1. Optimal alpha for Lasso: 0.00033516026509388476

Optimal alpha for ridge: 0.27185882427329455

If we double the alpha, for both lasso and ridge regression, the model fit and performance metrics will change. And the final coefficient of features will also change. The most important variable would be 'OverallQual'

2. I would implement ridge regression because it shows better performance.

Performance metrics for Lasso :

MSE for training

0.42552599736083674

r2 for training

0.8189276255700652

MSE for prediction

0.4599122931634163

r2 for prediction

0.7884806825971677

Performance metrics for Ridge:

MSE for training

0.4394754781800427

r2 for training

0.8068613040784228

MSE for prediction

0.4505229790228729

r2 for prediction

0.797029045372356

3. Five most important variables:

'TotRmsAbvGrd', 'FullBath', 'MasVnrArea', 'KitchenAbvGr', 'BedroomAbvGr'

4.

A model is considered to be robust if its output dependent variable (label) is consistently accurate even if one or more of the input independent variables (features) or assumptions are drastically changed due to unforeseen circumstances.

In a more practical manner, here are different dimensions that need to be validated in order to assume robustness.

(1) Model performance. Using metrics such as adjusted R squared, RMSE, MSE and MAE to make sure that the model is good enough to meet the project's benefits

(2) Model stability: The goal is to get the same performance every time. Of course, a performance that varies too much can be problematic. Using cross-validation to increase model stability.

(3) Model sensitivity: Sensitivity analysis determines how label is affected based on changes in features.

There are 2 different dimensions that you might want to validate:

Model tolerance to noise

Model tolerance to extreme scenarios (targeted noise)

Sensitivity analysis will allow you to explore the generalization of your model's decision boundaries, to really see the impact of a lack of generalization.

(4) Model tolerance to noise:

What happens if your data is a little bit messy?

The risk here is that the model is super narrow, and performance drops suddenly as soon as there is a little bit of noise. Noise could also reflect unseen scenarios. Let's say you develop a credit assessment in the subprime industry (small loans); what would happen if a millionaire applies for a loan?

You can easily test tolerance to noise by adding random noise to the features of your test dataset and see the impact.

(5) Model predictability