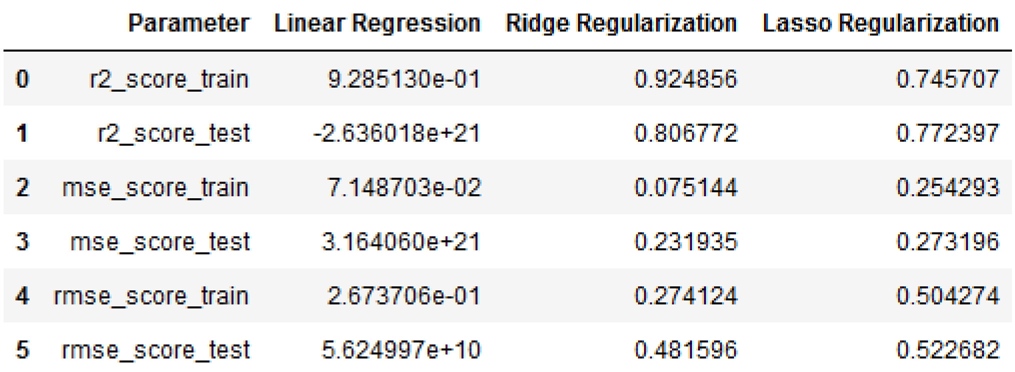
Question 1

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?

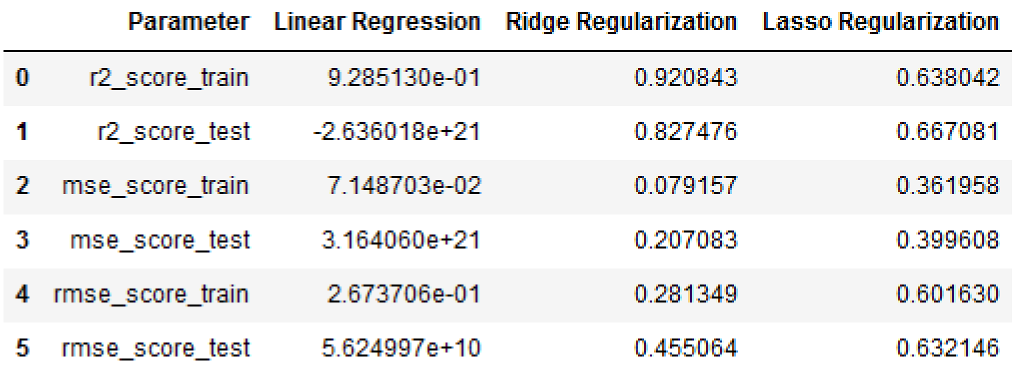
Answer : Optimal value of alpha for Ridge Regularization is **5.05050505050505** and Optimal value of alpha for Lasso regularization is **0.10101010101010101**.

Summary output:



	Parameter	Linear Regression	Ridge Regularization	Lasso Regularization
0	r2_score_train	9.285130e-01	0.924856	0.745707
1	r2_score_test	-2.636018e+21	0.806772	0.772397
2	mse_score_train	7.148703e-02	0.075144	0.254293
3	mse_score_test	3.164060e+21	0.231935	0.273196
4	rmse_score_train	2.673706e-01	0.274124	0.504274
5	rmse_score_test	5.624997e+10	0.481596	0.522682

When alpha value is doubled for Ridge and Lasso:



	Parameter	Linear Regression	Ridge Regularization	Lasso Regularization
0	r2_score_train	9.285130e-01	0.920843	0.638042
1	r2_score_test	-2.636018e+21	0.827476	0.667081
2	mse_score_train	7.148703e-02	0.079157	0.361958
3	mse_score_test	3.164060e+21	0.207083	0.399608
4	rmse_score_train	2.673706e-01	0.281349	0.601630
5	rmse_score_test	5.624997e+10	0.455064	0.632146

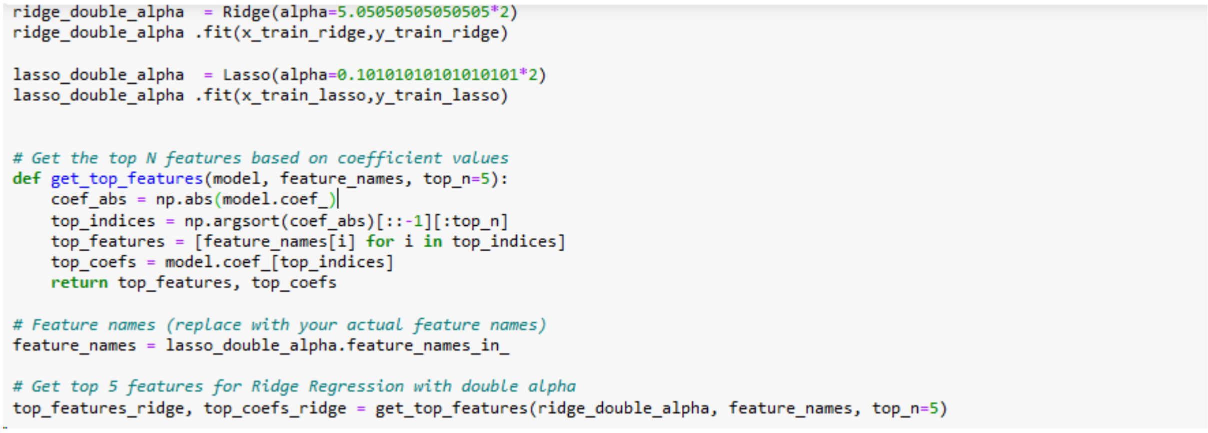
Here is the summary what will happen if we double the alpha values

