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 is : 2.0

Top ten contributors(with coefficient values) to affect the sale price are :

```
OverallQual_Excellent                                         0.272767
OverallQual_Very Good                                         0.192401
Neighborhood_NridgHt                                          0.160908
OverallCond_Excellent                                         0.147839
1stFlrSF                                                      0.142320
MSSubClass_1-STORY 1946 & NEWER ALL STYLES                    0.141089
Neighborhood_Crawfor                                          0.138269
MSSubClass_2-STORY 1946 & NEWER                               0.132113
MSSubClass_SPLIT FOYER                                        0.127381
Neighborhood_Somerst                                          0.124865
```

lasso is: 0.0001

Top ten contributors(with coefficient values) to affect the sale price are :

```
OverallQual_Excellent                                         0.286893
OverallQual_Very Good                                         0.195141
OverallCond_Excellent                                         0.166315
Neighborhood_NridgHt                                          0.159080
Neighborhood_Crawfor                                          0.145858
MSSubClass_1-STORY 1946 & NEWER ALL STYLES                    0.142981
MSSubClass_SPLIT FOYER                                        0.138020
1stFlrSF                                                      0.138009
MSSubClass_2-STORY 1946 & NEWER                               0.133752
Neighborhood_StoneBr                                          0.129646
```

After doubling the value of alpha i.e. if alpha for ridge regression is 4.0

Top ten contributors(with coefficient values) to affect the sale price are :

```
OverallQual_Excellent                                         0.259768
OverallQual_Very Good                                         0.190729
Neighborhood_NridgHt                                          0.160884
1stFlrSF                                                      0.146483
MSSubClass_1-STORY 1946 & NEWER ALL STYLES                    0.138072
MSSubClass_2-STORY 1946 & NEWER                               0.130266
Neighborhood_Crawfor                                          0.128710
OverallCond_Excellent                                         0.127794
2ndFlrSF                                                      0.125130
Neighborhood_Somerst                                          0.121756
```

and lasso regression is 0.0002, the coefficient values got changed.
Top 15 contributors(with coefficient values) to affect the sale price are :

```
OverallQual_Excellent                                          0.287468
OverallQual_Very Good                                          0.196157
Neighborhood_NridgHt                                           0.157213
OverallCond_Excellent                                          0.155948
Neighborhood_Crawfor                                           0.140475
MSSubClass_1-STORY 1946 & NEWER ALL STYLES                     0.140115
1stFlrSF                                                       0.139786
MSSubClass_2-STORY 1946 & NEWER                                0.130921
MSSubClass_SPLIT FOYER                                         0.128182
Neighborhood_NoRidge                                           0.122896
Neighborhood_Somerst                                           0.122528
2ndFlrSF                                                       0.122129
BsmtExposure_Gd                                                0.120430
Neighborhood_StoneBr                                           0.119696
OverallQual_Good                                               0.109921
```

The most important predictor variable still remains same i.e. OverallQual(Rating the overall material and finish of the house
). The sale price is maximum when the OverallQual is Excellent.

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: Will choose Lasso regression over Ridge regression because using Lasso regression number of features also are reduced e.g. the coefficient of Exterior1st_CBlock is determined as 0. Ridge regression does not help with feature reduction but only reduces the coefficient value of less significant variables to near zero.

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 following five most important predictor variables alongwith the coefficient values i.e.
OverallQual_Excellent   0.286893

OverallQual_Very Good	0.195141
OverallCond_Excellent	0.166315
Neighborhood_NridgHt	0.159080
Neighborhood_Crawfor	0.145858

The following new top 5 predictor variables(alongwith coefficient values) are identified using Lasso regression:

