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 regularization parameter, alpha, is a hyperparameter used for reducing overfitting by introducing bias in the model. Alpha can be any number from 0 to +ive infinity. Higher the value of alpha, higher the bias it introduces in the model or in other words, reduces the variance. But, there has to be some limit to which alpha can be increased before it causes the model to start underfitting. The optimal value of alpha is where bias variance trade-off is balanced. To find it, we use cross validation technique in which we run the model with a set of alpha values and then check which alpha value is giving the lowest mean absolute error for both train and test. This regularization parameter concept is same for Ridge and Lasso regression.

In the assignment, I got the following optimal alpha values: -

For Ridge Regression, optimal alpha = 2.0

For Lasso Regression, optimal alpha = 0.0001

If we choose to double the value of alpha, i.e., for Ridge, alpha = 4.0 and for Lasso, alpha = 0.0002, then following changes have occurred: -

1. The absolute values of the feature coefficients have gone down for both Ridge and Lasso models. It makes sense, as we have introduced more bias in the model by increasing alpha.
2. Ridge: -
   Train R2 score reduced from 0.8886 to 0.8844
   Test R2 score reduced from 0.8713 to 0.8691
   Lasso: -
   Train R2 score is reduced from 0.8919 to 0.8905
   Test r2 score is changed from 0.8705 to 0.8715

After doubling the alpha value, following are the most important predictor variables: -

For Ridge: -

Coefficient	Column
0.274396	OverallQual_9
-0.211842	OverallQual_2
0.199962	OverallCond_9
-0.191648	Fireplaces_3
0.181219	GrLivArea
-0.179382	MSSubClass_160
0.161955	OverallQual_8
0.157428	OverallCond_8
0.141546	OverallCond_7
0.141460	Neighborhood_NridgHt
-0.139404	YearBuilt
0.135493	Neighborhood_Crawfor
-0.131284	ExterQual_Fa
-0.124559	Neighborhood_IDOTRR
0.121360	Neighborhood_ClearCr
-0.119493	BsmtExposure_No Basement
-0.114801	MSSubClass_180 0.114801
0.105789	Exterior1st_BrkFace
0.105717	BsmtExposure_Gd
-0.102510	Exterior1st_BrkComm

For Lasso: -

Coefficient	Column
-0.457293	OverallQual_2
0.397554	OverallQual_9
-0.319557	Fireplaces_3
0.290377	OverallCond_9
0.243267	OverallQual_8
0.208841	OverallCond_8
-0.206932	MSSubClass_160
0.187215	OverallCond_7
-0.180240	MSSubClass_180
0.179483	OverallQual_10
0.175760	GrLivArea
-0.169392	Exterior1st_BrkComm
-0.153746	ExterQual_Fa
0.146869	Neighborhood_ClearCr
0.140667	Neighborhood_Crawfor
0.139407	OverallQual_7
0.138271	OverallCond_6
0.138086	Neighborhood_NridgHt
-0.136311	YearBuilt
-0.125933	BsmtExposure_No Basement

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:

I will apply Lasso model because of the following reasons: -

1. R2 score of Lasso is slightly better than R2 score of Ridge.
2. I have chosen Lasso over Ridge because Ridge and Lasso have very similar R2 scores, but Lasso model is simpler as it has made the coefficient of OverallQual_5 as 0.0. Notice that OverallQual_5 had a very high VIF value as well. So, Lasso model has taken care of multicollinearity automatically. Ridge also reduced the coefficient value of features to make the model simpler, but it didn't delete the feature completely.

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 after removing top 5 features from my Lasso model are: -

1. Exterior1st_BrkComm: -0.355
2. OverallQual_3: -0.274
3. OverallQual_4: -0.252

4. MSSubClass_160: -0.226
5. OverallQual_5: -0.220254

Earlier the top 5 predictors of my Lasso model were: -

1. OverallQual_2 -0.484486
2. OverallQual_9 0.403385
3. OverallCond_9 0.349710
4. Fireplaces_3 -0.343539
5. OverallCond_8 0.260521

I deleted these and rebuilt the model to get the new top 5 features.

