Assignment - Advanced Regression

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?

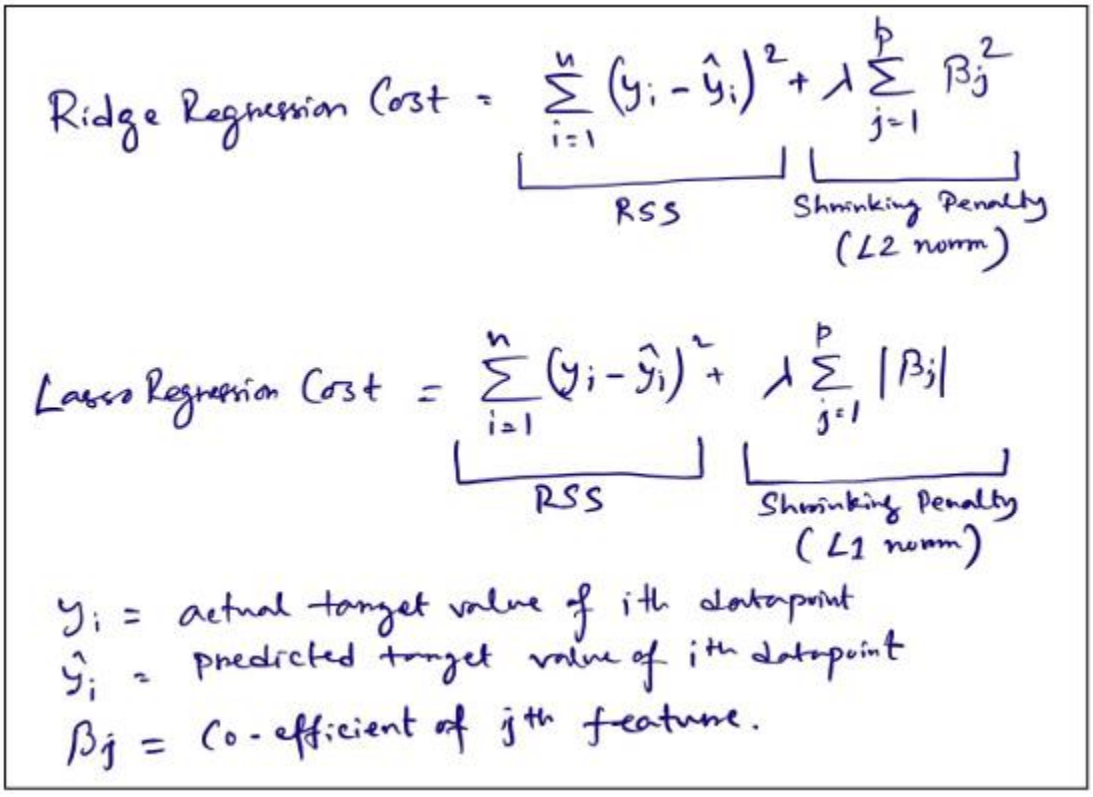
Answer:

Optimal value of alpha:

- Optimal alpha (Lambda) value for Ridge Regression model is: 7
- Optimal alpha (Lambda) value for Lasso Regression model is: 0.0006

Effect of choosing double the value of optimal alpha:

Before explaining the second part of the question, let's see the cost functions of Ridge and Lasso.



So, here it can be seen that in both the cases penalty term increases with higher value of beta co efficient. Ridge imposes more aggressive penalty as it uses sum of square of all beta coefficients (L2

norm) as shrinking penalty. Where Lasso uses sum of absolute values of all beta coefficients (L1 norm) as shrinking penalty. In both equations these norms are multiplied by lambda or alpha. This alpha is a hyperparameter and its optimal value can be obtained by performing cross validation. Value of alpha can be any number >= 0.

