Advanced Regression Assignment

Subjective Questions

<u>Question 1</u>

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

Optimal value of Ridge Regression is – 6.0

Optimal value of Lasso Regression is – 100

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

After Doubling the alpha values –

Ridge – 12.0

Lasso – 200

The prediction accuracy changes varies bit (although a negligible difference)

Ridge Training – (0.88 to 0.86)

Ridge Testing – (0.85 to 0.84)

Lasso Training – (0.88 to 0.87)

Lasso Testing – (0.848 to 0.845)

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	R2 Score (Train)	9.069813e-01	8.663972e-01	8.704871e-01
1	R2 Score (Test)	-2.400766e+22	8.416602e-01	8.452306e-01
2	RSS (Train)	5.936317e+11	8.526340e+11	8.265327e+11
3	RSS (Test)	6.783905e+34	4.474249e+11	4.373358e+11
4	MSE (Train)	2.411269e+04	2.889804e+04	2.845229e+04
5	MSE (Test)	1.243104e+16	3.192477e+04	3.156278e+04

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

- OverallQual
- TotalBsmtSF
- 2ndFlrSF
- Neighborhood_NoRidge
- GarageCars

Basically the above predictor variables are same before the change implementation as well, only the coefficients are little bit changed.

Before Doubling Alpha

```
betas['Ridge'].sort_values(ascending=False)[:10]
```

```
OverallQual             67320.691079
2ndFlrSF                64661.044910
Neighborhood_NoRidge    48726.046317
GarageCars              43061.017815
FullBath                41200.840035
```

```
betas['Lasso'].sort_values(ascending=False)[:10]
```

```
TotalBsmtSF             142119.813754
2ndFlrSF                109565.201362
OverallQual             108595.060632
Neighborhood_NoRidge     52311.712593
GarageCars               46627.803282
```

After Doubling Alpha

```
betas['Ridge'].sort_values(ascending=False)[:10]
✓ 0.0s
```

```
OverallQual             54578.985716
2ndFlrSF                48337.593181
Neighborhood_NoRidge    44034.761607
GarageCars              37532.170424
FullBath                37151.863400
```

```
betas['Lasso'].sort_values(ascending=False)[:10]
✓ 0.0s
```

```
OverallQual             118185.963656
TotalBsmtSF             116123.635555
2ndFlrSF                 93730.463562
Neighborhood_NoRidge     53672.553264
GarageCars               47962.160509
```

Note:- Please find the attached jupyter notebook for the code & results.

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?

Below are the Optimal Values, R2, RMSE values for

Ridge Regression

Optimal Value of Lambda – 6

R2 Score (Train) – 0.88

R2 Score (Test) – 0.85

Lasso Regression – 100

R2 Score (Train) – 0.88

R2 Score (Test) – 0.84

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	R2 Score (Train)	9.069813e-01	8.804186e-01	8.893337e-01
1	R2 Score (Test)	-2.400766e+22	8.505200e-01	8.487007e-01
2	RSS (Train)	5.936317e+11	7.631514e+11	7.062565e+11
3	RSS (Test)	6.783905e+34	4.223894e+11	4.275303e+11
4	MSE (Train)	2.411269e+04	2.733962e+04	2.630076e+04
5	MSE (Test)	1.243104e+16	3.101875e+04	3.120694e+04

Since both the regressions, have the R2 score approximately the same, but since Lasso shrinks some of the variable coefficients to 0 and helps in variable selection, those features are automatically excluded and hence the model becomes simple with less number of features. Hence Lasso Regression is preferred.

Note:- Please find the attached jupyter notebook for the code & results.

Question 3

After building the model, you realized that the 5 most important predictor variables in lasso model are not available in the incoming data. You will now have to create another model excluding the 5 most important predictor variables. Which are the 5 most important predictor variables now?

The five most important predictor variables as per Lasso Regression model are –

- TotalBsmtSF
- 2ndFlrSF
- OverallQual
- Neighborhood_NoRidge
- GarageCars

Now after removing the above 5 predictor variables below are the other 5 important predictor variables –

- FullBath
- LotArea
- MasVnrArea
- BedroomAbvGr
- BsmtUnfSF

Note:- Please find the attached jupyter notebook for the code & results.

Question 4

How can you make sure that a model is robust and generalisable?

As per the "Occam's Razor" – the fundamental principle of all the machine learning algorithms are the predictive model has to be as simple as possible, but no simpler.

Some of the parameters to look for complexity of a model is

- Number of parameters required to specify the model completely.
- Degree of the function, if it is a polynomial.
- Size of the best-possible representation of the model.
- Depth or size of a decision tree.

One simple thumb rule as per Occam's Razor is given 2 models that show similar performance, in the finite training or test data, we should pick the one that makes fewer assumptions about the data that is yet to be seen. That means we need to pick the "simpler" of the 2 models. The relationship between the complexity of a model & its usefulness in a learning context is

- Simpler models are usually more "generic".
- Simpler models require fewer training samples.
- Simpler models are more robust.
- Simpler models make more errors in the training set – Complex models lead to overfitting. Reducing the Overfitting makes the model more robust & generalized.

What are the implications of the same for the accuracy of the model and why?

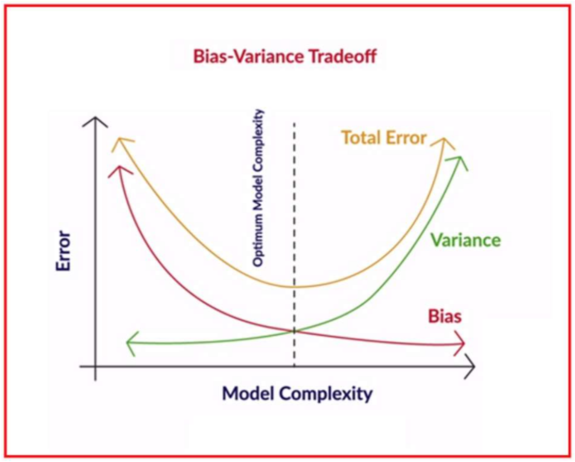
Regularization can be used to make the model simpler. For Regression this involves adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.

Making a model simple leads to the **Bias-Variance Trade Off**.

"**Variance**" of a model is the **variance in its output** on some test data with respect to the changes in the training data. In other words, variance here refers to the **degree of changes in the model itself** with respect to changes in the training data.

"Bias" quantifies how accurate the model is likely to be on future (test) data. Extremely simple models are likely to fail in predicting complex real-world phenomena. Simplicity has its own disadvantages.

Ideally, we want to reduce both bias and variance because the expected total error of a model is the sum of the errors in bias and variance, as shown in the figure given below.



In practice, however, we often cannot have a model with a low bias and a low variance. As the model complexity increases, the bias reduces, whereas the variance increases and, hence, the trade-off.