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 for Ridge model is : 2.0

The optimal value of alpha for Lasso model is : 0.0001

I have tried doubling the alphas and fitting the model with the new alpha. The change observed is below. The implementation is in the jupyter notebook shared.

Metric	Ridge Regression	Ridge Regression with double alpha	Lasso Regression	Lasso Regression with double alpha
R2 Score (Train)	0.941217	0.936012	0.940837	0.934582
R2 Score (Test)	0.882637	0.89029	0.887635	0.891271
RSS (Train)	1.742219	1.896509	1.753497	1.938891
RSS (Test)	1.524892	1.425461	1.459955	1.412714
MSE (Train)	0.044931	0.046878	0.045076	0.047399
MSE (Test)	0.064198	0.062069	0.062816	0.061791

As represented in the table, there is a slight reduction in the R2 Score, RSS and MSE values . This is expected as we opted for a higher alpha than the optimal one.

Beta values of top 5 features				Beta values of top 5 features			
Ridge Regression		Ridge Regression with doubled alpha		Lasso Regression		Lasso Regression with doubled alpha	
Feature	Coefficient	Feature	Coefficient	Feature	Coefficient	Feature	Coefficient
GrLivArea	0.167062	GrLivArea	0.142778	GrLivArea	0.371216	GrLivArea	0.364408
1stFlrSF	0.125555	1stFlrSF	0.110502	HouseAge	-0.191091	HouseAge	-0.181625
OverallQual_9	0.119426	OverallQual_9	0.0965	OverallQual_9	0.190527	OverallQual_9	0.175499
HouseAge	-0.112034	HouseAge	-0.083145	TotalBsmtSF	0.162481	TotalBsmtSF	0.165912
TotalBsmtSF	0.088676	TotalBsmtSF	0.07936	OverallQual_10	0.110382	OverallQual_10	0.102027

We see that the top 5 features are still the same, but the coefficients are reduced when the alpha value is doubled. This is because the higher the value of alpha, the model would move the coefficients closer to zero.

Question 2

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one would you choose to apply for and why?

Answer:

After building both Ridge and lasso models with the optimal lambda values, we observe the below findings from the models.

Metric	Ridge Regression	Lasso Regression
R2 Score (Train)	0.941217	0.940837
R2 Score (Test)	0.882637	0.887635
RSS (Train)	1.742219	1.753497
RSS (Test)	1.524892	1.459955
MSE (Train)	0.044931	0.045076
MSE (Test)	0.064198	0.062816

As we can see , both the models performed almost similarly.

As per Occam's Razor, given two models that show similar performance in the finite training or test data, we should pick the one that is simpler, because simples models are more generic and widely applicable, and they are more robust.

Considering this, I will choose Lasso regression . This is because Ridge shrinks the coefficients towards 0, but it will not set them to zero. On the other hand, Lasso will also set the coefficients to zero, helping in feature elimination and thus helping to build a more robust model with less features. I will choose lasso if both models perform similarly.

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 :

(The implementation is in the Jupyter notebook enclosed)

After implementing the model , the five most important predictor variables in the lasso model are : (GrLivArea','HouseAge','OverallQual_9','TotalBsmtSF','OverallQual_10')

This implies that Ground floor living area, age of the house, Over all quality of the house and basement area are the most influential predictors in the given data.

As suggested, I have removed these param from the data and created another model with the new training data.

I have noticed that the new model also has optimal alpha value (lambda) of 0.0001

Metric	Lasso Regression	Lasso regression after deleting top 5 predictors
R2 Score (Train)	0.940837	0.935415
R2 Score (Test)	0.887635	0.860639
RSS (Train)	1.753497	1.914201
RSS (Test)	1.459955	1.81071
MSE (Train)	0.045076	0.047096
MSE (Test)	0.062816	0.06995

We see a slight reduction in the model performance after deleting the features, but it's very minimal.

The top five new features after deleting the predictors are :

Feature	Coefficient
1stFlrSF	0.33929
2ndFlrSF	0.212103
OverallQual_3	-0.158811
BsmtFinSF1	0.137248
BsmtFinSF2	0.111035

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:

Ensuring that a model is robust and generalizable is very crucial in real-world applications. However, there is often a tradeoff between these factors and model accuracy, Finding the right balance between there and accuracy is vital for any model to perform well with unseen test data.

Robustness:

Robustness refers to a model's ability to perform well even when exposed to noisy, adversarial, or unexpected inputs. To improve robustness of a model, there are techniques like regularization and data augmentation, which helps generalize the model making it simpler and more robust. A more robust model is less likely to be overly influenced by outliers or small variations in the input data.

Implication for Accuracy: Enhancing robustness can sometimes lead to a decrease in accuracy on the training data. For instance, regularization techniques might prevent the model from fitting the training data perfectly, which could lower the training accuracy. However, this can be beneficial because it reduces the risk of overfitting and improves the model's ability to generalize to new data.

Generalizability:

Generalizability is the model’s ability to perform well on unseen data . Techniques like cross-validation and diverse training help to improve the generalizability of a model. A more generalizable model can handle different variations of the problem and adapt to new scenarios.

Implication for Accuracy: Improving generalizability often comes at the expense of achieving extremely high accuracy on the training data. Models that are overly complex and overfit the training data might have training accuracy but can fail to generalize to new data.

To summarize, the relationship between model accuracy, robustness, and generalizability involves trade-offs. While it might seem counterintuitive to intentionally lower training accuracy to improve generalization and robustness, this approach ultimately results in a model that is better suited for real-world challenges. The key is to strike the right balance by choosing appropriate model complexity, applying regularization, using diverse training data, and employing techniques like cross-validation.