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 1:

Optimal value of alpha for ridge is 10 and for Lasso is 0.001.

For Lasso Model

Train and test score of Lasso Model with Alpha Value 0.001 are as follows:

```
#final lasso model
alpha = 0.001

lasso = Lasso(alpha=alpha)

lasso.fit(X_train, y_train)
#checking R-squared value for Training set
y_train_pred = lasso.predict(X_train)
print("r2_score on train data for lasso:",metrics.r2_score(y_true=y_train, y_pred=y_train_pred))

# checking R-squared value for test data set
y_test_pred = lasso.predict(X_test)
print("r2_score on test data for Lasso:",metrics.r2_score(y_true=y_test, y_pred=y_test_pred))

print("mean squared error:", mean_squared_error(y_test, y_test_pred))
```

```
r2_score on train data for lasso: 0.9198071885440342
r2_score on test data for Lasso: 0.9111603280187931
mean squared error: 0.01395072276424643
```

Train and Test Scores of Lasso Model with Alpha Value 0.002

```
lasso = Lasso(alpha=0.002)
lasso.fit(X_train,y_train)

y_train_pred = lasso.predict(X_train)
y_test_pred = lasso.predict(X_test)

print("r2_score on train data using Lasso regression:",r2_score(y_true=y_train,y_pred=y_train_pred))
print("r2_score on test data using Lasso regression::",r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
r2_score on train data using Lasso regression: 0.9103332238822941
r2_score on test data using Lasso regression:: 0.9039626087836939
```

After we increase the Alpha value by double i.e. 0.002 in Lasso, the test score have increased by 0.0018. Test score is 0.9129 with alpha value is 0.9129 and 0.9111 with alpha value 0.001. It is clear form above attachments.

Columns for Alpha value of 0.001 and 0.002 are not changed in this Lasso Model as per these below figures.

```
lasso_coeff.sort_values(by='Coeff',ascending=False).head(10)
```

	Feature	Coeff
0	MSSubClass	11.775817
13	BsmtFullBath	0.124241
203	SaleType_Oth	0.093989
50	Neighborhood_Edwards	0.091828
4	OverallCond	0.083144
5	MasVnrArea	0.053747
166	Electrical_FuseF	0.053152
29	MSZoning_RH	0.051723
9	1stFlrSF	0.046500
31	MSZoning_RM	0.044126

The above are the top 10 features selected by the Lasso regression model

```
lasso_coeff.sort_values(by='Coeff',ascending=False).head(10)
```

	Feature	Coeff
0	MSSubClass	11.775817
13	BsmtFullBath	0.124241
203	SaleType_Oth	0.093989
50	Neighborhood_Edwards	0.091828
4	OverallCond	0.083144
5	MasVnrArea	0.053747
166	Electrical_FuseF	0.053152
29	MSZoning_RH	0.051723
9	1stFlrSF	0.046500
31	MSZoning_RM	0.044126

For Ridge Model

Train and test score of Ridge Model with Alpha Value 10 are as follows:

```
#checking the R-squared for training data set
y_train_pred = ridge.predict(X_train)
print("train score for Ridge regression",metrics.r2_score(y_true=y_train, y_pred=y_train_pred))

# checking on test data set
y_test_pred = ridge.predict(X_test)
print("Test score for Ridge regression",metrics.r2_score(y_true=y_test, y_pred=y_test_pred))

# checking RMSE
print(mean_squared_error(y_test, y_test_pred))
```

```
train score for Ridge regression 0.9307409930817205
Test score for Ridge regression 0.9123439784851639
0.01376485107947024
```

Train and test score of Ridge Model with Alpha Value 20 are as follows:

```
ridge = Ridge(alpha = 20)
ridge.fit(X_train,y_train)

y_pred_train = ridge.predict(X_train)
print("r2_score on train data using Ridge regression:",r2_score(y_train,y_pred_train))

y_pred_test = ridge.predict(X_test)
print("r2_score on train data using Ridge regression:",r2_score(y_test,y_pred_test))
```

```
r2_score on train data using Ridge regression: 0.9271022532039753
r2_score on train data using Ridge regression: 0.9129141322001608
```

