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 to 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 are as **Ridge Regression (alpha = 3.0)** and **Lasso Regression (alpha = 0.001)**

Ridge:

Train set
```
Evaluation metrics
MSE: 0.019937115332844563
RMSE: 0.14119885032408927
R2 Square: 0.8731664879049972
```

Test Set
```
Evaluation metrics
MSE: 0.022906198272994414
RMSE: 0.15134793778903766
R2 Square: 0.8607865439970377
```

Lasso:

Train set
```
Evaluation metrics
MSE: 0.016275915363092263
RMSE: 0.1275770957621009
R2 Square: 0.8964578639588242
```

Test Set
```
Evaluation metrics
MSE: 0.020310831397572553
RMSE: 0.14251607417260886
R2 Square: 0.8765600035653616
```

The value of alpha are doubled as **Ridge Regression (alpha = 6.0)** and **Lasso Regression (alpha = 0.002)**

Ridge:

Train set
```
Evaluation metrics
MSE: 0.020393084764102547
RMSE: 0.1428043583512161
R2 Square: 0.8702657571117547
```

Test Set
Evaluation metrics
MSE: 0.02280077281245668
RMSE: 0.15099924772149256
R2 Square: 0.8614272720016262

Lasso:

Train set
Evaluation metrics
MSE: 0.01778484263248654
RMSE: 0.13335982390692686
R2 Square: 0.88685855423532

Test Set
Evaluation metrics
MSE: 0.02077457605694001
RMSE: 0.1441338824043119
R2 Square: 0.873741574423867

After Doubling, Important features are

Ridge:

Features	Coefficient
MSZoning_RL	0.149472
Neighborhood_Crawfor	0.133602
Neighborhood_ClearCr	0.130618
OverallQual	0.113335
LotShape_IR3	-0.101912
Exterior1st_BrkFace	0.101097
Neighborhood_NridgHt	0.091034
Neighborhood_Somerst	0.089935
Neighborhood_NoRidge	0.088474
BsmtCond_Gd	0.086897

Lasso:

Features	Coefficient
GrLivArea	0.095426
OverallQual	0.094669
Neighborhood_Crawfor	0.061175
Neighborhood_Somerst	0.050207
Neighborhood_Edwards	-0.047694
GarageCars	0.044995
HouseAge	-0.044615
MSZoning_RL	0.044151
OverallCond	0.043771
MSZoning_RM	-0.029889

Note: All python coding and models for above explanation and there in Jupyter notebook

Inferences: Since the alpha is increasing , the model will move from overfitting patterns to underfitting patterns. Also the model parameter coefficients will become lower from original optimal 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?

Answer: The Optimal value of alpha are as **Ridge Regression (alpha = 3.0)** and **Lasso Regression (alpha = 0.001)**

Ridge:

Train set
```
Evaluation metrics
MSE: 0.019937115332844563
RMSE: 0.14119885032408927
R2 Square: 0.8731664879049972
```

Test Set
```
Evaluation metrics
MSE: 0.022906198272994414
RMSE: 0.15134793778903766
R2 Square: 0.8607865439970377
```

Lasso:

Train set
```
Evaluation metrics
MSE: 0.016275915363092263
RMSE: 0.1275770957621009
R2 Square: 0.8964578639588242
```

Test Set
```
Evaluation metrics
MSE: 0.020310831397572553
RMSE: 0.14251607417260886
R2 Square: 0.8765600035653616
```

Inferences: Both models are performing very good on the data and are giving good scores.
We can see that Lasso model a bit more good r2 score which shows good performance.
But if you look at coefficients of features selected from respective models , we would want to **prefer Lasso bcoz it has lower coefficients and it indicated lesser overfitting in the model**

Top5-Ridge:

MSZoning_RL	0.202846
Neighborhood_ClearCr	0.150609
Neighborhood_Crawfor	0.147373
MSZoning_RH	0.143102
MSZoning_FV	0.133173

