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 :

- For Lasso, the optimal alpha value is 0.0001, while for Ridge, it is 1.2. With these alphas, the model's R2 was roughly 0.91
- Optimal values for apha ridge and lasso regression :

```
Evaluation of Ridge Regression:
Best parameters: {'alpha': 1.2}
R2-square score (test): 0.9316989665539559
RSS (test): 0.0011314825247454611
RMSE (test): 0.033637516625718095

Evaluation of Lasso Regression:
Best parameters: {'alpha': 0.0001}
R2-square score (test): 0.9274597981933405
RSS (test): 0.0012017090597989 81
RMSE (test): 0.034665675527803884
```

Coefficients of ridge and lasso regression using optimal alpha value:

	Variable	Linear Regression	Ridge Regression	Lasso Regression
0	LotArea	5.205494e-02	0.045356	0.038063
1	OverallQual	1.030157e-01	0.089815	0.103375
2	OverallCond	6.534025e-02	0.060284	0.060043
3	YearBuilt	6.398386e-02	0.060813	0.069819
4	BsmtFinSF1	-2.598479e+10	0.092629	0.064988
5	BsmtFinSF2	-1.075970e+10	0.020875	0.000000
6	BsmtUnfSF	-2.556912e+10	0.027818	0.000000
7	TotalBsmtSF	3.807460e+10	0.087797	0.111238
8	1stFlrSF	-6.005063e+11	0.094914	0.000000
9	2ndFlrSF	-3.342832e+11	0.068670	0.000448

- With higher alpha, the penalty for large coefficients increases. This shrinks the coefficient magnitudes further
- Doubling alpha in ridge regression shrinks coefficients further but doesn't change feature selection
- Doubling alpha in lasso regression can select a smaller, potentially more relevant set of features. The remaining non-zero coefficients indicate the most important predictors after the change

-

```
coef_df[['Variable', 'Ridge Regression']].sort_values(by=["Ridge Regression"], ascending=False).head(5)
```

	Variable	Ridge Regression
11	GrLivArea	0.132682
8	1stFlrSF	0.094914
4	BsmtFinSF1	0.092629
1	OverallQual	0.089815
7	TotalBsmtSF	0.087797

```
[34] coef_df[['Variable', 'Lasso Regression']].sort_values(by=["Lasso Regression"], ascending=False).head(5)
```

	Variable	Lasso Regression
11	GrLivArea	0.270288
7	TotalBsmtSF	0.111238
1	OverallQual	0.103375
3	YearBuilt	0.069819
4	BsmtFinSF1	0.064988

```
[35] coef_df[['Variable', 'Ridge Regression']].sort_values(by=["Ridge Regression"], ascending=False).head(5)
```

```
R2-square score (test) for Ridge with alpha=2.4: 0.9295507905590122
R2-square score (test) for Lasso with alpha=0.0002: 0.9228424794414256
```

	Variable	Ridge Regression
11	GrLivArea	0.132682
8	1stFlrSF	0.094914
4	BsmtFinSF1	0.092629
1	OverallQual	0.089815
7	TotalBsmtSF	0.087797

```
coef_df[['Variable', 'Lasso Regression']].sort_values(by=["Lasso Regression"], ascending=False).head(5)
```

	Variable	Lasso Regression
11	GrLivArea	0.270288
7	TotalBsmtSF	0.111238
1	OverallQual	0.103375
3	YearBuilt	0.069819
4	BsmtFinSF1	0.064988

- Overall, doubling the alpha does not significantly alter the model because the alpha values are small.

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 :

- For Ridge and Lasso, the optimal lambda value is as follows:
    - Ridge Regression : 1.2
    - Lasso Regression 0.0001

```
Evaluation of Ridge Regression:
Best parameters: {'alpha': 1.2}
R2-square score (test): 0.9316989665539559
RSS (test): 0.0011314825247454611
RMSE (test): 0.033637516625718095

Evaluation of Lasso Regression:
Best parameters: {'alpha': 0.0001}
R2-square score (test): 0.9274597981933405
RSS (test): 0.001201709059798981
RMSE (test): 0.034665675527803884
```

- The Mean Squared Error for both models is nearly equal.
- Lasso has an advantage over Ridge in feature reduction (some features' coefficient values become zero), so it should be the final model.

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 :

- In the current lasso model with optimal lambda value (0.0001) , the following five predictor variables are the most significant:
    a. GrLivArea: Above grade (ground) living area square feet
    b. OverallQual: Rates the overall material and finish of the house
    c. TotalBsmtSF: Total square feet of basement area
    d. YearBuilt: Original construction date
    e. BsmtFinSF1: Type 1 finished square feet

```
X_train_rfe_new_df = X_train.drop(['GrLivArea', 'OverallQual', 'TotalBsmtSF', 'YearBuilt',

new_lasso = Lasso(alpha=0.0001)

new_lasso.fit(X_train_rfe_new_df, y_train)
y_pred_test_lasso = new_lasso.predict(X_test[X_train_rfe_new_df.columns])
r2_square_test_lasso = r2_score(y_test, y_pred_test_lasso)

print("R2-square score (test)", r2_square_test_lasso)
coef_df_with_columns_removed = pd.DataFrame({
    'Variable': X_train_rfe_new_df.columns,
    'Lasso Regression': new_lasso.coef_
})

coef_df_with_columns_removed.sort_values(by=["Lasso Regression"], ascending=False).head(5)
```

```
R2-square score (test) 0.9138526640391479
```

	Variable	Lasso Regression
4	1stFlrSF	0.406985
5	2ndFlrSF	0.146041
32	Neighborhood_StoneBr	0.074317
11	GarageCars	0.072759
28	Neighborhood_NoRidge	0.066160

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 :

- A robust and generalizable model should have low variance. This means the model's performance is consistent and doesn't fluctuate significantly with small changes in the training data. Techniques like regularization Lasso or Ridge regression and avoiding overly complex models help achieve this. A high variance model might perform very well on the specific training data it was built on, but its predictions might be unreliable for unseen data.
- A robust and generalizable model should have low bias. Bias refers to the systematic difference between the model's predictions and the true values. Techniques like using high quality, representative data and feature engineering that captures relevant information can help reduce bias. A model with high bias might consistently miss the mark, regardless of the data it's applied to.
- A robust and generalizable model should perform well on unseen data. This means the model can learn patterns from the training data and apply them effectively to predict outcomes for new data points it hasn't encountered before. Techniques like cross-validation help assess a model's generalizability by evaluating its performance on a separate testing set not used for training. A model that overfits the training data might not be able to handle the inherent variations present in real world data.