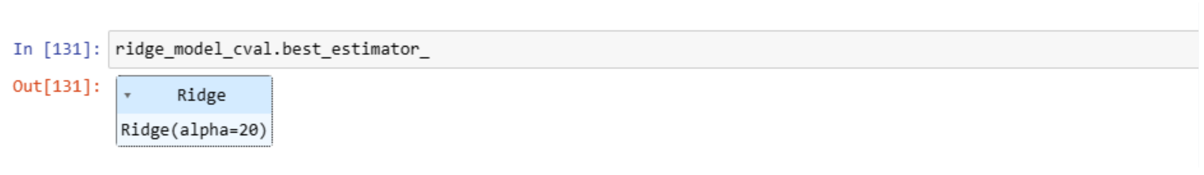
Question 1:

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?

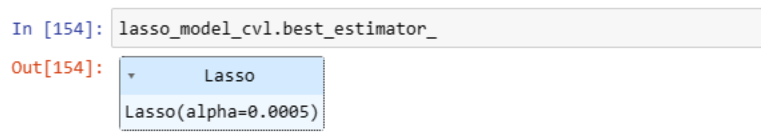
Answer:

For ridge the optimal value of alpha is 20



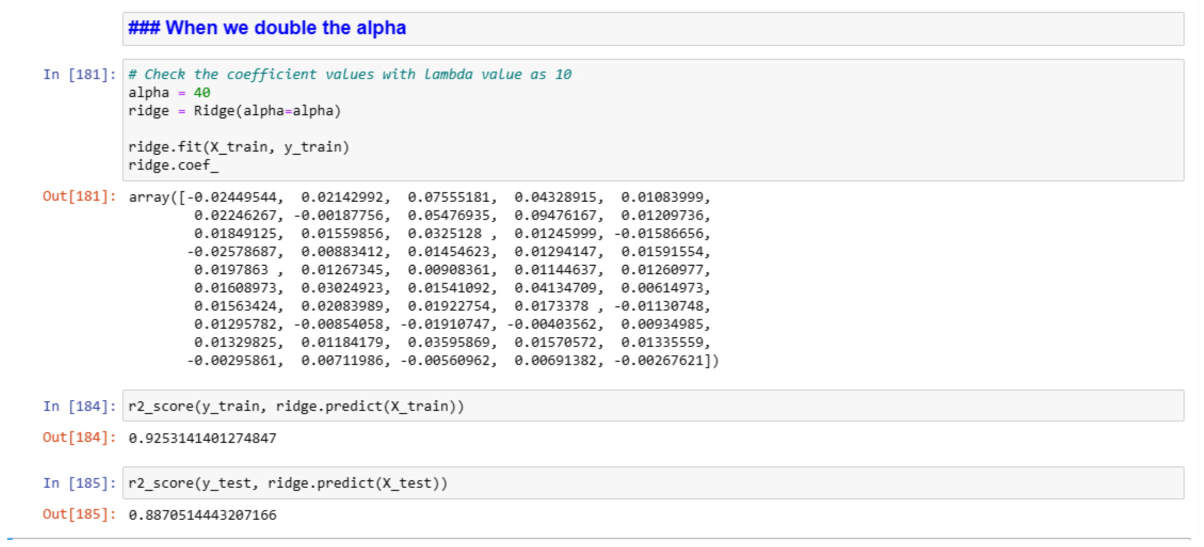
```
In [131]: ridge_model_cval.best_estimator_
Out[131]: ▾ Ridge
          Ridge(alpha=20)
```

For lasso the optimal value of alpha is 0.0005.



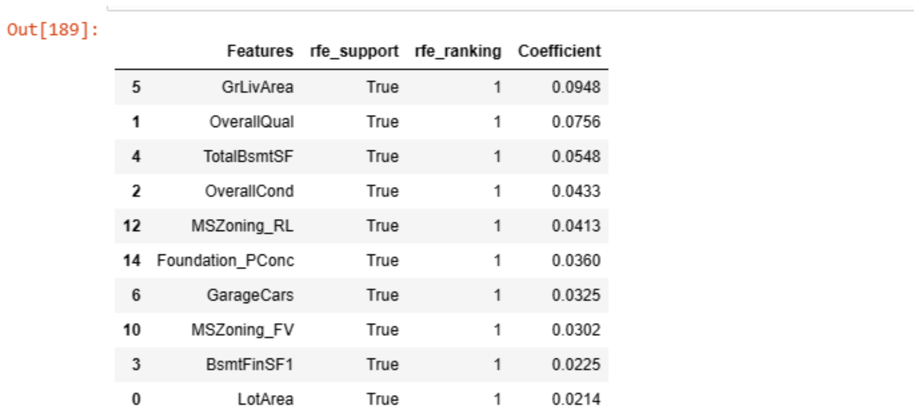
```
In [154]: lasso_model_cvl.best_estimator_
Out[154]: ▾ Lasso
          Lasso(alpha=0.0005)
```

