Advanced Regression Assignment Subjective Questions(Part 2)- Sudeep K S

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?

Ridge(alpha=3) Regression
R2 score (train) : 0.9393584980659788
R2 score (test) : 0.9170387823080166

Lasso(alpha=50) Regression
R2 score (train) : 0.9369069139272894
R2 score (test) : 0.918395595610946

Note: ALL QUESTIONS ALSO ANSWERED ON JUPYTER NOTEBOOK ATTACHED IN GITHUB

```
#Doubling the Alpha for ridge
alpha=6
ridge=Ridge(alpha=alpha)
ridge.fit(X_train,y_train)
```

```
Ridge(alpha=6)
y_train_pred = ridge.predict(X_train)
print(ridge," Regression")
print("===================================")
print('R2 score (train) : ',metrics.r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred = ridge.predict(X_test)
print('R2 score (test) : ',metrics.r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
Ridge(alpha=6)  Regression
===================================
R2 score (train) :  0.9361704060931698
R2 score (test) :  0.9157545719608025
#Doubling the Alpha for lasso
alpha=100
lasso=Lasso(alpha=alpha)
lasso.fit(X_train,y_train)

Y_train_lasso = y_train.copy()
model_lasso=lasso.fit(X_train, y_train)

print(lasso," Regression")
print("===================================")
y_train_pred = lasso.predict(X_train)
print('R2 score (train) : ',model_lasso.score(X_train, y_train))
print('R2 score (test) : ',model_lasso.score(X_test,y_test))
```

```
Lasso(alpha=100)  Regression
===================================
R2 score (train) :  0.9319360472310817
R2 score (test) :  0.9178871689477476
```

Observations after doubling the value of Alpha:
R2 score reduced for the training and test data when doubling the alpha

The top 10 predictor variables are:

1. **OverallQual**
2. **OverallCond**
3. **Neighborhood_StoneBr**
4. **Neighborhood_NoRidge**
5. **ExterQual_Gd**
6. **ExterQual_TA**
7. **LotArea**
8. **KitchenQual_Gd**
9. **KitchenQual_TA**
10. **YearBuilt**

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?

Based on the R2scores we can choose Lasso, As the R2score is slightly high.

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?

Creating new model after dropping top 5 predictor variables:

```
X_train_new=X_train.drop(['OverallQual','OverallCond','Neighborhood_StoneBr','Neighborhood_NoRidge','ExterQual_Gd'],axis=1)
X_test_new = X_test.drop(['OverallQual','OverallCond','Neighborhood_StoneBr','Neighborhood_NoRidge','ExterQual_Gd'],axis=1)
```

```
# Building Lasso Model with the new dataset
lasso_new = Lasso(alpha=50)
lasso_new.fit(X_train_new,y_train)
lasso_new_coef = lasso_new.coef_
y_test_pred_new = lasso_new.predict(X_test_new)
print('The R2 Score of the model on the test dataset is',r2_score(y_test, y_test_pred))
print('The MSE of the model on the test dataset is', mean_squared_error(y_test, y_test_pred))
lasso_new_coeff = pd.DataFrame(np.atleast_2d(lasso_new_coef),columns=X_train_new.columns)
lasso_new_coeff = lasso_new_coeff.T
lasso_new_coeff.rename(columns={0: 'Lasso Co-Efficient'},inplace=True)
lasso_new_coeff.sort_values(by=['Lasso Co-Efficient'], ascending=False,inplace=True)
print('The most important predictor variables are as follows:')
lasso_new_coeff.head(5)
```

```
The R2 Score of the model on the test dataset is 0.9157545719608025
The MSE of the model on the test dataset is 515161820.4258684
The most important predictor variables are as follows:
```

	Lasso Co-Efficient
MasVnrArea	23395.503266
Functional_Typ	21772.247027
SaleCondition_Partial	20746.762269
BsmtExposure_Gd	19793.335927
LotArea	14576.668865

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?

The most common rules to make a model robust and generalisable are:

1. Outlier Analysis is a must and we need to ensure that retain only those values which are relevant to the dataset
2. The model should be as simple as possible the more complex the model the more probability that it will overfit

If the above is not performed correctly there is a possiblity that the accuracy will decrease on the test set due to overfitting or underfitting.

The above can analyzed using the Bias-Variance tradeoff:
Bias is a phenomenon that skews the result of an algorithm in favor or against an idea.
Variance refers to the changes in the model when using different portions of the training data set.
Bias and variance are inversely related, we must find a fine line between this to make our model robust and generalisable