If we increase the value of alpha then shrinking penalty will be higher, so Ridge and Lasso both will try to shrink values of beta coefficients towards zero, so our model will be simpler. That means it will increase the bias where variance will be reduced. If we increase the value of alpha to a very large number, then all coefficients of Lasso become 0 and for Ridge coefficients become close to zero (as they cannot be exact 0 in Ridge). That means the model will have very high bias and low variance and it may result in underfitting. That means model will fail to learn the underlying data pattern in training dataset.

If we reduce the value of alpha then shrinking penalty will be lower, so model bias will reduce, and variance will increase. Now if we put value of alpha as 0, then the cost function of both Ridge and Lasso become OLS cost function (i.e., RSS) and we will get exact same model as we get using OLS. So, reducing Value of alpha reduces the effect of shrinking penalty, may lead to possible overfitting; for very low or close to zero values of alpha. So, we need to find the optimal value of alpha by performing hyperparameter tuning.

Top 10 features with beta coefficient values obtained from Ridge after using alpha=14

```
1stFlrSF                0.169180
2ndFlrSF                0.165877
GarageCars              0.165548
TotRmsAbvGrd            0.134917
YearRemodAdd            0.116432
Exterior1st_BrkFace     0.102340
KitchenQual_TA         -0.100293
BsmtQual_TA            -0.098355
FullBath                0.089936
CentralAir              0.086941
dtype: float64
```

Top 10 features with beta coefficient values obtained from Lasso after using alpha= .0012

```
OverallQual              0.355481
GrLivArea                0.351966
GarageCars               0.210659
OverallCond              0.145340
MSSubClass              -0.094918
Neighborhood_StoneBr     0.084574
Exterior1st_BrkFace      0.080393
YearBuilt                0.079261
Neighborhood_Crawfor     0.078374
Neighborhood_NridgHt     0.076751
dtype: float64
```

So, after Doubling value of alpha the most important variable:

In Ridge model: **1stFlrSF** (First Floor square feet)
In Lasso model: **OverallQual** (Rates the overall material and finish of the house)

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:

As per Occam's Razor a model should not be unnecessarily complex.

Model complexity depends upon two main things: No. of features or independent variables and Magnitude of beta coefficients. Normalization (Ridge and Lasso) already shrinks beta coefficients towards zero.

Now, Lasso and Ridge both have similar r2 score and MAE on the test dataset. But Lasso has dropped 110 features and the final no. of features in the Lasso Regression model is 116. Where Ridge has all 226 features. So, the Lasso model is simpler than Ridge with having similar r2 score and MAE.

Ridge:

```
r2 score on testing dataset: 0.8903120151675324
MSE on testing dataset: 0.018566486874924 85
RMSE on testing dataset: 0.13625889649826484
MAE on testing dataset: 0.09617590520158158
```

Lasso:

```
r2 score on testing dataset: 0.894179127084086
MSE on testing dataset: 0.017911914883724383
RMSE on testing dataset: 0.13383540220630857
MAE on testing dataset: 0.09405994331463174
```

As these two models shows almost similar performance on test dataset, we should choose the simpler model. So, I will choose Lasso as my final model

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:
Initially top 5 features in Lasso model are as below:

```
GrLivArea            0.339067
OverallQual          0.311471
GarageCars           0.188410
OverallCond          0.156641
Neighborhood_StoneBr 0.132668
```

As Neighborhood_StoneBr is a dummy variable, dropping entire Neighborhood feature. After dropping GrLivArea, OverallQual, OverallCond, GarageArea, Neighborhood features, rebuilt the Las so model again with rest of the features, now 5 most important predictor variables are as below.

```
1stFlrSF             0.340473
2ndFlrSF             0.303063
GarageCars           0.223494
Exterior1st_BrkFace  0.131593
YearRemodAdd         0.130971
```

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?

