Advanced Regression – Subjective Questions and Answers

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?

Solution:

- The optimal value of alpha for ridge regression has turned out to be 2.0 and for lasso regression it is 100.
- When we double the alpha values, we observe the following:
  - The R2_score has **come down** by more than 4%.
  - The error metrics like Residual Sum of Squares (RSS) and Root Mean Square Error (RMSE) have **increased**.
- The top 5 variables after applying the change are OverallQual, GrLivArea, TotalBsmtSF, 1stFlrSF, BsmtFinSF1.
- Code screenshot 1:

```
In [268]: alpha = 4.0 # Doubling the alpha from 2.0 to 4.0 for ridge regression:
          ridge_final_doubled = Ridge(alpha=alpha)
          ridge_final_doubled.fit(X_train_by_rfe, y_train)
          ridge_final_doubled.coef_

Out[268]: array([ 44322.53532262, 101028.89832321,  26880.58691125,  46014.2364581 ,
                  55559.55283092,  73925.34373911,  73115.2530656 ,  36567.05099791,
                  78310.55811018, -29227.01070236,  50631.2834665 , -12680.38351207,
                 -36459.77130134, -42146.47485528, -59435.28171894,  -9021.25280196,
                 -17737.51151858,  13802.69348953, -12454.21062108,   5226.75088049])

In [269]: print_model_evaluation_metrics(ridge_final_doubled, X_train_by_rfe, X_test_by_rfe)

          R2 Train: 0.8836755400433906
          R2 Test: 0.8567795835129987
          RSS Train: 569885863285.9297
          RSS Test: 378321289454.4883
          RMSE Train: 638885496.9573203
          RMSE Test: 859821112.3965642

Out[269]: [0.8836755400433906,
           0.8567795835129987,
           569885863285.9297,
           378321289454.4883,
           638885496.9573203,
           859821112.3965642]

In [270]: betas_new = pd.DataFrame(index=X_train_by_rfe.columns)
          betas_new.rows = X_train_by_rfe.columns
          betas_new['Ridge'] = ridge_final_doubled.coef_

In [271]: pd.set_option('display.max_rows', None)
          betas_new['Ridge'].sort_values(ascending=False)

Out[271]: OverallQual       101028.898323
          GrLivArea          78310.558110
          TotalBsmtSF        73925.343739
          1stFlrSF           73115.253066
          BsmtFinSF1         55559.552831
```

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?

Solution:

- I will choose lasso regression with alpha 100.
    - This is because lasso regression is giving us the maximum R2 score on test data set (86.37%).
    - Moreover, the error metrics like RSS and RMSE for the test dataset are the least for lasso regression.

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	R2 Score (Train)	8.493909e-01	8.880338e-01	8.861557e-01
1	R2 Score (Test)	8.125920e-01	8.605621e-01	8.637263e-01
2	RSS (Train)	7.378500e+11	5.485345e+11	5.577352e+11
3	RSS (Test)	4.950442e+11	3.683297e+11	3.599713e+11
4	MSE (Train)	8.271861e+08	6.149490e+08	6.252637e+08
5	MSE (Test)	1.125100e+09	8.371129e+08	8.181165e+08

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?

Solution:

- The top 5 features after creating a new lasso regression model by dropping the earlier top 5 features are - 1stFlrSF, 2ndFlrSF, YearBuilt, OverallCond, TotRmsAbvGrd.
- Code screenshot:

```
In [287]: # alpha 100
          alpha = 100
          lasso_dropped = Lasso(alpha=alpha)
          lasso_dropped.fit(X_train_lasso_dropped, y_train)

Out[287]: Lasso(alpha=100)
```

```
In [288]: print_model_evaluation_metrics(lasso_dropped, X_train_lasso_dropped, X_test_lasso_dropped)

          R2 Train: 0.83953125990279
          R2 Test: 0.8216025244871297
          RSS Train: 786153372341.6003
          RSS Test: 471242610704.0409
          RMSE Train: 881337861.3695071
          RMSE Test: 1071005933.4182748

Out[288]: [0.83953125990279,
           0.8216025244871297,
           786153372341.6003,
           471242610704.0409,
           881337861.3695071,
           1071005933.4182748]
```

```
In [289]: #important predictor variables
          betas = pd.DataFrame(index=X_train_lasso_dropped.columns)
          betas.rows = X_train_lasso_dropped.columns
          betas['Lasso_dropped'] = lasso_dropped.coef_
          pd.set_option('display.max_rows', None)
          betas['Lasso_dropped'].sort_values(ascending=False)

Out[289]: 1stFlrSF              293847.544371
          2ndFlrSF              107842.949920
          YearBuilt              88363.262521
          OverallCond            44771.522703
          TotRmsAbvGrd           41295.446761
          GarageType_BuiltIn      5305.974912
          Heating_OthW              -0.000000
          SaleType_Oth               0.000000
          SaleType_CWD           -9142.834965
          Exterior1st_Stone     -18581.214777
          Functional_Sev        -22258.998332
```

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?

Solution:

- One of the most important ways to generalise a model is by controlling(reducing) its complexity. Regularisation is used to control the model complexity thereby generalising the model.
- With regularisation, we have an opportunity to control the model complexity by tuning the hyper-parameter lambda. A complex model (non-generalisable) model shows high variance and low bias. By steadily increasing the value of lambda, we significantly bring down the variance of the model at the cost of slightly increased bias. A model with optimal value of variance and bias will have the lowest total error. This makes the overall model reach optimal generalization levels, thereby increasing the robustness of the model.
- In the above process, the accuracy of the model might come down since the bias(deviation) of the model is increased due to increase in value of lambda. Here, we are consciously trying increase the generalisability of the model by trading-off an acceptable level of accuracy of the model.