When we doubled the alpha for ridge: The r2 score decreases.



```
### When we double the alpha

In [181]: # Check the coefficient values with lambda value as 10
          alpha = 40
          ridge = Ridge(alpha=alpha)

          ridge.fit(X_train, y_train)
          ridge.coef_
Out[181]: array([-0.02449544,  0.02142992,  0.07555181,  0.04328915,  0.01083999,
                  0.02246267, -0.00187756,  0.05476935,  0.09476167,  0.01209736,
                  0.01849125,  0.01559856,  0.0325128 ,  0.01245999, -0.01586656,
                 -0.02578687,  0.00883412,  0.01454623,  0.01294147,  0.01591554,
                  0.0197863 ,  0.01267345,  0.00908361,  0.01144637,  0.01260977,
                  0.01608973,  0.03024923,  0.01541092,  0.04134709,  0.00614973,
                  0.01563424,  0.02083989,  0.01922754,  0.0173378 , -0.01130748,
                  0.01295782, -0.00854058, -0.01910747, -0.00403562,  0.00934985,
                  0.01329825,  0.01184179,  0.03595869,  0.01570572,  0.01335559,
                 -0.00295861,  0.00711986, -0.00560962,  0.00691382, -0.00267621])

In [184]: r2_score(y_train, ridge.predict(X_train))
Out[184]: 0.9253141401274847

In [185]: r2_score(y_test, ridge.predict(X_test))
Out[185]: 0.8870514443207166
```

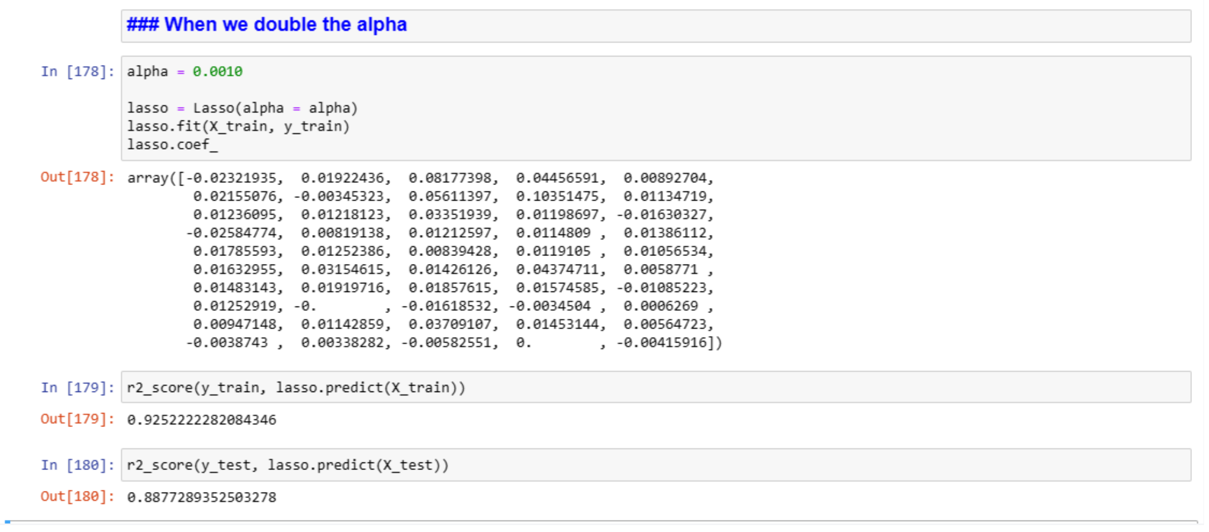
New top 5 features after doubling ridge alpha:



Out[189]:

