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 values of alpha when built on the complete prepared dataset is:

Ridge regression - 100.0

Lasso regression - 0.0025

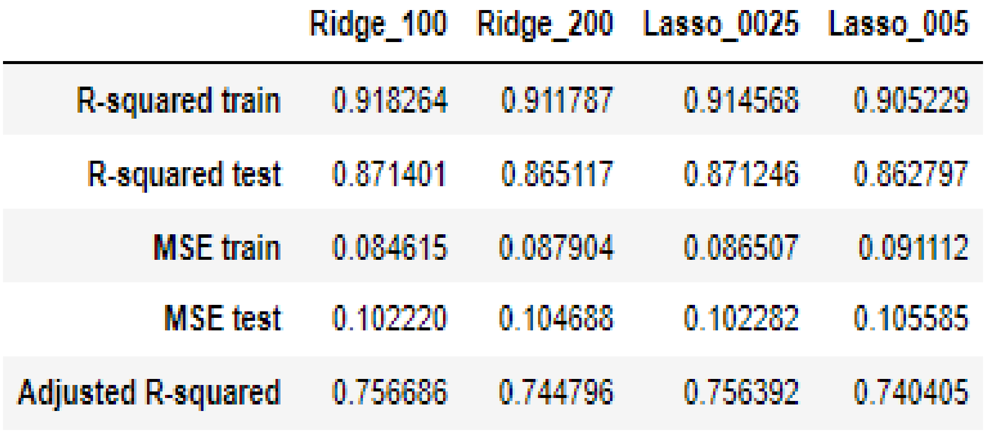
(Please refer to the 'Descriptive Questions' section of jupyter notebook for the code)

When the alpha values are doubled, we apply more regularization on the model and the model becomes less complex i.e., for ridge regression the features are pushed closer to zero and in lasso they can be even zero.

Post doubling, the alpha values will be:

Ridge regression - 200.0

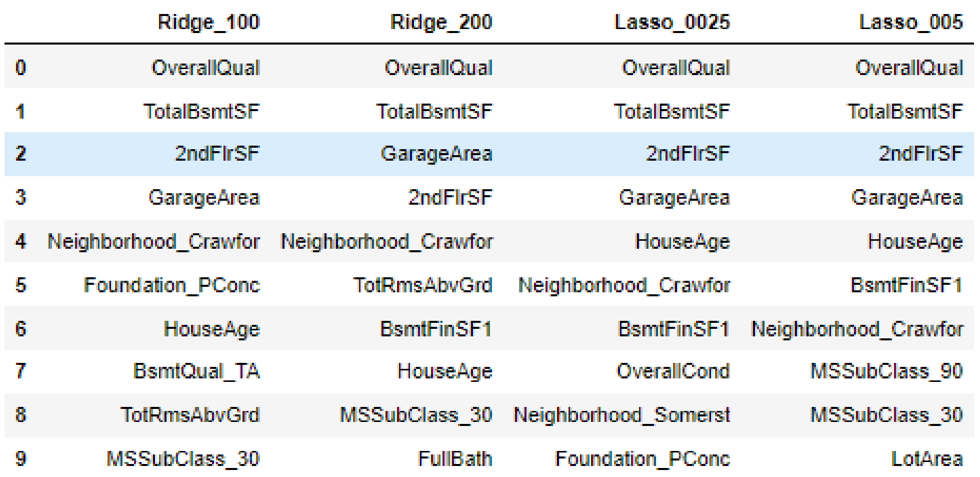
Lasso regression - 0.005



	Ridge_100	Ridge_200	Lasso_0025	Lasso_005
R-squared train	0.918264	0.911787	0.914568	0.905229
R-squared test	0.871401	0.865117	0.871246	0.862797
MSE train	0.084615	0.087904	0.086507	0.091112
MSE test	0.102220	0.104688	0.102282	0.105585
Adjusted R-squared	0.756686	0.744796	0.756392	0.740405

Above are the metrics achieved for different values of alpha for Ridge and Lasso regressions.

- As we can notice, as the value of alpha increases, the R-squared values for the ridge and lasso regression models has come down a bit.
- This is because, as the lambda increases, coefficients are pushed more towards zero and hence the bias increases. But the variance in the model will come down.