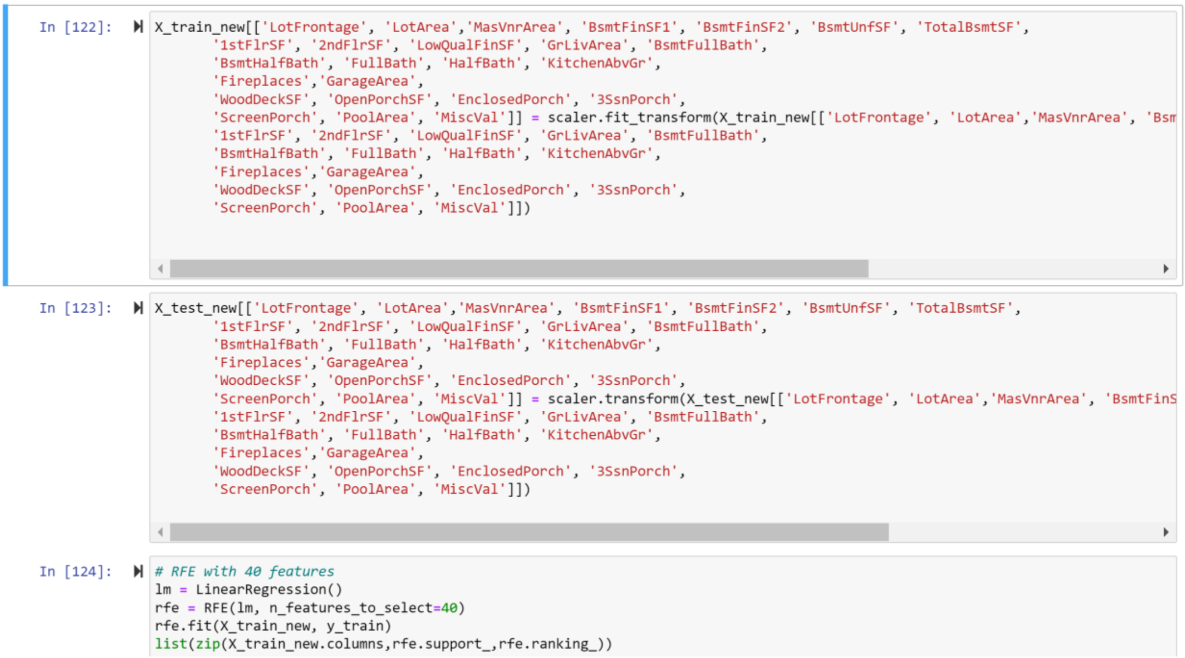
MSSubClass_2-1/2 STORY ALL AGES	0.300364
MSSubClass_2-STORY 1946 & NEWER	0.269665
MSSubClass_2-STORY 1945 & OLDER	0.217977
1stFlrSF	0.153573
BldgType_2fmCon	0.136378

Steps performed are:

1. Dropped the columns from from X_train and X_test
   X_test_new = X_test.drop(["OverallQual_Excellent", "OverallQual_Very Good", "OverallCond_Excellent", "Neighborhood_NridgHt", "Neighborhood_Crawfor"], axis=1)

   X_train_new = X_train.drop(["OverallQual_Excellent", "OverallQual_Very Good", "OverallCond_Excellent", "Neighborhood_NridgHt", "Neighborhood_Crawfor"], axis=1)

2. Performed scaling on numerical fields of test/train data and select 40 features using RFE.

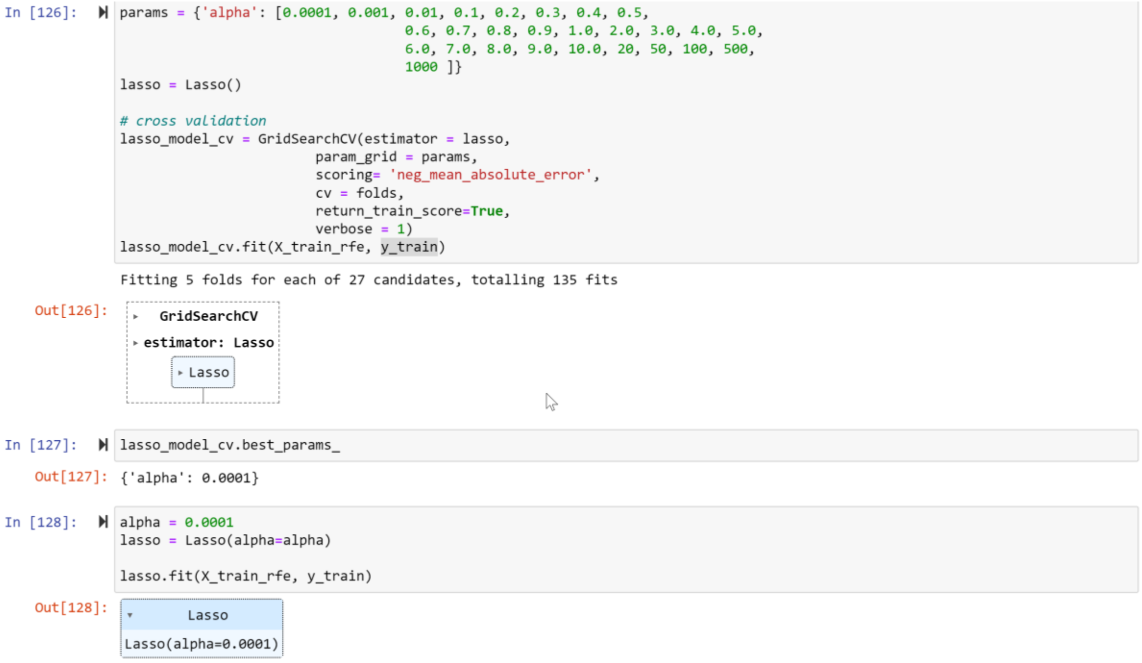


In [122]:
```
X_train_new[['LotFrontage', 'LotArea','MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',
       '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
       'BsmtHalfBath', 'FullBath', 'HalfBath', 'KitchenAbvGr',
       'Fireplaces','GarageArea',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal']] = scaler.fit_transform(X_train_new[['LotFrontage', 'LotArea','MasVnrArea', 'Bsm
       '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
       'BsmtHalfBath', 'FullBath', 'HalfBath', 'KitchenAbvGr',
       'Fireplaces','GarageArea',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal']])
```