	Features	rfe_support	rfe_ranking	Coefficient
5	GrLivArea	True	1	0.0948
1	OverallQual	True	1	0.0756
4	TotalBsmtSF	True	1	0.0548
2	OverallCond	True	1	0.0433
12	MSZoning_RL	True	1	0.0413
14	Foundation_PConc	True	1	0.0360
6	GarageCars	True	1	0.0325
10	MSZoning_FV	True	1	0.0302
3	BsmtFinSF1	True	1	0.0225
0	LotArea	True	1	0.0214

When we double alpha for lasso: The r2 score decreases



### When we double the alpha

In [178]:
```
alpha = 0.0010

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

Out[178]:
```
array([-0.02321935,  0.01922436,  0.08177398,  0.04456591,  0.00892704,
        0.02155076, -0.00345323,  0.05611397,  0.10351475,  0.01134719,
        0.01236095,  0.01218123,  0.03351939,  0.01198697, -0.01630327,
       -0.02584774,  0.00819138,  0.01212597,  0.0114809 ,  0.01386112,
        0.01785593,  0.01252386,  0.00839428,  0.0119105 ,  0.01056534,
        0.01632955,  0.03154615,  0.01426126,  0.04374711,  0.0058771 ,
        0.01483143,  0.01919716,  0.01857615,  0.01574585, -0.01085223,
        0.01252919, -0.        , -0.01618532, -0.0034504 ,  0.0006269 ,
        0.00947148,  0.01142859,  0.03709107,  0.01453144,  0.00564723,
       -0.0038743 ,  0.00338282, -0.00582551,  0.        , -0.00415916])
```

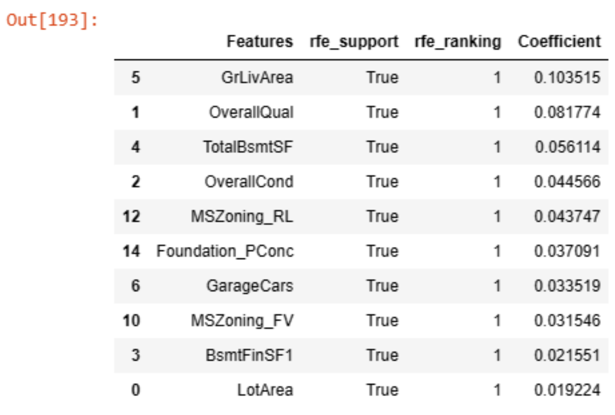
In [179]:
```
r2_score(y_train, lasso.predict(X_train))
```

Out[179]:
```
0.9252222282084346
```

In [180]:
```
r2_score(y_test, lasso.predict(X_test))
```

Out[180]:
```
0.8877289352503278
```

New top 5 features:



Out[193]:

	Features	rfe_support	rfe_ranking	Coefficient
5	GrLivArea	True	1	0.103515
1	OverallQual	True	1	0.081774
4	TotalBsmtSF	True	1	0.056114
2	OverallCond	True	1	0.044566
12	MSZoning_RL	True	1	0.043747
14	Foundation_PConc	True	1	0.037091
6	GarageCars	True	1	0.033519
10	MSZoning_FV	True	1	0.031546
3	BsmtFinSF1	True	1	0.021551
0	LotArea	True	1	0.019224

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?

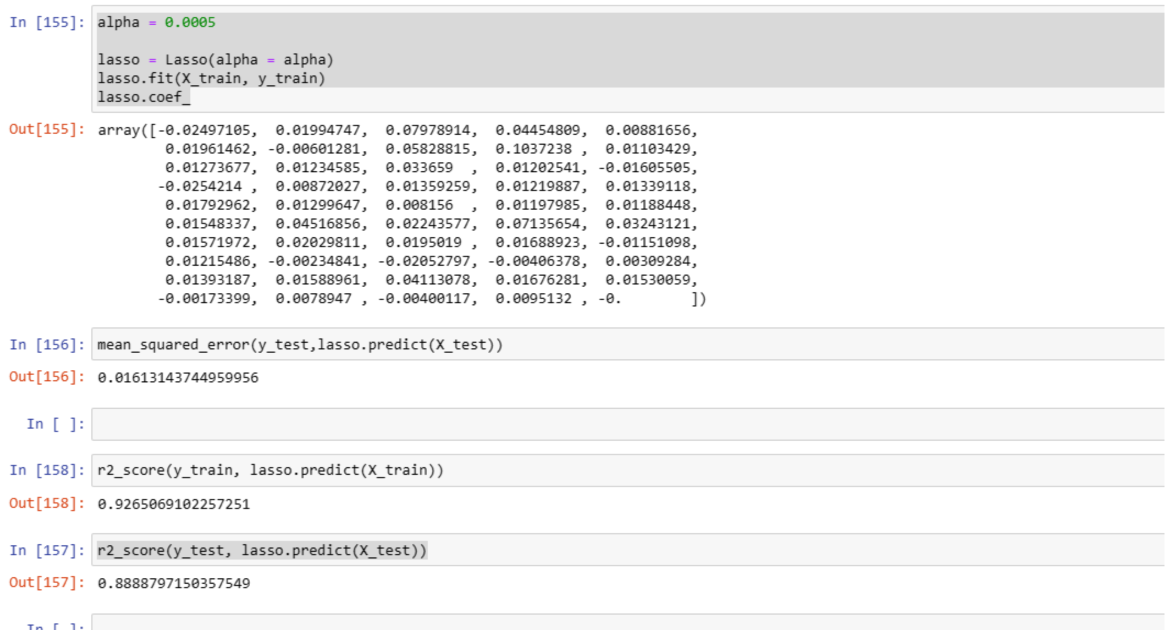
Answer:

We will choose lambda value of lasso because r2 score is better than ridge one.

R2 score of lasso:

Train score: 0.9265

Test Score: 0.8888

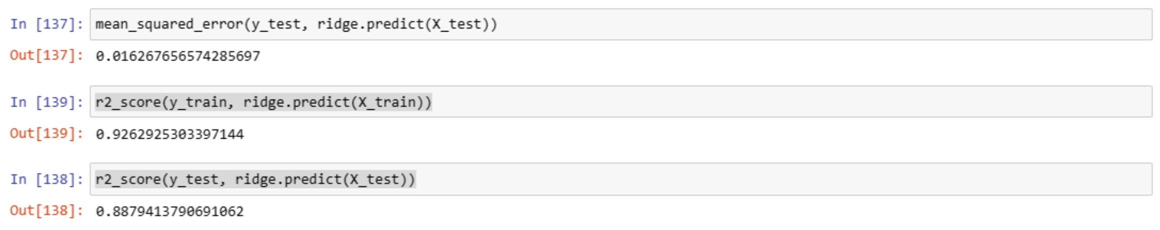