	Ridge_100	Ridge_200	Lasso_0025	Lasso_005
0	OverallQual	OverallQual	OverallQual	OverallQual
1	TotalBsmtSF	TotalBsmtSF	TotalBsmtSF	TotalBsmtSF
2	2ndFlrSF	GarageArea	2ndFlrSF	2ndFlrSF
3	GarageArea	2ndFlrSF	GarageArea	GarageArea
4	Neighborhood_Crawfor	Neighborhood_Crawfor	HouseAge	HouseAge
5	Foundation_PConc	TotRmsAbvGrd	Neighborhood_Crawfor	BsmtFinSF1
6	HouseAge	BsmtFinSF1	BsmtFinSF1	Neighborhood_Crawfor
7	BsmtQual_TA	HouseAge	OverallCond	MSSubClass_90
8	TotRmsAbvGrd	MSSubClass_30	Neighborhood_Somerst	MSSubClass_30
9	MSSubClass_30	FullBath	Foundation_PConc	LotArea

-

- Above are the top 10 most important predictor variables before and after doubling the alpha value.

- As we can notice, post doubling, few of the features becomes less significant (as their values are pushed towards zero) and new features make the top 10 significant features list.

- Top 3 features are almost same for all the models, as these are the features with highest coefficient/ significant variables, and they remain to be significant ones after the regularization. But few other features took a hit after doubling and they are no more in the top 10 significant features list.

- **Ridge top 10 features with optimal alpha** - 'OverallQual', 'TotalBsmtSF', '2ndFlrSF', 'GarageArea','Neighborhood_Crawfor', 'Foundation_PConc', 'HouseAge','BsmtQual_TA', 'TotRmsAbvGrd', 'MSSubClass_30'

- **Ridge top 10 features with double alpha** - 'OverallQual', 'TotalBsmtSF', 'GarageArea', '2ndFlrSF', 'Neighborhood_Crawfor', 'TotRmsAbvGrd', 'BsmtFinSF1', 'HouseAge', 'MSSubClass_30', 'FullBath'

- **Lasso top 10 features with optimal alpha** - 'OverallQual', 'TotalBsmtSF', '2ndFlrSF', 'GarageArea', 'HouseAge', 'Neighborhood_Crawfor', 'BsmtFinSF1', 'OverallCond', 'Neighborhood_Somerst', 'Foundation_PConc'

- **Lasso top 10 features with double alpha** - 'OverallQual', 'TotalBsmtSF', '2ndFlrSF', 'GarageArea', 'HouseAge', 'BsmtFinSF1', 'Neighborhood_Crawfor', 'MSSubClass_90', 'MSSubClass_30', 'LotArea'

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?

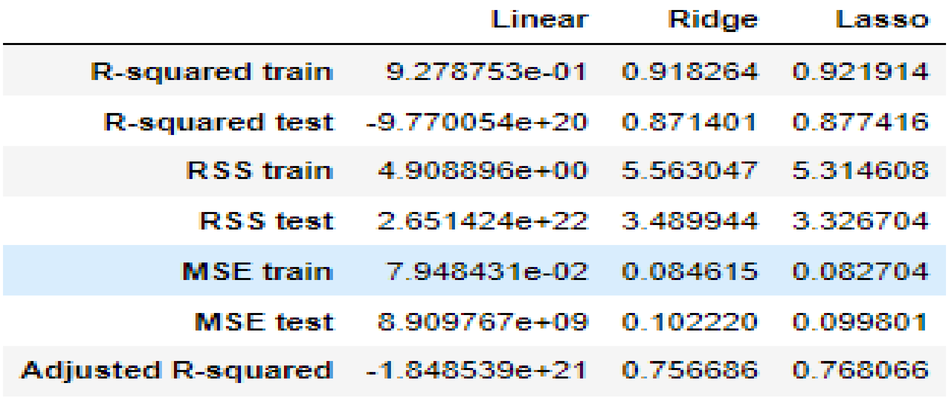
Answer:

The approach that I followed in this assignment is –

1. Understand, Clean and Prepare dataset
2. Create Linear Regression, Ridge and Lasso models on the entire dataset(with all features after data preparation step). Pick the model with decent R-squared and least MSE. (happened to be Lasso in this case)
3. Use lasso regression for feature reduction.
4. Perform further feature reduction using RFE and validate the model using stats mode API.
5. Create Linear Regression, Ridge and Lasso models on the final features post reduction. Pick the model with decent R-squared and least MSE. (happened to be Ridge in this case)

So basically, I have created the models twice, before and after reducing the features.

