Problem Statement - Part II

Q1. **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?**

Ans:
In the case of ridge regression:- When we plot the curve between negative mean absolute error and alpha we see that as the value of alpha increase from 0 the error term decrease and the train error is showing increasing trend when value of alpha increases .when the value of alpha is 2 the test error is minimum so we decided to go with value of alpha equal to 2 for our ridge regression.

For lasso regression I have decided to keep very small value that is 0.01, when we increase the value of alpha the model try to penalize more and try to make most of the coefficient value zero. Initially it came as 0.4 in negative mean absolute error and alpha.

When we double the value of alpha for our ridge regression no we will take the value of alpha equal to 10 the model will apply more penalty on the curve and try to make the model more generalized that is making model more simpler and no thinking to fit every data of the data set .from the graph we can see that when alpha is 10 we get more error for both test and train.
Similarly when we increase the value of alpha for lasso we try to penalize more our model and more coefficient of the variable will reduced to zero, when we increase the value of our r2 square also decreases.
The most important variable after the changes has been implemented for ridge regression are as follows:-

1. MSZoning_FV
2. MSZoning_RL
3. Neighborhood_Crawfor
4. MSZoning_RH
5. MSZoning_RM
6. SaleCondition_Partial
7. Neighborhood_StoneBr
8. GrLivArea
9. SaleCondition_Normal

The optimal value of alpha for ridge and lasso regression

Ridge Alpha 1

lasso Alpha 10

Ridge Regression
#Change the alpha value from 1 to 2
alpha = 3
ridge2 = Ridge(alpha=alpha)
ridge2.fit(X_train1, y_train)

r2_train_lr = r2_score(y_train, y_pred_train)

```
print(r2_train_lr)
metric2.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
metric2.append(r2_test_lr)

rss1_lr = np.sum(np.square(y_train - y_pred_train))
print(rss1_lr)
metric2.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test - y_pred_test))
print(rss2_lr)
metric2.append(rss2_lr)

mse_train_lr = mean_squared_error(y_train, y_pred_train)
print(mse_train_lr)
metric2.append(mse_train_lr**0.5)

mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)
metric2.append(mse_test_lr**0.5)
#Alpha 1
#R2score(train) 0.884340040460635
#R2score(test)  0.869613280468847
```

Output :

```
0.8797315810932455
0.87102821482729
607995142958.1414
320928407278.4619
680845624.8131483
729382743.8146862
```

Lasso

```
#Changed alpha 10 to 20
alpha =20
lasso20 = Lasso(alpha=alpha)
lasso20.fit(X_train1, y_train)
```

Output : Lasso(alpha=20)

```
# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = lasso20.predict(X_train1)
y_pred_test = lasso20.predict(X_test1)

metric3 = []
r2_train_lr = r2_score(y_train, y_pred_train)
print(r2_train_lr)
metric3.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
metric3.append(r2_test_lr)
```

```
rss1_lr = np.sum(np.square(y_train - y_pred_train))
print(rss1_lr)
metric3.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test - y_pred_test))
print(rss2_lr)
metric3.append(rss2_lr)

mse_train_lr = mean_squared_error(y_train, y_pred_train)
print(mse_train_lr)
metric3.append(mse_train_lr**0.5)

mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)
metric3.append(mse_test_lr**0.5)

#R2score at alpha-10
#0.8859222400899005
#0.8646666084570094
```

Output :

```
0.8854019698644757
0.8670105918827554
579329522648.7457
330925704989.0289
648745266.1240153
752103874.9750656
```

R2score of training data has decrease and it has increase on testing data

```
#important predictor variables
betas = pd.DataFrame(index=X_train1.columns)
betas.rows = X_train1.columns
betas['Ridge2'] = ridge2.coef_
betas['Ridge'] = ridge.coef_
betas['Lasso'] = lasso.coef_
betas['Lasso20'] = lasso20.coef_
pd.set_option('display.max_rows', None)
betas.head(68)
```

- LotArea---------------Lot size in square feet
- OverallQual---------Rates the overall material and finish of the house
- OverallCond--------Rates the overall condition of the house
- YearBuilt-------------Original construction date
- BsmtFinSF1--------Type 1 finished square feet
- TotalBsmtSF------- Total square feet of basement area
- GrLivArea-----------Above grade (ground) living area square feet
- TotRmsAbvGrd----Total rooms above grade (does not include bathrooms)
- Street_Pave--------Pave road access to property
- RoofMatl_Metal----Roof material_Metal

Predictors are same but the coefficent of these predictor has changed

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?

Ans :

The r2_score of lasso is slightly higher than lasso for the test dataset so we will choose lasso regression to solve this problem

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?

Ans :
X_train1

Those 5 most important predictor variables that will be excluded are :-

1. GrLivArea
2. OverallQual
3. OverallCond
4. TotalBsmtSF
5. GarageArea

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?

Ans :

The model should be generalized so that the test accuracy is not lesser than the training score. The model should be accurate for datasets other than the ones which were used during training. Too much importance should not given to the outliers so that the accuracy predicted by the model is high. To ensure that this is not the case, the outliers analysis needs to be done and only those which are relevant to the dataset need to be retained. Those outliers which it does not make sense to keep must be removed from the dataset. If the model is not robust, It cannot be trusted for predictive analysis.