Problem Statement – Part II

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 for:
- ridge regression: 6.0
- lasso regression: 0.0001

It can also be seen that for Ridge and lasso, on doubling the alpha value, R-Squared goes down.
Root mean squared error has gone up for both ridge and lasso.

Top 5 most important predictor variables using:

- ridge:
  - OverallCond : Overall condition of the house
  - BsmtFullBath: Basement full bathrooms
  - BsmtFinSF2: Type 2 finished square feet
  - 2ndFlrSF: Second floor square feet
  - 1stFlrSF: First Floor square feet

- lasso:
  - BsmtFullBath: Basement full bathrooms
  - OverallCond : Overall condition of the house
  - BsmtFinSF2: Type 2 finished square feet
  - BsmtFinSF1: Type 1 finished square feet
  - OverallQual: Rates the overall material and finish of the house

Note: Relevant code could be found here:
https://github.com/rahul2july/housingpriceprediction/blob/main/Housing_Price_Prediction.ipynb [Under ## Solving Subjective Questions]

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?

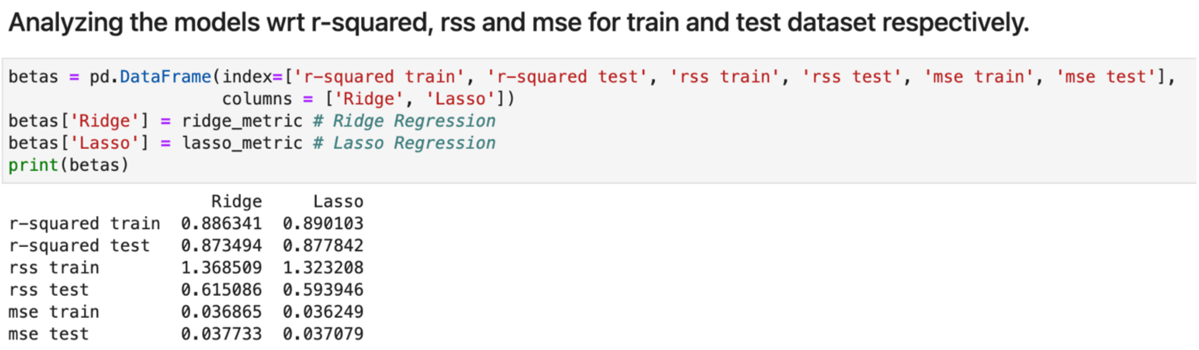
Answer:

Based on the assignment, we could see that :

- Residual Sum of Squares(RSS) is very close for both ridge and lasso regression.
- R-squared is a little better for lasso as compared to ridge.
- Lasso seems to be performing a little better out of the two models wrt. Root mean square error (RMSE)

Since Lasso will penalize more on the dataset and this could also help in feature elimination and making model more robust, we will choose this for the model.

Attaching the results for reference:



Analyzing the models wrt r-squared, rss and mse for train and test dataset respectively.

```
betas = pd.DataFrame(index=['r-squared train', 'r-squared test', 'rss train', 'rss test', 'mse train', 'mse test'],
                     columns = ['Ridge', 'Lasso'])
betas['Ridge'] = ridge_metric # Ridge Regression
betas['Lasso'] = lasso_metric # Lasso Regression
print(betas)
```

```
                    Ridge     Lasso
r-squared train  0.886341  0.890103
r-squared test   0.873494  0.877842
rss train        1.368509  1.323208
rss test         0.615086  0.593946
mse train        0.036865  0.036249
mse test         0.037733  0.037079
```

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:

Code and Solution:

Top 5 features defining "SalePrice" using lasso model[with alpha as 0.0001] currently are:

- BsmtFullBath
- OverallCond
- BsmtFinSF2
- BsmtFinSF1
- OverallQual

Code for removing the five most important predictor variables in lasso model could be found here:
https://github.com/rahul2july/housingpriceprediction/blob/main/Housing_Price_Prediction.ipynb [Under ## Solving Subjective Questions]

So the five most important features now:

- GrLivArea: Above grade (ground) living area square feet
- TotalBsmtSF: Total square feet of basement area
- GarageArea: Size of garage in square feet
- MasVnrArea: Masonry veneer area in square feet
- Neighborhood_NridgHt: Physical locations within Ames city limits- Northridge Heights

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:
In summary:

- a model is robust when any variation in the data does not affect its performance much.
- a generalizable model is able to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model.

The way to ensure model is robust and generalized is :

- to check the test score and ensure training and test set are almost similar.
- This eventually means that training set can reduce a bit to avoid overfitting.
- So the model should not be too complex in order to be robust and generalizable.
- Also outlier analysis and correlation analysis need to be done and only relevant attributes should be retained in the final dataset.

If model is not robust, it cannot be trusted for predictive analysis.
If we look at the from the perspective of accuracy, a too complex model will have a very high accuracy.

So, to make our model more robust and generalizable:

- we will have to decrease variance which will lead to some bias and this will a cause some decrease in accuracy.
- In general, we have to find balance between model accuracy and complexity.

This could be achieved by Regularization techniques like Ridge Regression and Lasso Regression.