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:

Optimal value of alpha for Ridge regression: 2

Optimal value of alpha for Lasso regression: 0.0001

Most important variables are (in decreasing order of their coefficient magnitude): Fireplaces_3, Condition2_PosN, GrLivArea, TotalBsmtSF, OverallQual_9, LotArea, Condition2_PosA , houseAge, BsmtFinSF1

When a model performs well on the data that is used to train it, but does not perform well with unseen data, we know we have a problem: **overfitting**.

Ridge regression tries to solve this problem by introducing shrinkage penalty in the model which is function of alpha. Equation for ridge equation is given by:

$$\text{Cost function} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{p} \beta_j^2$$

When alpha is 0, i.e., when there is no regularization, we have a model that is clearly overfitting. However, as the alpha value increases further, the model starts underfitting. We basically want models that do not overfit the data, but they should be able to identify underlying patterns in it. Hence, an appropriate choice of alpha becomes crucial. This can be achieved through "hyperparameter tuning".

Ridge regression does have one obvious disadvantage. It would include all the predictors in the final model. This may not affect the accuracy of the predictions but can make model interpretation challenging when the number of predictors is very large. Lasso regression helps in addressing this problem. Equation for lasso regression is given by:

$$\text{Cost function} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{p} |\beta_j|$$

Lasso regression pushes the model coefficients towards 0 to handle high variance, just like Ridge regression. But, in addition to this, Lasso also pushes some coefficients to be exactly 0 and thus performs variable selection. This variable selection results in models that are easier to interpret.

In Lasso regression, as alpha increases, the variance decreases and the bias increases.

If we double the value of alpha for ridge and lasso regression, r2 value for train dataset decreases but for test dataset it increases. So, it generalizes the model by decreasing the model complexity and compensating bias.

Once we double the alpha, most important variables are: Fireplaces_3, GrLivArea, Condition2_PosN, OverallQual_9, TotalBsmtSF

After doubling the alpha value, number of relevant features have decreased and also coefficients values are decreased in magnitude.

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:

Optimal value of alpha for Ridge regression: 2

Optimal value of alpha for Lasso regression: 0.0001

We will choose lambda = 0.0001 corresponding to lasso regression as it generalizes the model and reduce the coefficients to 0 which make model interpretable. By making coefficients to zero, it performs variable selection.

Lasso regression slightly loose accuracy with the training data but perform better with test data than the ridge regression

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:

When we build the model with all the variables, most important 5 predictors were (in decreasing order of their coefficient magnitude): LotArea, TotalBsmtSF, GrLivArea, houseAge, Condition2_PosN.

Most important predictor variables after removing five most important predictor variables in the lasso model are (in decreasing order of their coefficient magnitude): PoolQC_Gd, 1stFlrSF, Condition2_PosA, OverallQual_10, 2ndFlrSF, LotArea, BsmtFinSF1, houseAge

Question-4:

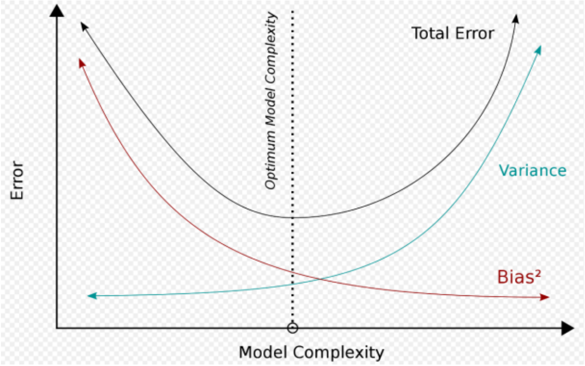
How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

Answer:

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.

We use regularization because we want our models to work well with unseen data, without missing out on identifying underlying patterns in the data. For this, we are willing to make a compromise by allowing a little bias for a significant reduction in variance.

We also understood that the more extreme the values of the model coefficients are, the higher are the chances of model overfitting. Regularization prevents this by shrinking the coefficients towards 0.



The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The variance is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

The bias–variance decomposition is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the irreducible error, resulting from noise in the problem itself.

The bias–variance tradeoff is a central problem in supervised learning. Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously. High variance learning methods may be able to represent their training set well but are at risk of overfitting to noisy or unrepresentative training data. In contrast, algorithms with high bias typically produce simpler models that may fail to capture important regularities (i.e., underfit) in the data.