## Q1. 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?

The optimal value for alpha for ridge and lasso regression are 50 and 0.001 respectively. Initially, the 10 most important features along with respective co-efficient for ridge regression model with alpha value of 50 is given as follows:

| | Feaure | Coef |
|---|---|---|
| 0 | MSSubClass | 13.555995 |
| 16 | OverallCond | 0.076113 |
| 45 | BsmtFullBath | 0.065881 |
| 17 | YearBuilt | 0.054842 |
| 43 | LowQualFinSF | 0.050725 |
| 2 | LotArea | 0.042188 |
| 59 | GarageQual | 0.030329 |
| 42 | 2ndFlrSF | 0.029544 |
| 28 | BsmtQual | 0.023942 |
| 3 | Street | 0.022813 |

The 10 most important features for lasso regression model with alpha value of 0.001 is given as follows:

| | Features | rfe_support | rfe_ranking | Coefficient |
|---|---|---|---|---|
| 10 | GrLivArea | True | 1 | 0.125134 |
| 3 | OverallQual | True | 1 | 0.080225 |
| 4 | OverallCond | True | 1 | 0.057157 |
| 0 | MSZoning | True | 1 | 0.049396 |
| 8 | CentralAir | True | 1 | 0.041041 |
| 7 | Heating | True | 1 | 0.033909 |
| 14 | GarageCars | True | 1 | 0.028924 |
| 2 | LandContour | True | 1 | 0.025527 |
| 1 | LotArea | True | 1 | 0.023584 |
| 9 | 2ndFlrSF | True | 1 | 0.014948 |

The comparison between two models is given as follows:

| | Metric | Ridge regression | Lasso regression |
|---|---|---|---|
| 0 | R2 Score Train | 0.917140 | 0.917109 |
| 1 | R2Score Test | 0.906068 | 0.906981 |
| 2 | RSS Train | 11.847226 | 11.851656 |
| 3 | RSS Test | 5.836310 | 5.779570 |
| 4 | MSE Train | 0.012226 | 0.012231 |
| 5 | MSE Test | 0.014030 | 0.013893 |

Now, the value of alpha for ridge is changed to 100 and for lasso it is changed to 0.002. the 10 most important features along with respective co-efficient for ridge regression model with alpha value of 100 is given as follows:

| | Featuere | Coef |
|---|---|---|
| 0 | MSSubClass | 13.342578 |
| 16 | OverallCond | 0.071063 |
| 45 | BsmtFullBath | 0.060636 |
| 17 | YearBuilt | 0.051808 |
| 43 | LowQualFinSF | 0.046931 |
| 2 | LotArea | 0.035701 |
| 59 | GarageQual | 0.030617 |
| 42 | 2ndFlrSF | 0.026902 |
| 3 | Street | 0.022754 |
| 28 | BsmtQual | 0.022211 |

The 10 most important features for lasso regression model with alpha value of 0.002 is given as follows:

| | Features | rfe_support | rfe_ranking | Coefficient |
|---|---|---|---|---|
| 10 | GrLivArea | True | 1 | 0.125134 |
| 3 | OverallQual | True | 1 | 0.080225 |
| 4 | OverallCond | True | 1 | 0.057157 |
| 0 | MSZoning | True | 1 | 0.049396 |
| 8 | CentralAir | True | 1 | 0.041041 |
| 7 | Heating | True | 1 | 0.033909 |
| 14 | GarageCars | True | 1 | 0.028924 |
| 2 | LandContour | True | 1 | 0.025527 |
| 1 | LotArea | True | 1 | 0.023584 |
| 9 | 2ndFlrSF | True | 1 | 0.014948 |

The comparison between ridge and lasso for doubled value of alpha is given by :

| | Metric | Ridge regression | Lasso regression |
|---|---|---|---|
| 0 | R2 Score Train | 0.914938 | 0.914425 |
| 1 | R2Score Test | 0.903996 | 0.904457 |
| 2 | RSS Train | 12.161977 | 12.235426 |
| 3 | RSS Test | 5.965086 | 5.936450 |
| 4 | MSE Train | 0.012551 | 0.012627 |
| 5 | MSE Test | 0.014339 | 0.014270 |

If you choose to double the value of alpha for both Ridge and Lasso regularization, it will have the following effects on the models:

**1. Ridge Regression:**

Increasing the value of alpha in Ridge regression will increase the strength of regularization.

