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:

Ridge model:

```
#Ridge Double alpha 8
ridge_double = Ridge(alpha=8,random_state=100)
ridge_double.fit(X_train_rfe2,y_train)
ridge_double_coef = ridge_double.coef_
y_test_pred = ridge_double.predict(X_test_rfe2)
print('The R2 Score  is',r2_score(y_test, y_test_pred))
print('The MSE is', mean_squared_error(y_test, y_test_pred))
ridge_double_coeff =
pd.DataFrame(np.atleast_2d(ridge_double_coef),columns=X_train_rfe2.columns
)
ridge_double_coeff = ridge_double_coeff.T
ridge_double_coeff.rename(columns={0: 'Ridge Doubled Alpha
Co-Efficient'},inplace=True)
ridge_double_coeff.sort_values(by=['Ridge Doubled Alpha Co-Efficient'],
ascending=False,inplace=True)
ridge_double_coeff.head(20)
```

```
The R2 Score  is 0.8306587905041612
The MSE is 0.027857935471359837
```

	Ridge Doubled Alpha Co-Efficient
TotRmsAbvGrd	0.255471
FullBath	0.222097
Fireplaces	0.192161
GarageArea	0.178627
LotFrontage	0.151963

Lasso model:

```
#Lasso Double alpha 0.0008
lasso_double = Lasso(alpha=0.0008,random_state=100)
lasso_double.fit(X_train_rfe2,y_train)
lasso_double_coef = lasso_double.coef_
y_test_pred = lasso_double.predict(X_test_rfe2)
print('The R2 is',r2_score(y_test, y_test_pred))
print('The MSE is', mean_squared_error(y_test, y_test_pred))
lasso_double_coeff =
pd.DataFrame(np.atleast_2d(lasso_double_coef),columns=X_train_rfe2.columns
)
lasso_double_coeff = lasso_double_coeff.T
lasso_double_coeff.rename(columns={0: 'Lasso Doubled Alpha
Co-Efficient'},inplace=True)
lasso_double_coeff.sort_values(by=['Lasso Doubled Alpha Co-Efficient'],
ascending=False,inplace=True)
lasso_double_coeff.head(5)
```

```
The R2 is 0.8321014935951614
The MSE is 0.027620599682079312
```

	Lasso Doubled Alpha Co-Efficient
TotRmsAbvGrd	0.379384
GarageArea	0.272563
FullBath	0.265347
Fireplaces	0.204057
LotArea	0.180123

As we can see there is not much difference even if we double the optimal alpha value but here and there few features importance is being increased.

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:
In the assignment our optimal alpha values are:
- Ridge : 4
- Lasso : 0.0004

In the assignment R2 scores for both models are:
- Ridge : 0.8342
- Lasso : 0.8902

In the assignment MSE for both models are:
- Ridge : 0.027271
- Lasso : 0.026480

From all the above observations it is better to consider the lasso model as it has a less MSE compared to the ridge model and also has a good R2 score. The other advantage we get with the Lasso model is the coefficient values of few features will be ‘0’ making it an advantage over the ridge 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:

```
#remove top 5
X_test_rfe3 = X_test_rfe2.drop(['FullBath','GarageArea','TotRmsAbvGrd','OverallQual_Very Excellent','LotArea'],axis=1)
X_train_rfe3 = X_train_rfe2.drop(['FullBath','GarageArea','TotRmsAbvGrd','OverallQual_Very Excellent','LotArea'],axis=1)
#model
lasso3 = Lasso(alpha=0.0004,random_state=100)
lasso3.fit(X_train_rfe3,y_train)
lasso3_coef = lasso3.coef_
y_test_pred = lasso3.predict(X_test_rfe3)
print('The R2 Score  is',r2_score(y_test, y_test_pred))
print('The MSE  is', mean_squared_error(y_test, y_test_pred))
lasso3_coeff = pd.DataFrame(np.atleast_2d(lasso3_coef),columns=X_train_rfe3.columns)
lasso3_coeff = lasso3_coeff.T
lasso3_coeff.rename(columns={0: 'Lasso Co-Efficient'},inplace=True)
lasso3_coeff.sort_values(by=['Lasso Co-Efficient'], ascending=False,inplace=True)
lasso3_coeff.head(5)
```

```
The R2 Score  is 0.8187863466956775
The MSE  is 0.029811044076694 39
```

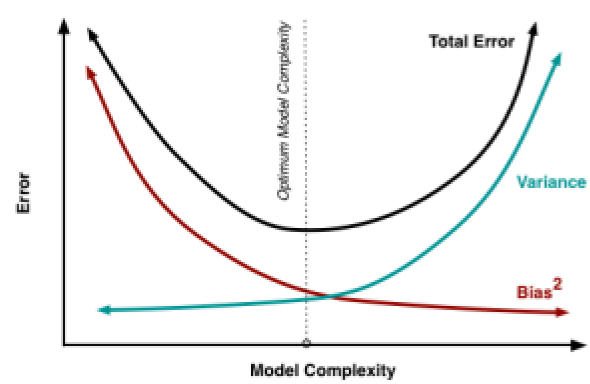
	Lasso Co-Efficient
BedroomAbvGr	0.501497
GarageCars	0.294150
LotFrontage	0.274705
Fireplaces	0.219501
Exterior1st_Stone	0.218091

If we remove the top 5 variables the R2 score is decreasing and MSE is increasing.

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:
Considering the points of Occam's Razor:

- If you have two theories that both explain the observed facts, then you should use the simplest until more evidence comes along.
- The simplest explanation for some phenomenon is more likely to be accurate than more complicated explanations.
- If you have two equally likely solutions to a problem, choose the simplest.
- The explanation requiring the fewest assumptions is most likely to be correct.
- Keep things simple.

Simpler models tend to make mistakes on training data but on unseen data they perform well compared to a complicated model which is overfit just to obtain good values.
The key point to note here is, we have to make the model simple but not very much simpler which again leads to underfitting.
In such circumstances regularization can be used which will erase the thin line between making the model simple and more simpler.



From the above depiction we can clearly say that bias quantifies how well the model will perform on test data. If we consider complex model they are often overfitted to get correct assumptions but they only perform well on trained set of specific data but not on test data thus it has higher error and variance on unseen data whereas the simple model will have a balance between the bias and variance which will ultimately perform better on unseen data.