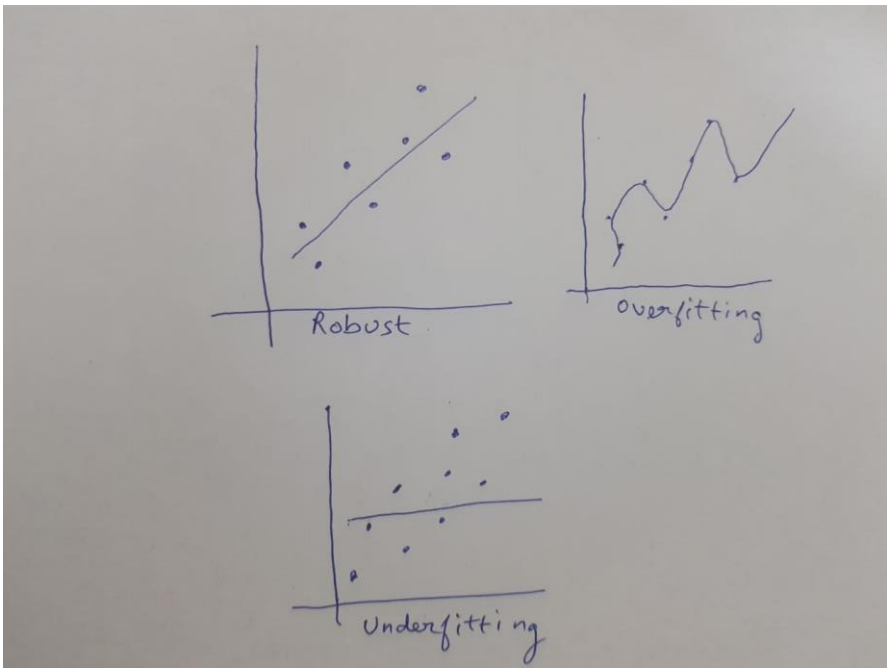
Question 4

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 robust and generalisable when it can perform on unseen Test data well. If the model is cannot perform well in the Test data but performed well in the Train data, then it shows that the model cannot generalise on unseen data and is hence, overfitting. When the model performs bad on Train and Test data both, then the model is underfitting.

An overfitting model tends to memorize all the data points available in the dataset and hence performs very well on Train data but fails to generalize its learning on unseen data. It can be seen from the below graphs: -



We can see from the above graphs, how an overfitting model is trying to memorize all the data points while an underfitting model is too naïve.

Hence, overfitting model become very sensitive to changes in training data and is also called a model with high variance and low bias. Similarly, an underfitting model has high bias and low variance. So, there's a trade off between these two because if we try to make the model too complicated, then it will tend to memorize every data point.

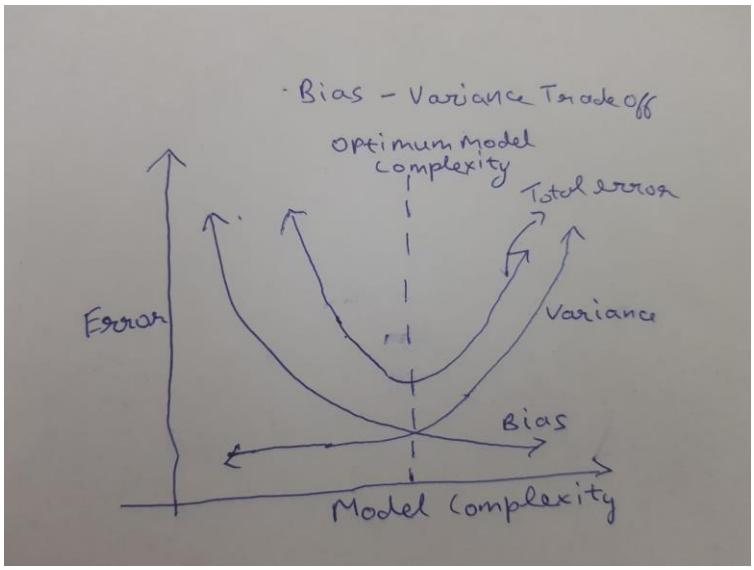
To achieve this trade-off between simplicity and complexity, we use a method called Regularization. Regularization is the process of deliberately simplifying models to achieve the correct balance between keeping the model simple and yet not too naive.

So, to make sure the model is robust and generalisable, we perform the following things: -

1. Use a Regularized regression technique like Lasso or Ridge.
2. Use Cross validation like k-fold cross validation to find the optimal regularization parameter, lambda, which would make the model just as complex as it's needed.
3. Separate some data points randomly for testing the model after building it using the above 2 steps. If train accuracy and test accuracy come up to be similar then we will be sure that our model is generalizing well on unseen test data.

Implications of generalizability and robustness on accuracy of the model will be: -

A generalizable and robust model will have to sacrifice some accuracy on the training set so that the test error is lowered. As we saw in the bias-variance trade-off diagram before, the optima is at the minimum of the total error. Total error here is the sum of test and train error.



So, a generalizable and robust model will have low total error but relatively higher train error and lower test error compared to a non-generalisable overfitting model in which Train error will be very low, but test error will be very high.