This means that the coefficients will be more heavily penalized, leading to a more constrained model.

The effect will be that the coefficients will tend towards zero, potentially reducing the model's sensitivity to the input features. The model might become more biased but less sensitive to noise or multicollinearity in the data.

2. **Lasso Regression:**

Like Ridge, increasing the value of alpha in Lasso regression will increase the strength of regularization. However, Lasso has a unique property of feature selection: as alpha increases, some coefficients may be driven all the way to zero. This means that some features may be entirely excluded from the model, effectively performing automatic feature selection.

Here, it can be seen that on increasing the value of alpha , the model accuracy slightly decreases and the co-efficient of the predictor values slightly decreases which means they are heavily penalized.

## 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?

Ans:

Based on the alpha/Lambda values I have got, Ridge regression does not zero any of the co efficient, Lasso zeroed one or two coefficients in the selected features, Lasso is better option and it also helps in the some of the feature elimination. The R2 score of both the models is almost same.

## 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?

The five most important predictor variables after removing the prior 5 most important variables for lasso regression are given as follows:

| | Featuere | Coef |
|---|---|---|
| **0** | MSSubClass | 11.575653 |
| **10** | Neighborhood | 0.205033 |
| **11** | Condition1 | 0.158588 |
| **15** | YearBuilt | 0.080554 |
| **1** | MSZoning | 0.078558 |

## 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?

Ensuring that a model is robust and generalizable is crucial for its effectiveness in real-world applications. Here are some strategies to achieve this:

1. **Cross-Validation**: Use techniques like k-fold cross-validation to evaluate the model's performance on multiple subsets of the data. This helps to assess how well the model will perform on unseen data.

2. **Feature Selection**: Choose relevant features and avoid overfitting by using techniques like L1 regularization (Lasso) or tree-based feature importance.

3. **Regularization**: Apply techniques like Ridge (L2) regression or Lasso (L1) regression to penalize large coefficients and prevent overfitting.

4. **Avoid Data Leakage**: Ensure that information from the test set doesn't inadvertently leak into the training process. For example, be cautious with data preprocessing steps that involve the entire dataset.

5. **Use a Holdout Set**: After training and validating the model using cross-validation, keep a separate holdout set that the model has never seen. This can be used for a final evaluation.

6. **Evaluate on Different Datasets:** Test the model on different datasets or different time periods if applicable. This helps to ensure that the model's performance is consistent across various scenarios.

7. **Ensemble Methods:** Combine multiple models to reduce individual model biases. This can lead to more robust predictions.

8**. Monitor Model's Performance:** Continuously monitor the model's performance over time. If the performance drops, it might indicate that the model is becoming less generalizable.

9. **Hyperparameter Tuning:** Avoid overfitting hyperparameters to the validation set. Use nested cross-validation or techniques like grid search with cross-validation to tune hyperparameters.

10. **Interpretability and Simplicity**: Prefer simpler models when they perform close to or as well as complex models. They are often more robust and easier to interpret.

**Implications for Accuracy:**

**Accuracy vs. Generalization**: A model that is overfitted to the training data might have very high accuracy on the training set but perform poorly on new data. This is because it's essentially memorizing the training data rather than learning the underlying patterns.

**Balancing Bias and Variance:** Achieving high accuracy on the training set doesn't necessarily mean the model will perform well in real-world scenarios. It's important to strike a balance between bias and variance to ensure the model generalizes well.

**Trade-off with Complexity**: Increasing model complexity can sometimes lead to higher accuracy on the training set, but it also increases the risk of overfitting. It's important to find the right level of complexity that generalizes well.

In summary, the goal is to build models that can make accurate predictions on new, unseen data. This involves careful evaluation, validation, and sometimes sacrificing a bit of training set accuracy to achieve a more robust and generalizable model.