In [123]:
```
X_test_new[['LotFrontage', 'LotArea','MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',
       '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
       'BsmtHalfBath', 'FullBath', 'HalfBath', 'KitchenAbvGr',
       'Fireplaces','GarageArea',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal']] = scaler.transform(X_test_new[['LotFrontage', 'LotArea','MasVnrArea', 'BsmtFinS
       '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
       'BsmtHalfBath', 'FullBath', 'HalfBath', 'KitchenAbvGr',
       'Fireplaces','GarageArea',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal']])
```

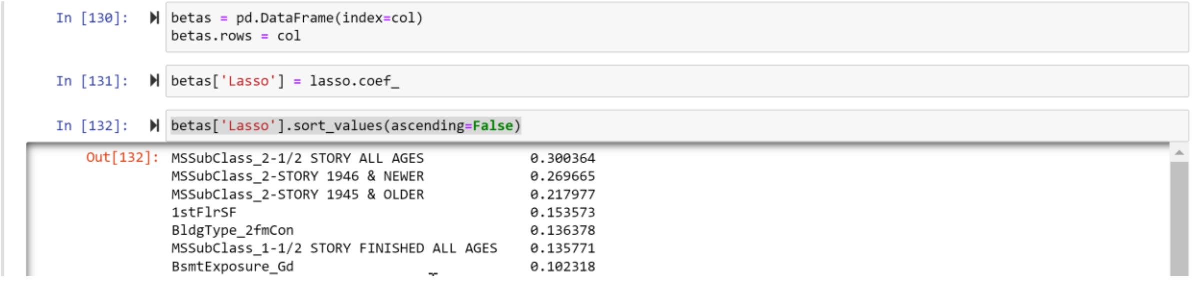
In [124]:
```
# RFE with 40 features
lm = LinearRegression()
rfe = RFE(lm, n_features_to_select=40)
rfe.fit(X_train_new, y_train)
list(zip(X_train_new.columns,rfe.support_,rfe.ranking_))
```

3. Perform lasso regression:



```
In [126]:  params = {'alpha': [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5,
                               0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0, 4.0, 5.0,
                               6.0, 7.0, 8.0, 9.0, 10.0, 20, 50, 100, 500,
                               1000 ]}
           lasso = Lasso()

           # cross validation
           lasso_model_cv = GridSearchCV(estimator = lasso,
                                   param_grid = params,
                                   scoring= 'neg_mean_absolute_error',
                                   cv = folds,
                                   return_train_score=True,
                                   verbose = 1)
           lasso_model_cv.fit(X_train_rfe, y_train)

           Fitting 5 folds for each of 27 candidates, totalling 135 fits

Out[126]:  ▸ GridSearchCV
           ▸ estimator: Lasso
               ▸ Lasso

In [127]:  lasso_model_cv.best_params_

Out[127]:  {'alpha': 0.0001}

In [128]:  alpha = 0.0001
           lasso = Lasso(alpha=alpha)

           lasso.fit(X_train_rfe, y_train)

Out[128]:  ▾        Lasso
           Lasso(alpha=0.0001)
```

4. Print coefficient values in decreasing order:



```
In [130]:  betas = pd.DataFrame(index=col)
           betas.rows = col

In [131]:  betas['Lasso'] = lasso.coef_

In [132]:  betas['Lasso'].sort_values(ascending=False)

Out[132]:  MSSubClass_2-1/2 STORY ALL AGES            0.300364
           MSSubClass_2-STORY 1946 & NEWER            0.269665
           MSSubClass_2-STORY 1945 & OLDER            0.217977
           1stFlrSF                                   0.153573
           BldgType_2fmCon                            0.136378
           MSSubClass_1-1/2 STORY FINISHED ALL AGES   0.135771
           BsmtExposure_Gd                            0.102318
```

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: A model is robust and generalisable if it works well on unseen data. In order to make the model robust and generalisable:

1. Split the data into the following three categories:
   Training data
   Validation data
   Test data
2. Remove any multicollinearity existing between the independent variables. This can be done by analysing the correlation between variables(graphically by plotting correlation table/heatmap) and by looking at the VIF values. Ideally all variables should have a VIF value of less than or equal to 5. Remove all correlated redundant variables.
3. Perform scaling on numerical variables.
4. Perform hyperparameter tuning e.g. number of feature selection to avoid model overfitting.

5. Detect outliers and perform data cleaning nicely e.g. replace missing values in the dataset with median if the outliers exist or with mean if there are no outliers.

Following are the implications for model accuracy:

1. More generalized models are simple. They work well on unseen data in general, have low variance but greater bias.
2. With more generalized models overfitting is avoided.
3. When a model becomes more generalized, its accuracy decreases.