Question-1:

a)What is the optimal value of alpha for ridge and lasso regression?

Optimal Value of alpha for ridge regression : 20
Optimal Value of alpha for lasso regression : 200

b)What will be the changes in the model if you choose double the value of alpha for both ridge and lasso?

Ridge Value if I will Double and make : 40

```
# Finding the accuracy of the model on train and test data given best alpha value * 2 = 40.

ridge = Ridge(alpha = 40)
ridge.fit(X_train,y_train)

y_pred_train = ridge.predict(X_train)
print(r2_score(y_train,y_pred_train))

y_pred_test = ridge.predict(X_test)
print(r2_score(y_test,y_pred_test))
```

```
0.8948904095221333
0.8993751248794412
```

```
# Finding the list of features with Co-efficient values.

model_parameter = list(abs(ridge.coef_))
model_parameter.insert(0,ridge.intercept_)
cols = house_train.columns
cols.insert(0,'constant')
ridge_coef = pd.DataFrame(list(zip(cols,model_parameter)))
ridge_coef.columns = ['Feaure','Coef']
```

```
# Finding Top 10 most important Predictor Variable/Feature  after building the Ridge Model ..

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

	Feaure	Coef
0	MSSubClass	196730.228923
6	OverallCond	15021.575217
22	BsmtFullBath	12594.418120
195	Neighborhood__OldTown	11480.778608
194	Neighborhood__NridgHt	10502.740765
53	BsmtExposure__Mn	9910.223701
20	LowQualFinSF	9867.069179
201	Neighborhood__Timber	8949.885864
125	Condition2__RRAn	8867.905924
144	Condition1__PosA	8608.285317

Note : Below points are same applies for Ridge and Lasso Regression

1. *As Alpha value is getting increased so RSS value for the model will also steadily increase so the accuracy is getting decrease .*
2. *As Alpha value is getting increased so coefficient value for the model will also change which may leads to get more error.*
3. *As Alpha value is getting increased so coefficient value for the model will also change for which model complexity may increase while leads overfitting may decrease and bias may increase.*

Lasso Value if I will Double and make : 400

```python
# Finding the accuracy of the model on train and test data given best alpha value * 2 = 400.

lasso = Lasso(alpha=400)
lasso.fit(X_train,y_train)

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

print(r2_score(y_true=y_train,y_pred=y_train_pred))
print(r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
0.8827264741516115
0.8957609182765225
```

```python
# Finding the list of features with Co-efficient values.

model_parameter = list(abs(lasso.coef_))
model_parameter.insert(0,lasso.intercept_)
cols = house_train.columns
cols.insert(0,'constant')
lasso_coef = pd.DataFrame(list(zip(cols,model_parameter)))
lasso_coef.columns = ['Feaure','Coef']
```

```python
# Finding Top 10 most important Predictor Variable/Feature  after building the Lasso Model ..

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

	Feaure	Coef
0	MSSubClass	205883.009514
22	BsmtFullBath	24534.855229
195	Neighborhood__OldTown	18213.394638
6	OverallCond	16746.425082
194	Neighborhood__NridgHt	14590.989248
201	Neighborhood__Timber	12924.615412
53	BsmtExposure__Mn	11947.708293
140	SaleType__Oth	9464.902497
144	Condition1__PosA	8558.027538
121	Condition2__Feedr	8511.345708

c) What will be the most important predictor variables after the change is implemented?

For Ridge Regression :

After Changes :

	Feaure	Coef
0	MSSubClass	196730.228923
6	OverallCond	15021.575217
22	BsmtFullBath	12594.418120
195	Neighborhood__OldTown	11480.778608
194	Neighborhood__NridgHt	10502.740765
53	BsmtExposure__Mn	9910.223701
20	LowQualFinSF	9867.069179
201	Neighborhood__Timber	8949.885864
125	Condition2__RRAn	8867.905924
144	Condition1__PosA	8608.285317

Before changes :

	Feaure	Coef
0	MSSubClass	186021.819807
125	Condition2__RRAn	16242.484926
195	Neighborhood__OldTown	16077.993295
115	Functional__Maj2	15236.707817
194	Neighborhood__NridgHt	15182.422532
6	OverallCond	14597.942334
201	Neighborhood__Timber	13838.482143
22	BsmtFullBath	13214.663290
190	Neighborhood__NAmes	12284.862719
53	BsmtExposure__Mn	12223.227593

For Lasso Regression :

After Changes :

	Feaure	Coef
0	MSSubClass	205883.009514
22	BsmtFullBath	24534.855229
195	Neighborhood__OldTown	18213.394638
6	OverallCond	16746.425082
194	Neighborhood__NridgHt	14590.989248
201	Neighborhood__Timber	12924.615412
53	BsmtExposure__Mn	11947.708293
140	SaleType__Oth	9464.902497
144	Condition1__PosA	8558.027538
121	Condition2__Feedr	8511.345708

Before changes :

	Feaure	Coef
0	MSSubClass	198696.992856
125	Condition2__RRAn	103523.998700
115	Functional__Maj2	38396.543202
195	Neighborhood__OldTown	25692.779836
201	Neighborhood__Timber	24549.056473
22	BsmtFullBath	24112.288837
194	Neighborhood__NridgHt	22877.518561
140	SaleType__Oth	16555.429242
53	BsmtExposure__Mn	14797.788765
6	OverallCond	14769.447654

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?