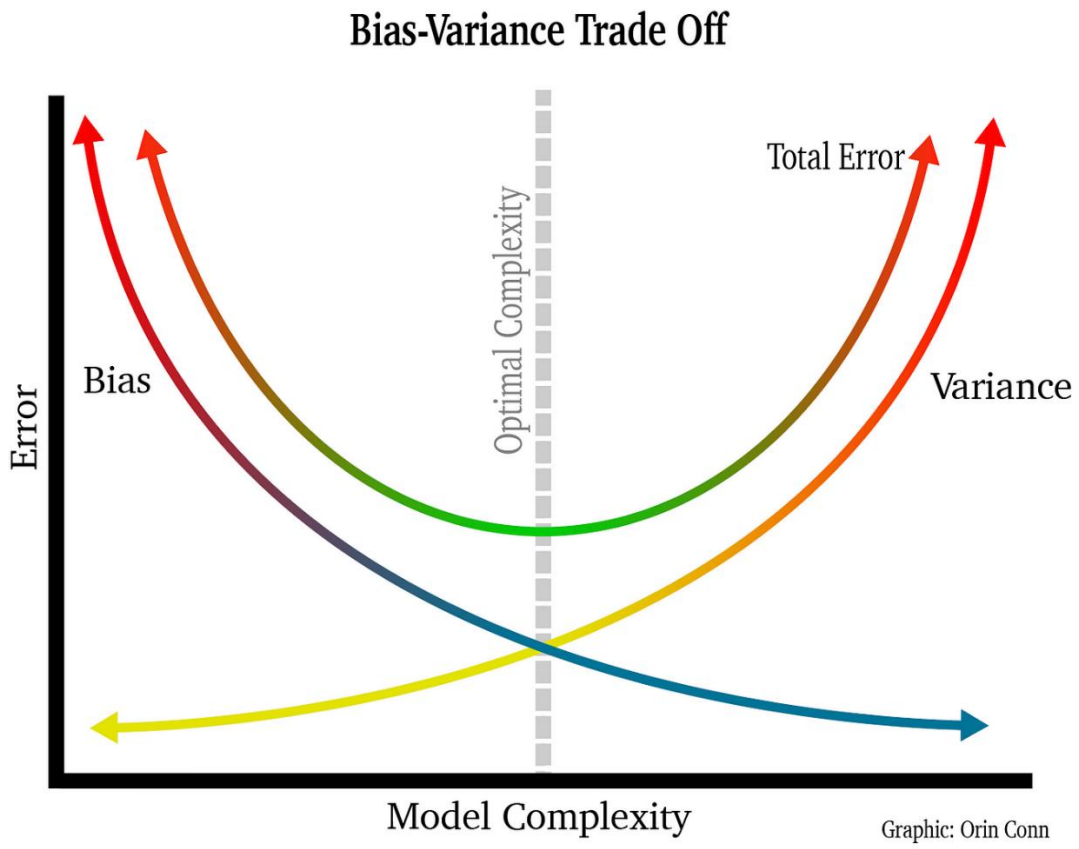
Answer:

A model should be complex enough that it learns the data patterns in the training dataset but not too complex that it also learns the noise in the training dataset. The model should be general enough and not too complex so that it memorizes every datapoint in the training dataset.

An underfitting model usually has high bias and low variance. It fails to understand the data pattern in the training dataset, so it performs badly both on the training and testing dataset. Whereas an overfitting model usually has low bias and high variance. It performs well on the training dataset but performs badly on the testing dataset or unseen data.

A scenario of overfitting can be easily identified by comparing model performance in training and testing datasets. If there is a huge difference in model performance (r2 score, model accuracy, MAE, RMSE, Confusion Matrix, etc. other evaluation metrics) on training and testing datasets, then it's a case of overfitting.

A robust model should have low bias and low variance, and it should not suffer from underfitting and overfitting. It can be achieved by doing a trade-off between bias and variance. One of the ways to remove overfitting to create a robust and generalizable model is to reduce model complexity.



Model complexity depends on two main things: **a number of features or independent variables and magnitude of beta coefficients**. Normalization (Ridge and Lasso) already shrinks beta coefficients towards zero. Again, Lasso also helps in reducing a number of features by shrinking some beta coefficients to an exact 0. Thus it helps to overcome overfitting. The accuracy of a robust and generalizable model should be almost same/closer on training and testing datasets.