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

Optimum value of alpha for -

Ridge : 0.34

Lasso : 0.0003

Doubling the value of alpha did not have much effect, which may indicate the alpha term is quite low (as doubling it is not having much effect on coefficient size). The R-squared values in both cases are roughly the same, and there are very small changes in the size of the coefficients.

In both ridge and lasso, the top 7 variables are the same pre/post doubling of alpha value - BsmtUnfSF, BsmtFinSF1, 2ndFlrSF, BsmtFinSF2, HouseAge, OverallCond, GarageArea

This indicates that size of house (square foot related variables), age of house and the condition the house is in plays a large role in predicting the price.

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

In this case, Lasso and Ridge have similar outputs, but since Lasso has slightly better R-squared on the test set, I will choose to apply that model.

In general, Lasso will be a better choice as it can set coefficients to zero, leading to a more interpretable model, especially when we think that several features in our dataset may not be influential.

As Ridge regression will never set the coefficient sizes to zero, the coefficients are less interpretable than in Lasso. It is a better choice when we are sure that all the features have some influence on the target variable.

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

In case of both Ridge, and Lasso regression, the most important variables have drastically changed (model performance has also dropped significantly).

Ridge : GarageArea, Neighborhood_NoRidge, RemodelAge, Exterior1st_BrkComm, LotArea

Lasso : GarageArea, Neighborhood_NoRidge, RemodelAge, LotArea, Neighborhood_StoneBr

Many of these were not even in the top 10 in the previous model, and seem to have gained some predictive power after removing the top 5 variables.

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

There are several steps we can take to ensure our model is robust and generalisable -

1. Train/Validation/Test - ensuring that we maintain separate datasets for training the model, validation (tuning of hyperparameters), and then a separate test set for evaluating the performance will ensure we are not overfitting to a specific dataset. In case we want to use as many samples as possible for training, cross-validation can be used instead of a validation dataset. Ensuring that our model performs similarly on the training and test data is a sign that we have a robust model.
2. Regularization - placing penalties on the size and number of parameters in the model can make the model more robust by discouraging overly complex models that have a tendency to overfit.
3. Data cleaning & Feature Selection - treating outliers, removing highly correlated features or features that are not influential can also make our model more robust and generalisable

The general guiding principle is to prefer sparsity and simplicity. In doing so, we are trading off accuracy on the training data in order to achieve a more generalisable model (Bias-Variance tradeoff). Machine learning models can be very powerful, and without the above listed techniques, the model may simply memorize the training data, or learn extremely complicated patterns that do not perform well on unseen data. So even if the model has very high accuracy on the training dataset, we may see very different results when we use it on new data. As most models are trained for future usage on fresh data, it makes sense that we would trade off some training accuracy for more generalisability.

Hence, increased robustness and generalisability can come at the cost of decreased accuracy on the training data, but we accept that as the actual performance of the model in a live environment will be much better as a result.