Assignment – Advance 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

The optimal value of Alpha for both Ridge and Lasso is **500**

Here is a table that shows how the different params changes based on Alpha of 500 and 1000

Metric	Ridge Regression(Alpha=500)	Ridge Regression(Alpha=1000)	Lasso Regression(Alpha=500)	Lasso Regression(Alpha=1000)
R2 Score (Train)	0.8936	0.8707	0.9325	0.8990
R2 Score (Test)	0.8618	0.8505	0.8460	0.8410
RSS (Train)	678841300000.00	825197600000.00	430563600000.00	644591700000.00
RSS (Test)	389581300000.00	421496900000.00	434143900000.00	448139400000.00
MSE (Train)	25785.25	28429.30	20535.52	25126.36
MSE (Test)	29823.73	31021.31	31483.27	31986.70

Table 1: Metrices for Ridge and Lasso

We see that in **Ridge Regression** the R2 Score changes for **Train** dataset from **89% to 87%** and for **Test** the value changes from **86% to 85%**. Though it looks like the model is a better fit at Alpha of 1000, though the accuracy reduced a bit. We also see that the mean sum of square (The price difference between actual and predicted) is also much better. For Alpha of 500 the difference is (29824-25785=) 4039 vs (31021-28429=) 2592 in Alpha 1000. So, we see that in Alpha 1000 the model is a better fit.

Similarly in **Lasso Regression** the R2 Score changes in Train dataset from **93%** to **89%** and but for Test dataset the value remains at around **84% (84.60% vs 84.10%)**.

Let's have a look at **top 5** and **bottom 5** parameters for **Alpha of 500**

Param Name	Ridge	Lasso
GrLivArea	7643.665667	30191.30962
OverallQual_10	7120.076658	12568.16545
OverallQual_9	6895.192201	12375.30905
Neighborhood_NoRidge	5578.578832	5033.186571
1stFlrSF	5224.662931	0
KitchenQual_TA	-2901.40524	-3006.54193
Neighborhood_Edwards	-2994.46287	-2150.64747
OverallQual_6	-3027.44803	0
BsmtQual_Gd	-3479.45636	-5351.9827
Condition2_PosN	-6206.01105	-13973.0691

Let's have a look at **similar params** for **Alpha of 1000**

Parameter	Ridge	Lasso
GrLivArea	6334.415528	28457.98482
OverallQual_10	5734.984339	12400.37579
OverallQual_9	5520.593989	13729.42741
Neighborhood_NoRidge	4774.068542	5346.859707
1stFlrSF	4710.580698	0
BsmtExposure_No	-2431.719741	-1807.742287
OverallQual_6	-2478.371752	0
KitchenQual_TA	-2563.965148	-740.313869
ExterQual_TA	-2682.973418	-2516.832228
Condition2_PosN	-4121.4652	-11844.5192

If you look at both the tables, you can see that the top 5 parameters remain same for both Alpha 500 and Alpha 1000. **But the bottom 5 parameters changes**.

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

If we check the table below

Metric	Ridge Regression	Lasso Regression
R2 Score (Train)	0.8936	0.9325
R2 Score (Test)	0.8618	0.8460
RSS (Train)	678841300000.00	430563600000.00
RSS (Test)	389581300000.00	434143900000.00
MSE (Train)	25785.25	20535.52
MSE (Test)	29823.73	31483.27

We can see that though Ridge has a lower R2 Score for Train set then Lasso (**89%** vs **93 %**), in train set the value changes too much (86% with **3% change for Ridge** vs 84% with **9% change** for Lasso). I would prefer the model with Ridge regression for my final model.

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?

The top 5 parameters are at present -

1. GrLivArea : Above grade (ground) living area square feet
2. OverallQual_9 : Overall quality is 9
3. OverallQual_10 **:** Overall quality is 10
4. OverallQual_8 **:** Overall quality is 8
5. GarageCars **:** Size of garage in car capacity

After dropping those values the **new most important top 5 parameters** are –

1. **RoofMatl_CompShg** : Standard (Composite) Shingle Roof material
2. **2ndFlrSF** : Second floor square feet
3. **RoofMatl_Tar** : Gravel & Tar Roof material
4. **RoofMatl_WdShngl** : Wood Shingles Roof material
5. **1stFlrSF** : First Floor square feet

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

To ensure that the machine learning model is robust we can do the following things –

- **Cross–Validation**: we can use techniques like K-fold cross validation that assesses the performance of the model on different subsets of the data. This is especially good for smaller datasets.
- **Train-Test split:** In this process the data is split into train and test sets and then the model is trained on the train set and validated in the test set to make sure the model is performing well on unseen data.
- **Validation Set**: We can also have a validation set (apart from train and test set). That can be used for doing hyperparameter tuning and making decision on various model architecture
- **Regularization Techniques:** Regularization penalizes overly complex models (too many parameters) and thus prevents the data from overfitting and encourages simpler model (lesser number of parameters) that generalizes the model better.
- **Feature Engineering:** This includes pre-processing the data and identifying categorical and continuous parameter. scaling, dropping unnecessary parameters, adding dummy variables for categorical data etc.
- **Data Augmentation:** We can also add data in our existing dataset from data available online. For example if we are building a model about cycle rentals we can augment the data with weather to understand how weather patterns effects demand.

Implication on Model Accuracy

Ensuring robustness and generalizability may sometimes come at the cost of sacrificing accuracy on the training data. For example -

- Regularization techniques may prevent the model from fitting the training data too closely, leading to slightly lower accuracy on the training set.
- Cross-validation and train-test splits provide a more realistic estimate of the model's performance but may result in slightly lower accuracy compared to evaluating solely on the training data.
- Feature engineering and data augmentation may introduce noise or distortions that reduce accuracy on the training set but improve the model's ability to generalize to unseen data.

However, prioritizing robustness and generalizability over maximizing accuracy on the training data is crucial for ensuring that the model performs well in real-world scenarios where the data may differ from the training set. A model that generalizes well will typically perform better on new, unseen data, which is the final goal of most machine learning applications.