If we increase the Alpha value 20 in ridge Model, the test score have increased by 0.006 i.e. with alpha 10 test score is 0.9123 and with alpha = 20, the test score is 0.9129

Columns for Alpha value of 0.001 and 0.002 are not changed in this Ridge Model too as per these below figures.

```
ridge_coeff.sort_values(by='Coeff',ascending=False).head(10)
```

	Feature	Coeff
0	MSSubClass	11.793927
50	Neighborhood_Edwards	0.090038
66	Neighborhood_Timber	0.072070
13	BsmtFullBath	0.069639
4	OverallCond	0.066803
166	Electrical_FuseF	0.063713
70	Condition1_PosA	0.057686
29	MSZoning_RH	0.052771
5	MasVnrArea	0.052128
31	MSZoning_RM	0.048665

```
ridge_coeff.sort_values(by='Coeff',ascending=False).head(10)
```

	Feature	Coeff
0	MSSubClass	11.793927
50	Neighborhood_Edwards	0.090038
66	Neighborhood_Timber	0.072070
13	BsmtFullBath	0.069639
4	OverallCond	0.066803
166	Electrical_FuseF	0.063713
70	Condition1_PosA	0.057686
29	MSZoning_RH	0.052771
5	MasVnrArea	0.052128
31	MSZoning_RM	0.048665

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: Lambda value for Ridge regression is 10 and Lasso Regression is 0.001 as they had their best to built robust Model.

Lasso Model with lambda values 0.001 are 0.9198 for train Set and 0.9111 for test set

```
#final lasso model
alpha = 0.001

lasso = Lasso(alpha=alpha)

lasso.fit(X_train, y_train)
#checking R-squared value for Training set
y_train_pred = lasso.predict(X_train)
print("r2_score on train data for lasso:",metrics.r2_score(y_true=y_train, y_pred=y_train_pred))

# checking R-squared value for test data set
y_test_pred = lasso.predict(X_test)
print("r2_score on test data for Lasso:",metrics.r2_score(y_true=y_test, y_pred=y_test_pred))

print("mean squared error:", mean_squared_error(y_test, y_test_pred))
```

```
r2_score on train data for lasso: 0.9198071885440342
r2_score on test data for Lasso: 0.9111603280187931
mean squared error: 0.01395072276424643
```

Ridge Model with lambda values 10 are 0.9307 for train Set and 0.9123 for test set.

```
#checking the R-squared for training data set
y_train_pred = ridge.predict(X_train)
print("train score for Ridge regression",metrics.r2_score(y_true=y_train, y_pred=y_train_pred))

# checking on test data set
y_test_pred = ridge.predict(X_test)
print("Test score for Ridge regression",metrics.r2_score(y_true=y_test, y_pred=y_test_pred))

# checking RMSE
print(mean_squared_error(y_test, y_test_pred))
```

```
train score for Ridge regression 0.9307409930817205
Test score for Ridge regression 0.9123439784851639
0.01376485107947024
```

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: Following are the best the best predictor variables after dropping 05 most predictor variables from original Lasso Model:

Original 05 predictors

```
Lasso_sort=lasso_coeff.sort_values(by='Coeff',ascending=False).head(5)
Lasso_sort
```

	Feature	Coeff
0	MSSubClass	11.775817
13	BsmtFullBath	0.124241
203	SaleType_Oth	0.093989
50	Neighborhood_Edwards	0.091828
4	OverallCond	0.083144

New Predictors are in RHS Figure

```
lasso_coef.sort_values(by='Coef',ascending=False).head(10)
```

	Feaure	Coef
1	LotArea	11.793927
0	LotFrontage	11.725284
51	Neighborhood_NAmes	0.090038
67	Condition1_PosN	0.072070
14	BedroomAbvGr	0.069639
5	BsmtFinSF2	0.066803
167	KitchenQual_Gd	0.063713
71	Condition1_RRNn	0.057686
30	LotShape_IR3	0.052771
6	TotalBsmtSF	0.052128

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 4 : The Model should neither be under fit (robust) nor over fit (generalizable). Accuracy can be checked by adjusted R squared value of Model. Ideally difference between R square score of training and test data set of same model should be as minimum as possible. This indicates that model is working in same manner on both seen and unseen data points.