```
In [155]: alpha = 0.0005

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

Out[155]: array([-0.02497105,  0.01994747,  0.07978914,  0.04454809,  0.00881656,
                  0.01961462, -0.00601281,  0.05828815,  0.1037238 ,  0.01103429,
                  0.01273677,  0.01234585,  0.033659  ,  0.01202541, -0.01605505,
                 -0.0254214 ,  0.00872027,  0.01359259,  0.01219887,  0.01339118,
                  0.01792962,  0.01299647,  0.008156  ,  0.01197985,  0.01188448,
                  0.01548337,  0.04516856,  0.02243577,  0.07135654,  0.03243121,
                  0.01571972,  0.02029811,  0.0195019 ,  0.01688923, -0.01151098,
                  0.01215486, -0.00234841, -0.02052797, -0.00406378,  0.00309284,
                  0.01393187,  0.01588961,  0.04113078,  0.01676281,  0.01530059,
                 -0.00173399,  0.0078947 , -0.00400117,  0.0095132 , -0.        ])

In [156]: mean_squared_error(y_test,lasso.predict(X_test))

Out[156]: 0.01613143744959956

  In [ ]:

In [158]: r2_score(y_train, lasso.predict(X_train))

Out[158]: 0.9265069102257251

In [157]: r2_score(y_test, lasso.predict(X_test))

