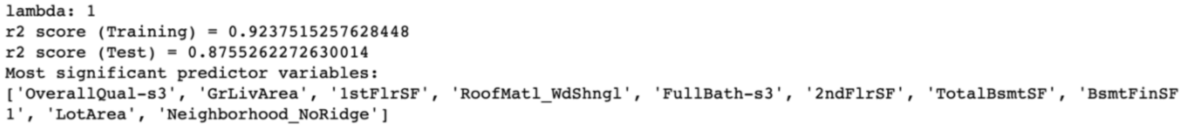
Subjective Questions

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?

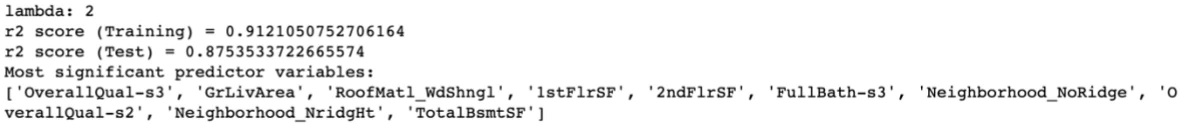
Answer:

- Analysis with ridge regression :
    - The optimal value of alpha is at around “1”. Below is the screenshot from the notebook:



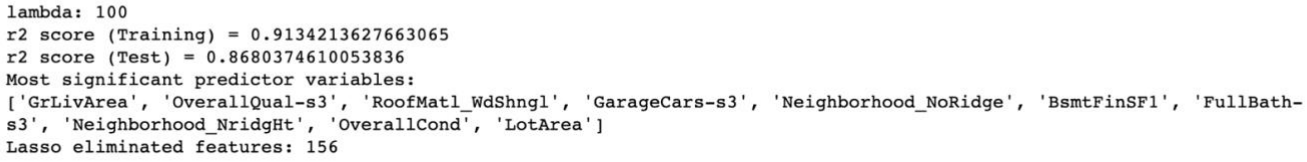
```
lambda: 1
r2 score (Training) = 0.9237515257628448
r2 score (Test) = 0.8755262272630014
Most significant predictor variables:
['OverallQual-s3', 'GrLivArea', '1stFlrSF', 'RoofMatl_WdShngl', 'FullBath-s3', '2ndFlrSF', 'TotalBsmtSF', 'BsmtFinSF
1', 'LotArea', 'Neighborhood_NoRidge']
```

    - When we double the alpha, it has increased the bias as we can see from the r2 score of train and test datasets. We can also observe that the top impactful variables have also changed slightly. Below is the screenshot from the notebook:



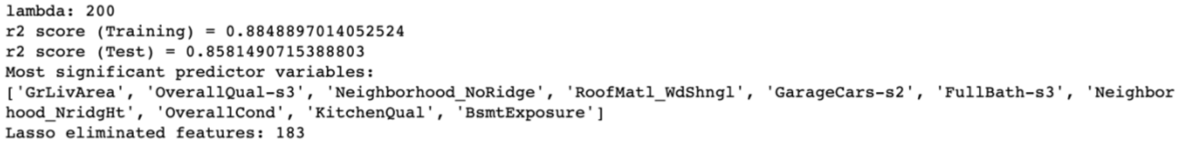
```
lambda: 2
r2 score (Training) = 0.9121050752706164
r2 score (Test) = 0.8753533722665574
Most significant predictor variables:
['OverallQual-s3', 'GrLivArea', 'RoofMatl_WdShngl', '1stFlrSF', '2ndFlrSF', 'FullBath-s3', 'Neighborhood_NoRidge', 'O
verallQual-s2', 'Neighborhood_NridgHt', 'TotalBsmtSF']
```

- Analysis with Lasso regression :
    - The optimal value of alpha is at around “100”. As we can see, the lasso has eliminated 156 features. Below is the screenshot from the notebook:



```
lambda: 100
r2 score (Training) = 0.9134213627663065
r2 score (Test) = 0.8680374610053836
Most significant predictor variables:
['GrLivArea', 'OverallQual-s3', 'RoofMatl_WdShngl', 'GarageCars-s3', 'Neighborhood_NoRidge', 'BsmtFinSF1', 'FullBath-
s3', 'Neighborhood_NridgHt', 'OverallCond', 'LotArea']
Lasso eliminated features: 156
```

    - When we double the alpha, we can see that the algorithm has now eliminated 183 features further simplifying the model. But it has increased the bias as we can see from the r2 score of train and test datasets. We can also observe that the top impactful variables have also changed slightly. Below is the screenshot from the notebook:



```
lambda: 200
r2 score (Training) = 0.8848897014052524
r2 score (Test) = 0.8581490715388803
Most significant predictor variables:
['GrLivArea', 'OverallQual-s3', 'Neighborhood_NoRidge', 'RoofMatl_WdShngl', 'GarageCars-s2', 'FullBath-s3', 'Neighbor
hood_NridgHt', 'OverallCond', 'KitchenQual', 'BsmtExposure']
Lasso eliminated features: 183
```

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?