Ridge Regression:

- **Larger Regularization Penalty:**
  - With double the value of alpha, the regularization penalty term in the Ridge regression objective function will be twice as strong.
- **Smoother Coefficient Estimates:**
  - As a result of increased regularization, Ridge regression tends to produce smoother and more evenly distributed coefficient estimates.
- **Decreased Model Complexity:**
  - Larger alpha values encourage simpler models by penalizing the model for complex coefficient patterns.

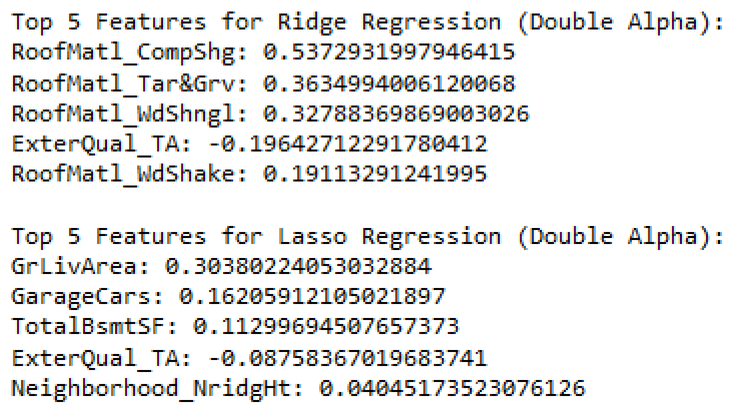
Lasso Regression:

- **Feature Selection:**
  - With a higher alpha, more features may be completely excluded from the model.
- **Variable Importance:**
  - Features with larger coefficients in Lasso are relatively more important in predicting the target variable. Increasing alpha will downweight the importance of some features.



```
ridge_double_alpha  = Ridge(alpha=5.050505050505*2)
ridge_double_alpha .fit(x_train_ridge,y_train_ridge)

lasso_double_alpha  = Lasso(alpha=0.10101010101010101*2)
lasso_double_alpha .fit(x_train_lasso,y_train_lasso)


# Get the top N features based on coefficient values
def get_top_features(model, feature_names, top_n=5):
    coef_abs = np.abs(model.coef_)
    top_indices = np.argsort(coef_abs)[::-1][:top_n]
    top_features = [feature_names[i] for i in top_indices]
    top_coefs = model.coef_[top_indices]
    return top_features, top_coefs

# Feature names (replace with your actual feature names)
feature_names = lasso_double_alpha.feature_names_in_

# Get top 5 features for Ridge Regression with double alpha
top_features_ridge, top_coefs_ridge = get_top_features(ridge_double_alpha, feature_names, top_n=5)
```

Output:

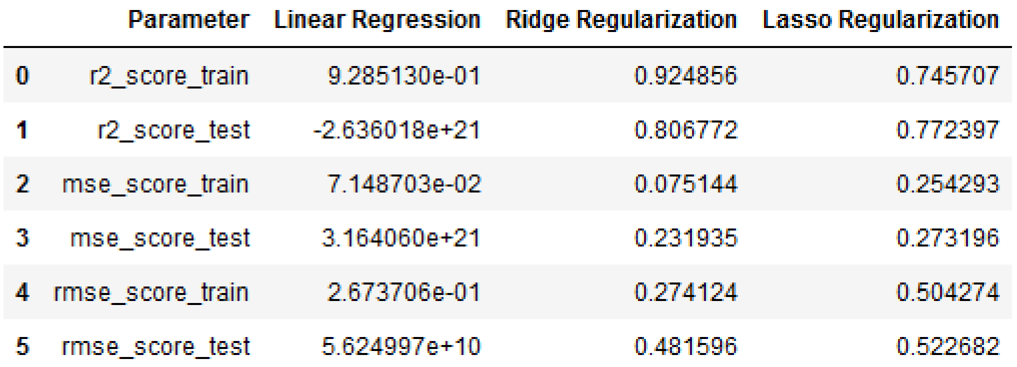


```
Top 5 Features for Ridge Regression (Double Alpha):
RoofMatl_CompShg: 0.5372931997946415
RoofMatl_Tar&Grv: 0.3634994006120068
RoofMatl_WdShngl: 0.32788369869003026
ExterQual_TA: -0.19642712291780412
RoofMatl_WdShake: 0.19113291241995

Top 5 Features for Lasso Regression (Double Alpha):
GrLivArea: 0.30380224053032884
GarageCars: 0.16205912105021897
TotalBsmtSF: 0.11299694507657373
ExterQual_TA: -0.08758367019683741
Neighborhood_NridgHt: 0.04045173523076126
```

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: Looking at the summary output I will choose Lasso over Ridge for this use case as we can see that there is not much variance in the training and testing dataset r2 score value,where as in the Ridge we can see some variance.



