Name:Mehta Akshat Rohitkumar
Roll No:4609206
Batch:ML-C45

Advanced Regression Assignment

(Subjective Questions)

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 1

Optimal Values of Alpha

- ✓ Ridge Regression Model-**1.0**
- ✓ Lasso Regression Model-**0.0001**

<u>When Double value of Alpha of Ridge Regression-</u>

Feature Coeffients and R2 Value is Same for Both Alpha Values(From Below 2 Pics).

```
]: #selecting the top 10 variables with Alpha-1.0
ridge_coef.sort_values(by='Mod',ascending=False).head(10)
```

]:

	Features	Coefficient	Mod
0	LotFrontage	10.274098	10.274098
55	MSSubClass_85	-0.433124	0.433124
3	OverallCond	0.403853	0.403853
16	HeatingQC	0.308760	0.308760
13	BsmtFinSF2	0.281259	0.281259
10	BsmtFinType1	0.274331	0.274331
14	BsmtUnfSF	0.264105	0.264105
7	BsmtQual	0.262287	0.262287
33	GarageFinish	0.242151	0.242151
2	OverallQual	0.232591	0.232591

```
]: # Prediction using ridge regression
y_train_ridge_pred = ridge.predict(X_train)
print("Ridge regression train r2: ",round(metrics.r2_score(y_true=y_train, y_pred=y_train_ridge_pred),4))
y_test_ridge_pred = ridge.predict(X_test)
print("Ridge regression test r2: ",round(metrics.r2_score(y_true=y_test, y_pred=y_test_ridge_pred),4))

Ridge regression train r2:  0.9253
Ridge regression test r2:  0.7834
```

```
2]: #selecting the top 10 variables with Alpha-2.0
ridge_coef.sort_values(by='Mod',ascending=False).head(10)
```

2]:

	Features	Coefficient	Mod
0	LotFrontage	10.274098	10.274098
55	MSSubClass_85	-0.433124	0.433124
3	OverallCond	0.403853	0.403853
16	HeatingQC	0.308760	0.308760
13	BsmtFinSF2	0.281259	0.281259
10	BsmtFinType1	0.274331	0.274331
14	BsmtUnfSF	0.264105	0.264105
7	BsmtQual	0.262287	0.262287
33	GarageFinish	0.242151	0.242151
2	OverallQual	0.232591	0.232591

```
5]: y_train_pred = ridge_modified.predict(X_train)
y_test_pred = ridge_modified.predict(X_test)

print("Ridge Regression train r2:",r2_score(y_true=y_train,y_pred=y_train_pred))
print("Ridge Regression test r2:",r2_score(y_true=y_test,y_pred=y_test_pred))

Ridge Regression train r2: 0.9216363030938198
Ridge Regression test r2: 0.7881312243876577
```

When Double value of Alpha of Lasso Regression-

Feature Coeffients and R2 Value is Same for Both Alpha Values(From Below 2 Pics).

[725]:
```
# After performing grid search we found the same alpha that ue use before with-lasso-0.0001
lasso = Lasso(alpha=0.0001)
lasso.fit(X_train,y_train)

y_train_pred = lasso.predict(X_train)
y_test_pred = lasso.predict(X_test)

print("Lasso Regression train r2:",r2_score(y_true=y_train,y_pred=y_train_pred))
print("Lasso Regression test r2:",r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
Lasso Regression train r2: 0.9270055122443596
Lasso Regression test r2: 0.7799611994514969
```

[727]:
```
lasso_coef
```

[727]:

	Feature	Coef	mod
0	LotFrontage	10.249130	10.249130
1	LotArea	0.104492	0.104492
2	OverallQual	0.328184	0.328184
3	OverallCond	0.439960	0.439960
4	MasVnrArea	0.221770	0.221770
...	...	...	...
85	SaleType_Con	0.000000	0.000000
86	SaleType_Oth	0.000000	0.000000
87	SaleCondition_Alloca	0.095036	0.095036
88	SaleCondition_Normal	-0.052354	0.052354
89	SaleCondition_Partial	0.091058	0.091058

:
```
#Lasso Regression with Alpha-0.0002
y_train_pred = lasso_modified.predict(X_train)
y_test_pred = lasso_modified.predict(X_test)

print("Lasso Regression train r2:",r2_score(y_true=y_train,y_pred=y_train_pred))
print("Lasso Regression test r2:",r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
Lasso Regression train r2: 0.9236089706982725
Lasso Regression test r2: 0.7781683582071568
```

:
```
#selecting the top 10 variables
lasso_coef.sort_values(by='mod',ascending=False).head(10)
```

:

	Feature	Coef	mod
0	LotFrontage	10.249130	10.249130
55	Condition2_RRAe	-0.770999	0.770999
3	OverallCond	0.439960	0.439960
10	HeatingQC	0.378368	0.378368
13	2ndFlrSF	0.368625	0.368625
14	LowQualFinSF	0.348884	0.348884
33	MSZoning_RH	0.328228	0.328228
2	OverallQual	0.328184	0.328184
35	MSZoning_RM	0.278846	0.278846
36	Utilities_NoSeWa	0.236642	0.236642

- Overall since the alpha values are small, we do not see a huge change in the model after doubling the alpha.

