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: the optimal value of alpha for ridge and lasso regression is 9.0 and 0.001. if we double the alpha value, its R2_score decreases. Below is the image for reference.

Before doubling the alpha value:

	Metrics	Linear Regression	Ridge Regression	Lasso Regression
0	R2_Score(Train)	9.266839e-01	0.914459	0.916170
1	R2_Score (Test)	-4.818295e+17	0.874812	0.875896
2	RSS (Train)	7.485577e+01	87.336873	85.590128
3	RSS (Test)	2.218479e+20	57.640028	57.141184
4	MSE (Train)	2.707695e-01	0.292473	0.289534
5	MSE (Test)	7.108785e+08	0.362351	0.360780

after doubling the alpha value:

	Ridge Regression	Lasso Regression
0	0.907176	0.905854
1	0.869480	0.871051

Top5 most important coefficient for Ridge regression before and after double the alpha value:

```
1 betas['Ridge'].sort_values(ascending=False).head()
```

```
OverallQual_Excellent    0.370237
OverallQual_Very Good    0.265409
Neighborhood_NridgHt     0.253835
Neighborhood_Crawfor     0.243177
Neighborhood_Somerst     0.204905
Name: Ridge, dtype: float64
```

```
1 betas['ridge_double'].sort_values(ascending=False).head()
```

```
OverallQual_Excellent    0.284483
OverallQual_Very Good    0.227210
Neighborhood_NridgHt     0.204993
Neighborhood_Crawfor     0.200981
CentralAir_Y             0.174936
Name: ridge_double, dtype: float64
```

Top5 most important coefficient for Lasso regression before and after double the alpha value:

```
1 betas['Lasso'].sort_values(ascending=False).head()
```

```
OverallQual_Excellent    0.564182
OverallQual_Very Good    0.328329
Neighborhood_Crawfor     0.304465
Neighborhood_NridgHt     0.292958
Neighborhood_Somerst     0.248177
Name: Lasso, dtype: float64
```

```
1 betas['lasso_double'].sort_values(ascending=False).head()
```

```
OverallQual_Excellent    0.537957
OverallQual_Very Good    0.316586
Neighborhood_Crawfor     0.272914
Neighborhood_NridgHt     0.249348
GrLivArea                0.249168
Name: lasso_double, dtype: float64
```

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: the optimal value of alpha for ridge and lasso regression is 9.0 and 0.001. As per the result below, I would choose Ridge regression as it is fetching us a little better result than lasso, but not much.

	Metrics	Linear Regression	Ridge Regression	Lasso Regression
0	R2_Score(Train)	9.266839e-01	0.914459	0.916170
1	R2_Score (Test)	-4.818295e+17	0.874812	0.875896
2	RSS (Train)	7.485577e+01	87.336873	85.590128
3	RSS (Test)	2.218479e+20	57.640028	57.141184
4	MSE (Train)	2.707695e-01	0.292473	0.289534
5	MSE (Test)	7.108785e+08	0.362351	0.360780

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 removing the top5 variable, below is the top5 most important predictor variable.

```
1 lasso_coeff['lasso_coef'].sort_values(ascending=False).head()
```

```
MSZoning_FV              0.253818
OverallQual_Very Good    0.222869
2ndFlrSF                 0.216080
Exterior1st_BrkFace      0.202498
MSZoning_RL              0.193850
Name: lasso_coef, dtype: float64
```

Question 4: How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

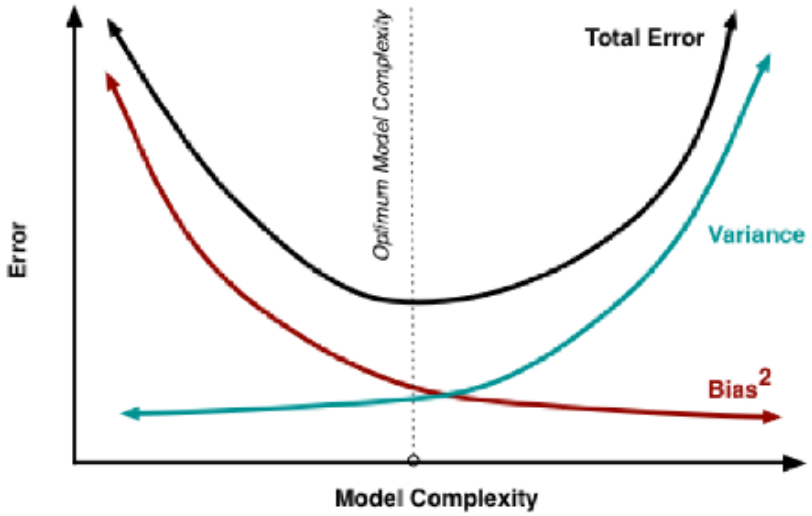
Answer: there are few aspects of a model to be robust and generalizable:

1. Occam's Razor:
   A. a simpler model is more generic.
   B. simpler model requires few training samples.
   C. simpler models are more robust.

2. implications of the same for the accuracy of the model and why:

   Regularization: it is a process used to simplify models. It tries to make a delicate balance between keeping the model simple yet not making it too naïve. it helps the model to avoid the risk of overfitting. And, the fitting procedures involves a loss function known as residual sum of squares or RSS. The coefficients are chosen, such that they minimize this loss function.

   BIAS-VARIANCE trade-off: making a model simple leads to BIAS-VARIANCE trade-off. variance is the degree of the change in the model itself with respect to change in training data. While Bias is how accurate is the model likely to be on test data.
   There is typical trade-off between Bias and Variance: low complexity models have high bias and low variance and high complexity models have low bias but high variance.



   Thus accuracy of the model is keeping the balance between the BIAS and VARIANCE as it minimises the error as shown in the image above.