  means that accuracy will decrease.
- Striking a balance between model accuracy and complexity is paramount.
   Regularization techniques such as Ridge Regression and Lasso offer effective means of achieving this balance. By imposing constraints on model parameters, regularization techniques help prevent overfitting, thereby promoting robustness and generalizability while tempering the potential decrease in accuracy associated with bias introduction.