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 to double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer 1

Optimal Value of Alpha:

- Alpha (Ridge Regression) : 10.0
- Alpha (Lasso Regression) : 0.001

Changes required in the model if we choose to double the alpha value for Ridge and Lasso regression models:

```
# Model Building
ridge_model = Ridge(alpha=20.0)
ridge_model.fit(X_train_rfe, y_train)

# Predicting
y_train_pred = ridge_model.predict(X_train_rfe)
y_test_pred = ridge_model.predict(X_test_rfe)

print("Model Evaluation : Ridge Regression, alpha=20.0")
print('R2 score (train) : ',round(r2_score(y_train,y_train_pred), 4))
print('R2 score (test) : ',round(r2_score(y_test,y_test_pred), 4))
print('RMSE (train) : ', round(np.sqrt(mean_squared_error(y_train, y_train_pred)), 4))
print('RMSE (test) : ', round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 4))
```

```
Model Evaluation : Ridge Regression, alpha=20.0
R2 score (train) :  0.9165
R2 score (test) :  0.8709
RMSE (train) :  0.1132
RMSE (test) :  0.1536
```

```
lasso_model = Lasso(alpha=0.002)
lasso_model.fit(X_train_rfe, y_train)
y_train_pred = lasso_model.predict(X_train_rfe)
y_test_pred = lasso_model.predict(X_test_rfe)

print("Model Evaluation : Lasso Regression, alpha=0.002")
print('R2 score (train) : ',round(r2_score(y_train,y_train_pred), 4))
print('R2 score (test) : ',round(r2_score(y_test,y_test_pred), 4))
print('RMSE (train) : ', round(np.sqrt(mean_squared_error(y_train, y_train_pred)), 4))
print('RMSE (test) : ', round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 4))
```

```
Model Evaluation : Lasso Regression, alpha=0.002
R2 score (train) :  0.9146
R2 score (test) :  0.8759
RMSE (train) :  0.1144
RMSE (test) :  0.1506
```

Important predictor variable after implementing the above changes:

1. Ridge model: **1stFlrSF**
2. Lasso model: **1stFlrSF**

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 2

Lasso Regression produced a model which is performing slightly better on the test set as compared to ridge and the drop in R-squared between train and test is least in Lasso. Thus, in the given scenario, I have opted for Lasso.

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
Optimum alpha for ridge is 10.000000
ridge  Regression with  10.0
====================================
R2 score (train) :  0.9166751151617185
R2 score (test) :  0.8704201226046138
RMSE (train) :  0.11304414693007463
RMSE (test) :  0.15390088041290242
[Parallel(n_jobs=1)]: Done 135 out of 135 | elapsed:    0.7s finished
```

```
# Lasso Regression
params = {'alpha': [0.000001, 0.00001,0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500, 1000, 1(

lasso_final_model, y_test_predicted = build_model(X_train_rfe, y_train, X_test_rfe, param
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
Optimum alpha for lasso is 0.001000
lasso  Regression with  0.001
====================================
R2 score (train) :  0.9157339730212566
R2 score (test) :  0.8745163217343286
RMSE (train) :  0.1136807627429514
RMSE (test) :  0.15144883684391253
[Parallel(n_jobs=1)]: Done  60 out of  60 | elapsed:    0.3s finished
```

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 3

Most important variables under the given scenario would be:

1. GarageArea
2. KitchenQual
3. LotArea
4. Fireplaces
5. BsmtQual

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 4

To make sure model is robust and generalizable is critical for making accurate predictions on the unseen data. Quick indicator to check if the model is robust and generalizable is by comparing the R-squared values on the training set and the test set. If the drop is huge, then it indicates that the model has high variance.

To ensure that model is robust and generalizable, we need to sacrifice some level of accuracy on the training data and strike a right balance between bias & variance.

Few techniques to achieve this are:

1. Cross-validation & Out of sample evaluation – ensures that model's performance is consistent across different samples
2. Regularization – Adding a penalization term to reduce the size of coefficients and prevent overfitting
3. Feature Engineering – create feature which captures important relationships in the data
4. Model Complexity – avoid overly complex models, simpler models are often robust and generalisable
5. Hyperparameter tuning - Optimize the model's hyperparameters using techniques such as grid search or random search. Tuning hyperparameters helps to find the optimal balance between bias and variance, leading to a model that generalizes well to new data.

It's important to strike the right balance between bias and variance when designing a model. While overly complex models may achieve high accuracy on the training data, they are more prone to overfitting and generalize poorly to new data. A simpler model with slightly lower training accuracy may ultimately perform better in practice by avoiding overfitting and capturing the underlying patterns in the data more effectively.