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?

- ✓ The optimum lambda value in case of Ridge and Lasso is as follows:-
  - Ridge – 1
  - Lasso – 0.0001
- ✓ The Mean Squared Error in case of Ridge and Lasso are:
  - Ridge - 0.0018396090787924262
  - Lasso - 0.0018634152629407766
- ✓ The Mean Squared Error of both the models are almost same.
- ✓ We will make use of Lasso Regression model because it is using less numbers of variables and giving almost the same accurate. Its more efficient model than Ridge regression 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?

```
#selecting the top 5 variables in lasso model with alpha=0.001
lasso_coef.sort_values(by='mod',ascending=False).head(5)
```

	Feature	Coef	mod
0	LotFrontage	10.249130	10.249130
55	Condition2_RRAe	-0.770999	0.770999
3	OverallCond	0.439960	0.439960
10	HeatingQC	0.378368	0.378368
13	2ndFlrSF	0.368625	0.368625

```
X_train_new = X_train.drop(['LotFrontage','Condition2_RRAe','OverallCond','HeatingQC','2ndFlrSF'],axis=1)
X_test_new = X_test.drop(['LotFrontage','Condition2_RRAe','OverallCond','HeatingQC','2ndFlrSF'],axis=1)

X_test_new.head()
X_train_new.shape
```

- ✓ The five most important predictor variables in the current lasso model is:-

  1. LotFrontage

  2. Condition2_RRAe

  3. OverallCond

  4. HeatingQC

  5. 2ndFlrSF

- ✓ We build a Lasso model in the Jupiter notebook after removing these attributes from the dataset.

  The new Top 5 predictors are:-

  1.LotArea

  2.KitchenAbvGr

  3. Condition2_RRAn

  4. MasVnrArea

  5. OverallQual

```
]: #selecting the top 5 variables
   lasso_coef.sort_values(by='mod',ascending=False).head(5)
```

]:

	Feature	Coef	mod
0	LotArea	10.335595	10.335595
12	KitchenAbvGr	0.929802	0.929802
51	Condition2_RRAn	-0.780277	0.780277
2	MasVnrArea	0.447469	0.447469
1	OverallQual	0.406310	0.406310

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?

- ✓ As Per, Occam's Razor— given two models that show similar 'performance' in the finite training or test data, we should pick the one that makes fewer on the test data due to following reasons:-
- ✓ Simpler models are usually more 'generic' and are more widely applicable
- ✓ Simpler models require fewer training samples for effective training than the more complex ones and hence are easier to train.
- ✓ Simpler models are more robust.
  - Complex models tend to change wildly with changes in the training dataset.
  - Simple models have low variance, high bias and complex models have low bias, high variance
  - Simpler models make more errors in the training set. Complex models lead to overfitting — they work very well for the training samples, fail miserably when applied to other test samples

- ✓ Therefore, to make the model more robust and generalizable, make the model simple but not simpler which will not be of any use.
- ✓ Regularization can be used to make the model simpler. Regularization helps to strike the delicate balance between keeping the model simple and not making it too naive to be of any use.
- ✓ For regression, regularization involves adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.
- ✓ Also, Making a model simple leads to Bias-Variance Trade-off:
    - A complex model will need to change for every little change in the dataset and hence is very unstable and extremely sensitive to any changes in the training data.
    - A simpler model that abstracts out some pattern followed by the data points given is unlikely to change wildly even if more points are added or removed.

- ✓ Bias quantifies how accurate is the model likely to be on test data. A complex model can do an accurate job prediction provided there is enough training data. Models that are too naïve, for e.g., one that gives same answer to all test inputs and makes no discrimination whatsoever has a very large bias as its expected error across all test inputs are very high.
- ✓ Variance refers to the degree of changes in the model itself with respect to changes in the training data.
- ✓ Thus accuracy of the model can be maintained by keeping the balance between Bias and Variance as it minimizes the total error as shown in the below graph.