Out[157]: 0.8888797150357549
```

R2 score for Ridge:

Train score: 0.9262

Test Score: 0.8879



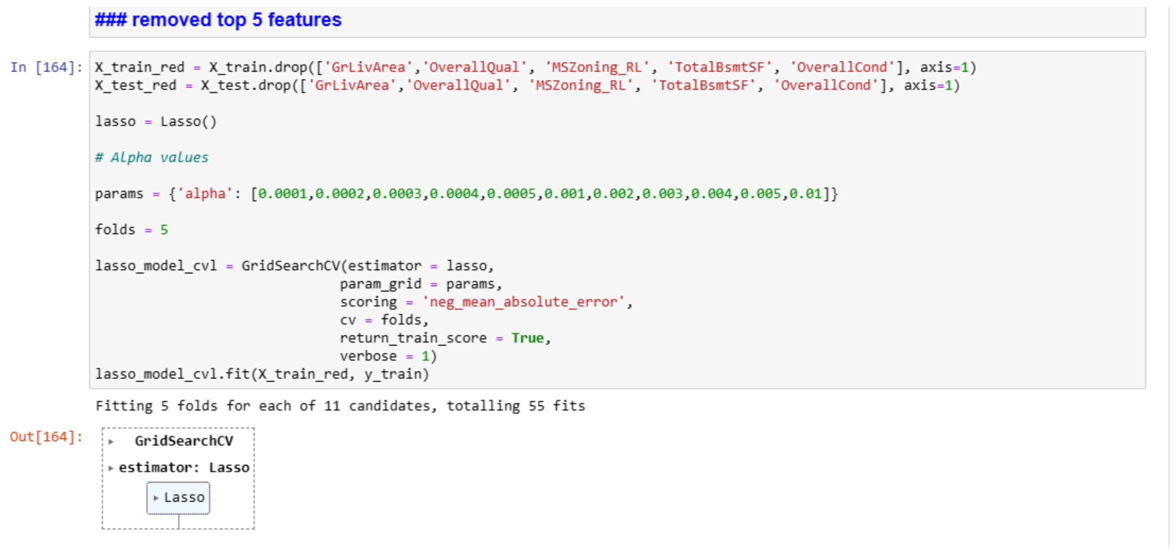
```
In [137]: mean_squared_error(y_test, ridge.predict(X_test))

Out[137]: 0.016267656574285697

In [139]: r2_score(y_train, ridge.predict(X_train))

Out[139]: 0.9262925303397144

In [138]: r2_score(y_test, ridge.predict(X_test))

Out[138]: 0.8879413790691062
```

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 top 5 features



### removed top 5 features

```
In [164]: X_train_red = X_train.drop(['GrLivArea','OverallQual', 'MSZoning_RL', 'TotalBsmtSF', 'OverallCond'], axis=1)
          X_test_red = X_test.drop(['GrLivArea','OverallQual', 'MSZoning_RL', 'TotalBsmtSF', 'OverallCond'], axis=1)

          lasso = Lasso()

          # Alpha values

          params = {'alpha': [0.0001,0.0002,0.0003,0.0004,0.0005,0.001,0.002,0.003,0.004,0.005,0.01]}

          folds = 5

          lasso_model_cvl = GridSearchCV(estimator = lasso,
                                  param_grid = params,
                                  scoring = 'neg_mean_absolute_error',
                                  cv = folds,
                                  return_train_score = True,
                                  verbose = 1)
          lasso_model_cvl.fit(X_train_red, y_train)

          Fitting 5 folds for each of 11 candidates, totalling 55 fits

Out[164]: GridSearchCV
          estimator: Lasso
            Lasso
```

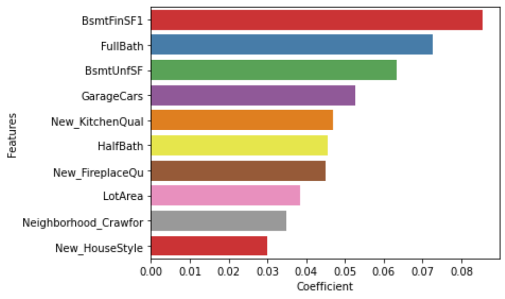
Below are new top 5 features after removing top 5 features:

- BsmtFinSF1
- FullBath
- BsmtUnfSF
- GarageCars
- New_KitchenQual

Out[174]:

	Features	rfe_support	rfe_ranking	Coefficient
1	BsmtFinSF1	True	1	0.085478
3	FullBath	True	1	0.072656
2	BsmtUnfSF	True	1	0.063297
5	GarageCars	True	1	0.052717
9	New_KitchenQual	True	1	0.047018
4	HalfBath	True	1	0.045439
10	New_FireplaceQu	True	1	0.045035
0	LotArea	True	1	0.038325
12	Neighborhood_Crawfor	True	1	0.034913
11	New_HouseStyle	True	1	0.029864

In [175]:
```
plt.figure(figsize = (20,20))
plt.subplot(4,3,1)
se.barplot(y = 'Features', x = 'Coefficient', palette = 'Set1', data = dtl_ten)
plt.show()
```



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:

The balance between bias and variance makes model more robust and generalised. We use ridge or lasso technique to achieve this. This will ensure that the model is robust and generalized