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:

Optimal value of alpha for Lasso: **0.018**

Optimal value of alpha for Ridge: **0.76**

Upon doubling the alpha:

Change in the predictor variables for Lasso:

Before -> ***GrLivArea (0.27) > OverallQual (0.21) > GarageCars (0.1) > Fireplaces (0.09) > OverallCond (0.09)***

After -> ***GrLivArea (0.25) > OverallQual (0.22) > Fireplaces (0.08) > OverallCond (0.07) > GarageArea (0.06)***

As you can see, with increase in the alpha value some of the predictors got skipped and the coefficient reduced for all

Change in the predictor variables for Ridge:

Before -> ***MSZoning_FV (0.67) > MSZoning_RH (0.66) > MSZoning_RL (0.64) > MSZoning_RM (0.62) > RoofMatl_WdShngl (0.46)***

After -> ***MSZoning_FV (0.47) > MSZoning_RH (0.46) > MSZoning_RL (0.45) > MSZoning_RM (0.43) > RoofMatl_WdShngl (0.33)***

As you can see, with increase in the alpha value the coefficient has reduced for all

---

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: I would prefer to use Lasso regression for this case as we have ~250 features, and since Lasso regression would eliminate less significant features without affecting prediction accuracy too much, it would help create a lightweight model

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: After excluding the 5 most important predictors, we are getting a new set of predictors (not the next 5 from the last model). They are:

YearBuilt (0.002) > YearRemodAdd (0.001) > GarageArea (0.0003) > 2ndFlrSF (0.0003) > 1stFlrSF (0.0002)

```
house = house_price.drop(columns=['GrLivArea','OverallQual','GarageCars','Fireplaces','Overal
train1, test1 = train_test_split(house, train_size = 0.7, test_size = 0.3, random_state = 100
# 'SalePrice' is our dependent variable. So we add all independent variables into an array
independent_columns= [i for i in house.columns if i != 'SalePrice']

X_train1 = train1[independent_columns]
y_train1 = train1['SalePrice']

X_test1 = test1[independent_columns]
y_test1 = test1['SalePrice']

lasso1 = LassoCV(cv=5, random_state=0, max_iter=10000)

# Fit model
lasso1.fit(X_train1, y_train1)

# alpha value
lasso1.alpha_

coeff = [(a,b) for a,b in zip(lasso1.coef_,lasso1.feature_names_in_) if a and b]
print(sorted(coeff, key = lambda x: x[0], reverse=True))
```

```
[(0.0025469241801980447, 'YearBuilt'), (0.0010014757842449953, 'YearRemodAdd'), (0.0003909959
003239000765999416, '2ndFlrSF'), (0.00027166035801738056, '1stFlrSF'), (0.0001826912002285535
4797906303, 'WoodDeckSF'), (4.392970434087118e-05, 'BsmtFinSF1'), (4.00645479854871e-05, 'Mas
-06, 'LotArea'), (1.2507128372932005e-07, 'MiscVal'), (-0.0004756050286839886, 'PoolArea')]
```

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: For a model to be robust and generalizable, it should perform equally well on both training and test data, i.e., if the model does extremely well on the training data (high accuracy) but predicts poorly on the test data, then the model is not robust. This often happens when a model overfits and learns even the noise in the training data. This is often discussed in terms of Bias-Variance tradeoff.

Bias refers to the error in predicting the value. High bias would mean the model could not learn the data well

Variance refers the sensitivity to changes in the training set. High variance would mean the model has overfit the data (including noise)

For robustness, we need to reduce the variance so it can predict well with unseen data. Often achieving robustness would involve some trade-off with accuracy

Some of the ways to achieve this are:

1. **Outlier analysis**: We need to analyze data for outliers as they can mislead the model.
2. **Regularization**: We can use Lasso/Ridge regression to achieve robustness by intentionally introducing bias that can help reduce variance
3. **Cross-Validation**: We can use cross validation techniques like k-fold or leave-one-out to test the model with different datasets
4. **Data Augmentation**: We can use data augmentation techniques to increase sample size if we have limited data