When the models are built on the complete prepared data (157 features), got the results as below.

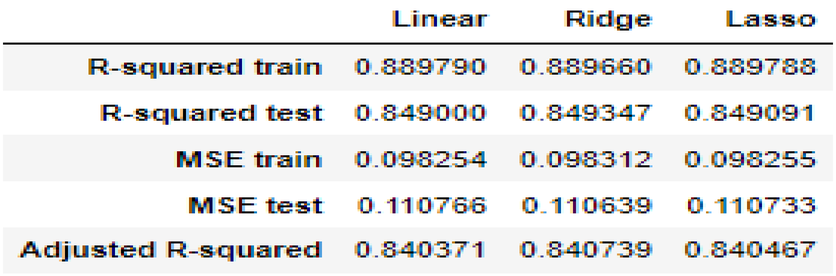


	Linear	Ridge	Lasso
R-squared train	9.278753e-01	0.918264	0.921914
R-squared test	-9.770054e+20	0.871401	0.877416
RSS train	4.908896e+00	5.563047	5.314608
RSS test	2.651424e+22	3.489944	3.326704
MSE train	7.948431e-02	0.084615	0.082704
MSE test	8.909767e+09	0.102220	0.099801
Adjusted R-squared	-1.848539e+21	0.756686	0.768066

- Clealry, the linear regression model is clearly overfitted as R-squared value on test dataset is very poor. Also, MSE is very high.
- Post performing regularisation using Ridge and Lasso regression techniques, the complexity of the model got reduced we see good R-squared value on test dataset.

- **Performing a comparison between Ridge and Lasso models, the R-squared on both train and test sets are higher for Lasso. Also, the MSE is low for Lasso model as compared with Ridge.**
- **We can observe that the adjusted R2 value is quite low as compared to R2. Hence, we need to perform feature reduction. Lasso model can help here with feature reduction, as they reduce coefficients of insignificant features to zero.**
- **For these reasons, Lasso is the better approach to apply.**

Post performing the feature reduction, when models are built again (with 18 features), attained the following results.



	Linear	Ridge	Lasso
R-squared train	0.889790	0.889660	0.889788
R-squared test	0.849000	0.849347	0.849091
MSE train	0.098254	0.098312	0.098255
MSE test	0.110766	0.110639	0.110733
Adjusted R-squared	0.840371	0.840739	0.840467

- In this case, R-squared for train and test sets are higher for Ridge model.
- Also, the adjusted R-squared is better for Ridge.
- Hence, I went ahead in picking the Ridge model(alpha=9.0) to determine the final coefficients of the model.

So, to summarize, we want to build a model which is simple, generalizable, fits well on unseen data and identify the underlying pattern present in the data. The R-squared value should be high on both train and test sets. MSE should be as low as possible and Adjusted-squared should be on-par with R-squared.

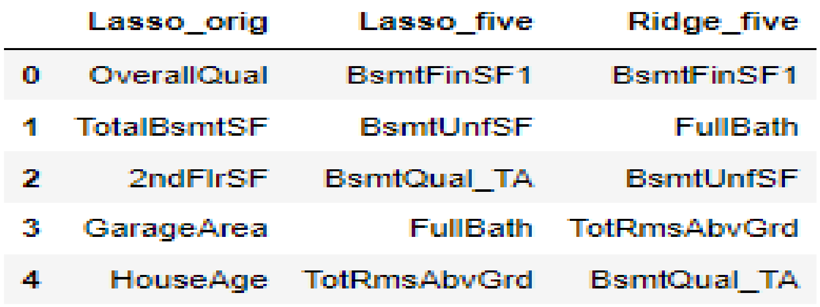
Question 3

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: (*Please refer to the jupyter notebook for the code execution of this question)*

The five most important predictor variables in lasso model are: - 'OverallQual', 'TotalBsmtSF', '2ndFlrSF', 'GarageArea', 'HouseAge'

When we drop these features and rebuild the model, below are the new top 5 predictor variables for Lasso and Regression models.