	Parameter	Linear Regression	Ridge Regularization	Lasso Regularization
0	r2_score_train	9.285130e-01	0.924856	0.745707
1	r2_score_test	-2.636018e+21	0.806772	0.772397
2	mse_score_train	7.148703e-02	0.075144	0.254293
3	mse_score_test	3.164060e+21	0.231935	0.273196
4	rmse_score_train	2.673706e-01	0.274124	0.504274
5	rmse_score_test	5.624997e+10	0.481596	0.522682

In summary, the choice between Ridge and Lasso regression, as well as the optimal value of lambda, depends on specific situation. If we want to preserve all features and handle multicollinearity, Ridge might be more appropriate. If we are looking for feature selection and a simpler model, Lasso could be the better choice. Ultimately, conducting thorough cross-validation experiments and understanding the characteristics of data are crucial steps in making an good decision.

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: After excluding the top 5 most important model in the Lasso ,we have below variables next on priority along with its coefficent values.

```
Heating_OthW: 0.3286523908943169
BsmtFinType1_Unf: 0.15774083649415407
BsmtUnfSF: 0.1030726325228914
BsmtFinType1_GLQ: 0.09713363599610424
SaleType_New: -0.07825441483453448
```

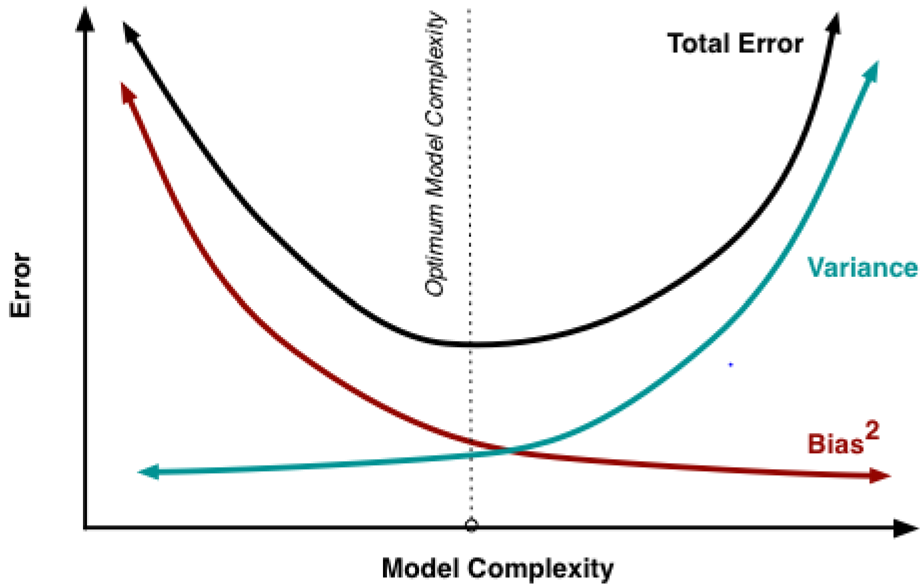
Question 4

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:

To ensure a model is robust and generalizable ,we can consider below points:

1. **Cross-Validation Technique:** We can use the cross validation techniques like K fold which will help us train our model even with less dataset.
2. **Train-Test Split:** Separate data into training and testing sets for model evaluation.
3. **Regularization:** Apply techniques like Lasso or Ridge regularization to prevent overfitting . We can make use trade of between bias and variance for finding better alpha value for regularization.



4. **Metrics function of sklearn:** multiple metrics (precision, recall, etc.) rather than relying solely on accuracy. As sometimes our accuracy would be good in training set but on testing data set our model might not perform well.
5. **Error-Term Analysis**: Analysing the error term found by the model.

Implications for Accuracy:

1. **Training Accuracy vs. Test Accuracy:**
   - If a model has high training accuracy but low test accuracy, it may be overfitting the training data and failing to generalize well.
2. **Consistent Performance:**
   - A robust and generalizable model will exhibit consistent performance across different subsets of the data. The performance metrics on the training, validation, and test sets should be similar.
3. **Avoiding Overfitting:**
   - Regularization techniques and careful feature engineering help prevent overfitting, ensuring that the model doesn't learn noise in the training data.
4. **Balancing Bias and Variance:**
   - Achieving a balance between bias and variance is essential. High bias (underfitting) or high variance (overfitting) can both lead to poor generalization.

In summary, robust and generalizable models strike a balance between fitting the training data well and being able to make accurate predictions on new, unseen data. The practices mentioned above contribute to model stability and performance across various scenarios. Regular evaluation, validation, and fine-tuning based on real-world performance are key to building models that generalize effectively.