For Ridge Regression the Accuracy is :

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

y_pred_train = ridge.predict(X_train)
print(r2_score(y_train,y_pred_train))

y_pred_test = ridge.predict(X_test)
print(r2_score(y_test,y_pred_test))
```

```
0.9022470165195846
0.9023647804749249
```

For Lasso Regression the Accuracy is :

```
lasso = Lasso(alpha=200)
lasso.fit(X_train,y_train)

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

print(r2_score(y_true=y_train,y_pred=y_train_pred))
print(r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
0.90793551097258
0.9062265564087699
```

In the above assignment for both the model Ridge and Lasso is giving almost similar Accuracy near to 90% . But there are some advantage of Lasso Regression over Ridge Regression Model Like : Lasso regression not only publishing high values of the coefficients but actually setting them to zero if they are not relevant. So, it might end up with fewer features / Predictor variables included in the model before started, which is a huge advantage.

That is why I will prefer to go with Lasso Regression 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?

In the Above assignment After creating Lasso Regression Model the top 5 most important Predictor Variable I am getting as :

Housing_dataset =Housing_dataset.drop(columns=['MSSubClass',‘Condition2’,'Functional','Neighborhood’,'BsmtFullBath'])

Where

'MSSubClass' (Positively Correlated)
‘Condition2’ (Negatively Correlated)
'Functional' (Positively Correlated)
'Neighborhood’ (Positively Correlated)
'BsmtFullBath' (Positively Correlated)

So If I am dropping the feature from input file and then creating the model. Now most 5 predictor variables are:

Lasso Regression After dropping the top 5 variable : LotFrontage, BsmtHalfBath, overallCond, BsmtExposure, SaleType (All Features are positively correlated)

```
# finding the best Param / alpha
model_cv_lasso.best_params_
```

```
{'alpha': 400}
```

```
# Finding the accuracy of the model on train and test data given best alpha value.

lasso = Lasso(alpha=400)
lasso.fit(X_train,y_train)

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

print(r2_score(y_true=y_train,y_pred=y_train_pred))
print(r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
0.8713342998183783
0.8855745851017353
```

```
# Finding the list of features with Co-efficient values.

model_parameter = list(lasso.coef_)
model_parameter.insert(0,lasso.intercept_)
cols = house_train.columns
cols.insert(0,'constant')
lasso_coef = pd.DataFrame(list(zip(cols,model_parameter)))
lasso_coef.columns = ['Feaure','Coef']
```

```
# Finding Top 5 most important Predictor Variable/Feature  after building the Lasso Model ..

lasso_coef.sort_values(by='Coef',ascending=False).head(5)
```

	Feaure	Coef
0	LotFrontage	213400.501724
21	BsmtHalfBath	23012.938418
5	OverallCond	18078.840303
51	BsmtExposure_Mn	10780.879182
126	SaleType_Oth	10641.706412

Ridge Regression After dropping the top 5 variable :

```
# finding the best Param / alpha
model_cv.best_params_
```

```
{'alpha': 50}
```

```
# Finding the accuracy of the model on train and test data given best alpha value.

ridge = Ridge(alpha = 50)
ridge.fit(X_train,y_train)

y_pred_train = ridge.predict(X_train)
print(r2_score(y_train,y_pred_train))

y_pred_test = ridge.predict(X_test)
print(r2_score(y_test,y_pred_test))
```

```
0.8790610614507977
0.8886621360657327
```

```
# Finding the list of features with Co-efficient values.
model_parameter = list(ridge.coef_)
model_parameter.insert(0,ridge.intercept_)
cols = house_train.columns
cols.insert(0,'constant')
ridge_coef = pd.DataFrame(list(zip(cols,model_parameter)))
ridge_coef.columns = ['Feaure','Coef']
```

```
# Finding Top 5 most important Predictor Variable/Feature  after building the Ridge Model ..
ridge_coef.sort_values(by='Coef',ascending=False).head(5)
```

	Feaure	Coef
0	LotFrontage	206539.891686
5	OverallCond	16305.538185
21	BsmtHalfBath	12026.666023
19	LowQualFinSF	9491.572268
51	BsmtExposure__Mn	8635.734874

Note : For coding Reference Please find the file: Final After Removing 5 important character from Lasso (Assignment Part 2).ipynb

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?

Here For the above Problem statement we have created model using Ridge Regression and Lasso Regression., But If we will compare the general Model irrespective of method we have received similar accuracy that is 90% for both the model.

Also If I will compare between training and test dataset and accuracy for each model , if training dataset is giving 90% accuracy then test dataset also giving 90% Accuracy . that indicates that model tuning is perfectly ok as it is behaving same irrespective of dataset as well as model. So we can also conclude here that model is having low bias (Difference between predictions made by the model and the correct value that we are trying to predict)and low variance(Variability of the model predictions on the test dataset).So We can say model is more robust as there is no scenario of overfitting or underfitting while choosing any of the model.

Also If we will compare the feature or predictions variables for both the method model is giving almost similar result . So we can consider as generalized.