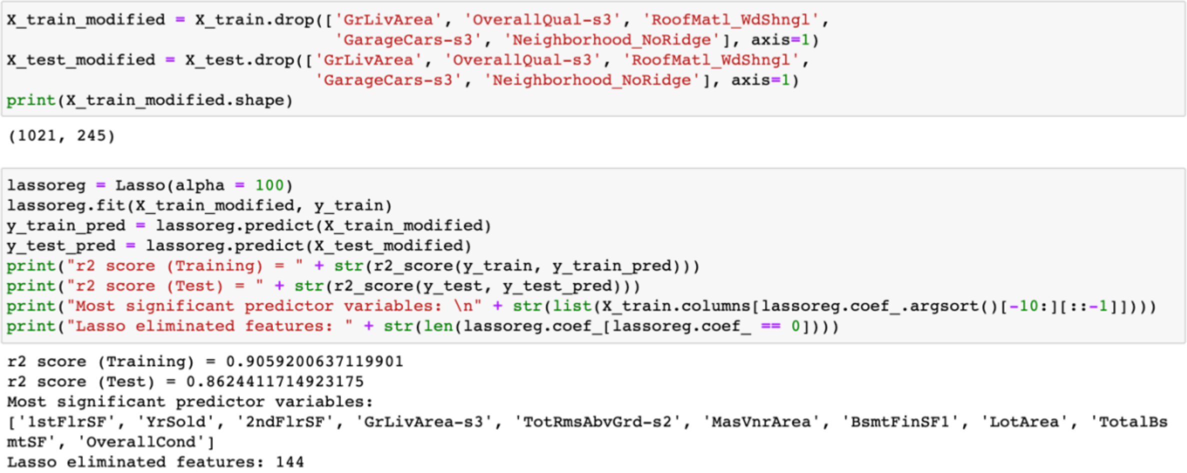
Answer:

As we can see from the screenshots above, the performance of the models from ridge and lasso regression at their corresponding optimal alpha values are almost identical. The r2 scores on train/test datasets are quite identical. Since total number of predictor variables are quite high in the data, so are the number of coefficients in the model. We can also observe that among those high number of predictor variables, many of them have insignificant coefficients compared to others. So it makes sense to go with “Lasso” regression here.

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:

Below is the screenshot of the model built after removing five most important predictor variables:



```
X_train_modified = X_train.drop(['GrLivArea', 'OverallQual-s3', 'RoofMatl_WdShngl',
                                 'GarageCars-s3', 'Neighborhood_NoRidge'], axis=1)
X_test_modified = X_test.drop(['GrLivArea', 'OverallQual-s3', 'RoofMatl_WdShngl',
                               'GarageCars-s3', 'Neighborhood_NoRidge'], axis=1)
print(X_train_modified.shape)
```

```
(1021, 245)
```

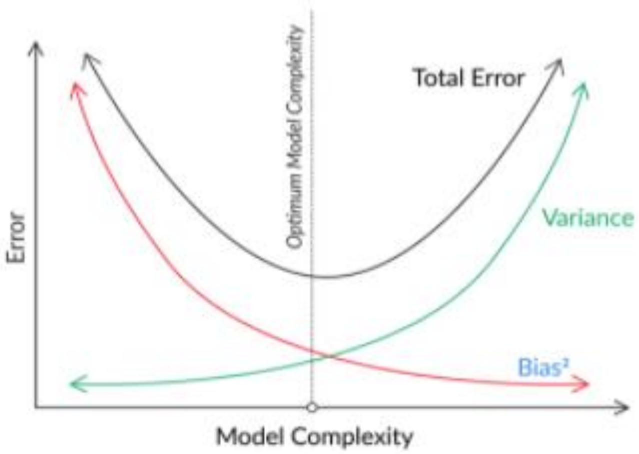
```
lassoreg = Lasso(alpha = 100)
lassoreg.fit(X_train_modified, y_train)
y_train_pred = lassoreg.predict(X_train_modified)
y_test_pred = lassoreg.predict(X_test_modified)
print("r2 score (Training) = " + str(r2_score(y_train, y_train_pred)))
print("r2 score (Test) = " + str(r2_score(y_test, y_test_pred)))
print("Most significant predictor variables: \n" + str(list(X_train.columns[lassoreg.coef_.argsort()[-10:][::-1]])))
print("Lasso eliminated features: " + str(len(lassoreg.coef_[lassoreg.coef_ == 0])))
```

```
r2 score (Training) = 0.9059200637119901
r2 score (Test) = 0.8624411714923175
Most significant predictor variables:
['1stFlrSF', 'YrSold', '2ndFlrSF', 'GrLivArea-s3', 'TotRmsAbvGrd-s2', 'MasVnrArea', 'BsmtFinSF1', 'LotArea', 'TotalBs
mtSF', 'OverallCond']
Lasso eliminated features: 144
```

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 model should perform well on unseen data as well for it to be called robust and generalisable. One way to ensure that is the model should perform equally well on both train and test data sets. As we know that the total error of a model is the sum of “bias” and “variance”, we should find an optimum hyper-parameter to build the model. Below is the screenshot illustrating the same:



As we can see, finding the optimum complexity may involve sacrificing on “bias”. Although it would bring down accuracy on the known data (training data), it would be generalized to perform well on unseen data which is what we finally want to achieve.