	Lasso_orig	Lasso_five	Ridge_five
0	OverallQual	BsmtFinSF1	BsmtFinSF1
1	TotalBsmtSF	BsmtUnfSF	FullBath
2	2ndFlrSF	BsmtQual_TA	BsmtUnfSF
3	GarageArea	FullBath	TotRmsAbvGrd
4	HouseAge	TotRmsAbvGrd	BsmtQual_TA

Post removing top 5 features, below are new top 5 features for different models:

Lasso Regression : 'BsmtFinSF1', 'BsmtUnfSF', 'BsmtQual_TA', 'FullBath','TotRmsAbvGrd'

Ridge Regression: 'BsmtFinSF1', 'FullBath', 'BsmtUnfSF', 'TotRmsAbvGrd','BsmtQual_TA'

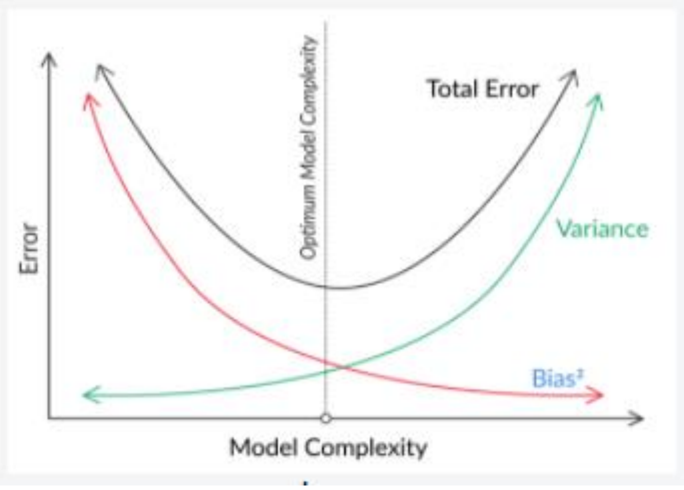
Question 4

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 should be as simple as necessary but not too naive. When in doubt, choose a simpler model.

Advantages of simplicity are generalisability, robustness, requirement of a few assumptions and less data required for learning.



- From the above **Bias-Variance trade off** figure, we can see that the variance in the mode increases as the model complexity increases i.e., the model is not flexible with respect to the changes in the training data.
- It may perform very well on seen data(training datasets), but on the unseen it doesn't perform well. Also, called as **Overfitting,** where the model memorises the data rather than intelligently learning the underlying trends in it.
- Hence for a complex model, the variance is very high, which eventually makes the total error high.

- ➢ Simple models are **generic** as compared to a complex model. This is important because generic/simple models perform better on unseen datasets.
- ➢ A simpler model requires fewer training data points.
- ➢ A simple model is more robust and does not change significantly if the training data points undergo small changes.
- ➢ A simple model may make more errors in the training phase but is bound to outperform complex models when it views new data. This happens because of overfitting.

However, model cannot be too simple, because though the variance is low, the bias is very high i.e., the model cannot accurately measure/describe the task at hand.
So, the total error will be high because of high bias in the system.

It is extremely crucial to strike a balance between the variance and bias in the model, so that we can keep the error in the system minimal. As seen in the model, for a right balance

of bias and variance, the total error is minimum, and it is the desired ‘Optimal Model Complexity’.

To evaluate if our model, fits the optimal model complexity we can verify the following.

- **R-squared values** of test datasets should be on par with the training datasets. This will confirm that we have a generalisable model which is performing well on the unseen data as well. If the difference in R-squared is very high, then we can confirm that our model is **overfit**.
- **Mean Square Error** of the system should be low on both the train and test datasets. This confirms that we did strike a balance between variance and bias in the system and hence the low error.
- It is important to keep model simple, having too many features is not recommended if a quite similar performance can be achieved with fewer features. Hence, we need to verify the **Adjusted R-squared** scores and confirm that it is close to R-squared score of the model.

What we need is lowest total error, i.e., low bias and low variance, such that the model identifies all the patterns that it should and is also able to perform well with unseen data.

If we have a overfit model, **Regularization** helps with managing model complexity by essentially shrinking the model coefficient estimates towards 0. This discourages the model from becoming too complex, thus avoiding the risk of overfitting. For this, we are willing to make a compromise by allowing a little bias for a significant reduction in variance.

Regularization can also help in detecting/eliminating the multicollinearity between in the predictor variables, thus avoiding the variability of model coefficients.