Top5-Lasso

GrLivArea	0.104875
Neighborhood_Crawfor	0.103423
OverallQual	0.088281
Neighborhood_Somerst	0.086798
MSZoning_RL	0.075296

Question 3

After building the model, you realized 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:

Important top 5 Predictors for Lasso Regression (alpha=0.001) **earlier:**

Features	Coefficient
GrLivArea	0.104875
Neighborhood_Crawfor	0.103423
OverallQual	0.088281
Neighborhood_Somerst	0.086798
MSZoning_RL	0.075296

Important top 5 Predictors for Lasso Regression (alpha=0.001) **after dropping the earlier 5 important predictors :**

Features	Coefficient
Neighborhood_MeadowV	-0.110897
Neighborhood_Edwards	-0.095155
1stFlrSF	0.093723
2ndFlrSF	0.092373
Neighborhood_IDOTRR	-0.090289

Note:

All python coding and models for above explanation and there in Jupyter notebook

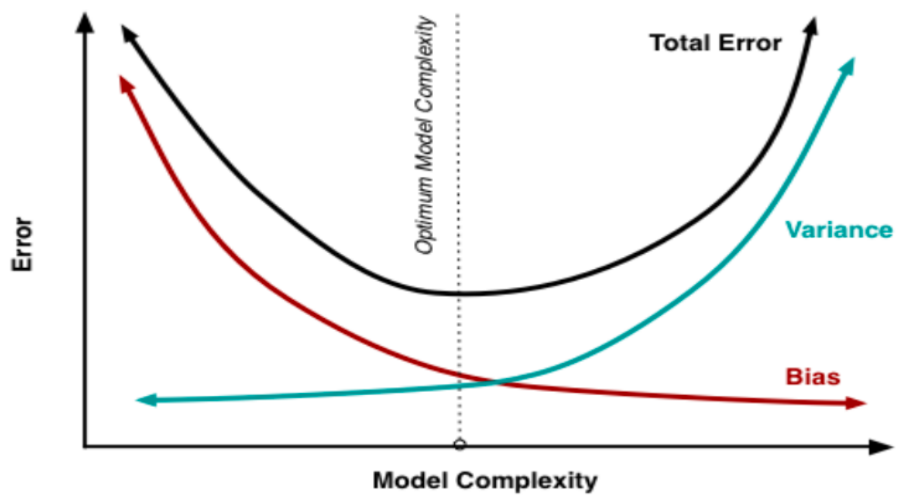
Question 4

How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

Answer:

We can use Occam's Razor to bring the model more robust and generalizable. Occam's Razor is perhaps the most important thumb rule in machine learning, and basic assumptions of it goes as follows :

- Simpler models are usually more 'generic' and are more widely applicable
- Simpler models require fewer training samples for effective training than the more complex ones and are consequently easier to train.
- A simple model is more robust and does not change significantly if the training data points undergo small changes.
- simple models have low variance, high bias and complex models have low bias, high variance
- Simpler models make more errors in the training set — that's the price one pays for greater predictability. Complex models lead to overfitting — they work very well for the training samples, fail miserably when applied to other test samples.
- Bias is the difference between this estimator's expected value and the true value of the parameter being estimated. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- Variance is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).
- Thus, accuracy of the model can be maintained by keeping the balance between Bias and Variance as it minimizes the total error (ideology of **bias-variance trade-off**)



Regularization

Regularization is the process used in machine learning to deliberately simplify models. This method is done during ‘Learning algorithm’ phase.

Some typical regularization steps for various classes examples of ML algorithms include:

1. Adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.
2. For decision trees this could mean ’pruning’ the tree to control its depth and/or size.
3. For neural networks a common strategy is to include a *dropout*—dropping a few neurons and/or weights at random.

Overfitting: Overfitting is a phenomenon where a model becomes way too complex ,which indicates that model performs too well on model training data but miserably fails on test data(Memorising the data points)

To have a robust model , reduce overfitting as much possible.

Note: All the images above are mostly screenshots taken from the Jupyter notebook