Student: SM Arun Kumar

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** is **4** and **lasso** is **0.001**.

In Lasso regularization, double the alpha value will be 0.002.

```
########################################################################
Lasso Train Model r2 value post doubling alpha: 0.8884414964885949
Lasso Test Set r2 value post doubling alpha: 0.8393946241843337
------------------------------------------------------------------------
Lasso Train Set r2 value pre doubling alpha: 0.8939960156755192
Lasso Test Set r2 value pre doubling alpha: 0.8393946241843337
########################################################################
```

By looking at lasso results, we don't see significant changes in the model. Train set R square has decreased slightly. Most important predictor variable is Neighborhood : NoRidge with co-efficient 0.444801.

In Ridge regularization, double the value of alpha will be 8.

```
########################################################################
Ridge Train Model r2 post doubling alpha: 0.8929831746760066
Ridge Test dataset r2 post doubling alpha: 0.8386526122908717
------------------------------------------------------------------------
Ridge Train Set r2 value pre doubling alpha: 0.8967672361860246
Ridge Test Set r2 value pre doubling alpha: 0.8391435866954426
########################################################################
```

By looking at Ridge results, we don't see significant changes in the model. Train set R square has decreased slightly. Most important predictor variable is Neighborhood: NoRidge with co-efficient 0.392869.

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

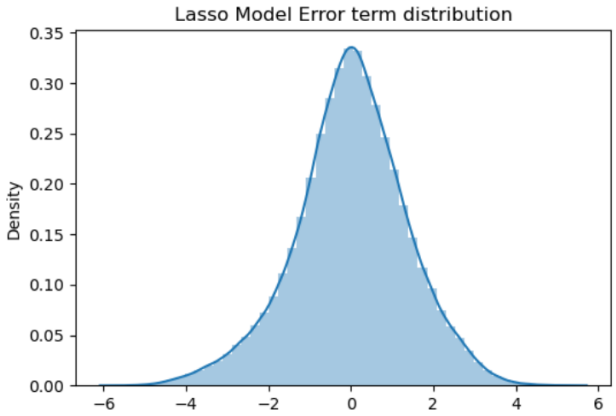
	Metric	RFERegression	Ridge Regression	Lasso Regression
0	HyperParameter	1.150000e+02	{'alpha': 4.0}	{'alpha': 0.001}
1	R2 Score (Train)	9.001997e-01	0.896767	0.893996
2	R2 Score (Test)	-2.903453e+16	0.839144	0.839395
3	RSS (Train)	1.008981e+02	104.368324	1917825.37617
4	RSS (Test)	1.247702e+19	69.124883	354570.671243
5	MSE (Train)	9.980033e-02	0.103233	0.106004
6	MSE (Test)	2.874889e+16	0.159274	0.159025

Tain and Test score of Ridge and Lasso model seems to be more or less same. Further, Lasso model has penalized 46 features to ZERO. Only 69 features are used to build the model. This makes the model simpler. So, choosing Lasso Regularization model.

```
#Lasso Coefficients
MainLassoCoeff = pd.Series(lasso.coef_,index=X_train_rfe.columns)
abs(MainLassoCoeff[MainLassoCoeff !=0]).sort_values(ascending=False)
```

```
Neighborhood_NoRidge    0.508991
LotShape_IR3            0.468144
Neighborhood_NridgHt    0.425812
Neighborhood_StoneBr    0.346527
MSSubClass_120          0.298645
                            ...
SaleCondition_Alloca    0.013237
Condition1_RRAn         0.011538
RoofStyle_Gable         0.011134
GarageType_Basment      0.010546
Neighborhood_SawyerW    0.006244
Length: 69, dtype: float64
```

Error term distribution is normal.



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?

Following are the top features from the original Lasso Model.

```
NoZeroMainLassoCoeff = MainLassoCoeff[MainLassoCoeff !=0]
abs(NoZeroMainLassoCoeff).sort_values(ascending=False)[0:20]
```

```
Neighborhood_NoRidge     0.508991
LotShape_IR3             0.468144
Neighborhood_NridgHt     0.425812
Neighborhood_StoneBr     0.346527
MSSubClass_120           0.298645
SaleCondition_Partial    0.297312
Neighborhood_Crawfor     0.281472
2ndFlrSF                 0.273122
```

Following top 5 features are excluded from the original dataset:

Neighborhood, LotShape, MSSubClass, SaleCondition and 2ndFlrSF

New model is rebuilt with lasso regularization and following features seems to be top predictor features:

```
TestMainLassoCoeff = pd.Series(aslasso.coef_,index=aslasso.feature_names_in_)
abs(TestMainLassoCoeff[TestMainLassoCoeff !=0]).sort_values(ascending=False)[0:5]
```

```
BldgType_Twnhs         0.365036
LandContour_HLS        0.291586
Exterior1st_BrkFace    0.253861
GarageType_BuiltIn     0.251272
HouseStyle_SLvl        0.244918
dtype: float64
```

```
len(TestMainLassoCoeff[TestMainLassoCoeff !=0])
```

```
77
```

The newly built model is with length 77.

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?

Following steps can be followed to ensure model is robust and generalisable:

1. **Train and Test data split**: Spilt the given data set into train and test set. Build model on the train set and run the predictive model on the unseen test set. The result obtained can be measured with different metrics as per the Model built.

   Metric example:
   Linear Regression: R Square, Adjusted R-Square, RMSE
   Logistic Regression: Accuracy, Sensitivity and Specificity.

   ROC curve can be plotted to get the optimal point to check the accuracy of Logistic models.

2. Variance Inflation Factor: This is used to check the multicollinearity between the features.
3. Regularization techniques: Lasso and Ridge. Both the techniques penalizes the features to ZERO. This is mainly used to avoid the model overfit.

4. Cross Validation: When the data is small OR to check the consistency of the mode, train data set is divided into multiple smaller subsets. Techniques like k-fold cross validation is used.
5. Hyperparameter tuning: Algorithms like RFE, Ridge, Lasso, Decision Trees have hyperparameters. This is tuned to avoid overfitting.

6. Domain Knowledge: This is very essential to understand the data set and remove the unwanted featured.

Implications for Accuracy:

A robust and generalizable models have consistent across training data and unseen test data.

Ensuring robustness involves balancing bias and variance, leading to a model neither overfit not underfit. Overall objective is to have model with high accuracy and resilient to variations in input data.