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?

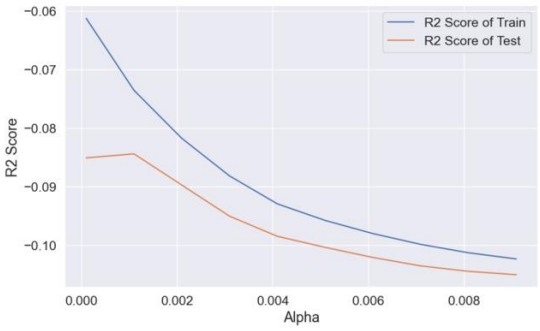
Answer:

Alpha for Ridge = 8.0 * 2 = 16
Alpha for Lasso = 0.0011 *2 = 0.0022

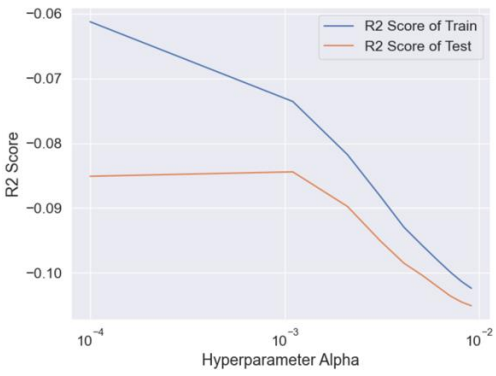
if we double the value of lambda (alpha) for both Ridge and Lasso regularizations, it would lead to increased regularization strength. In Ridge regression, higher lambda values would increase the penalty for large coefficient values, resulting in a more pronounced shrinkage effect. This would likely lead to a decrease in the magnitudes of the coefficients, making the model more conservative and less prone to overfitting.
In Lasso regression, doubling the lambda value would further increase the penalty for non-zero coefficients. As a result, more coefficients would be forced to zero, leading to a more sparse model with fewer predictors contributing to the outcome. This increased sparsity can be advantageous in feature selection and reducing model complexity.

For the assignment model doubling the alpha value haven't made much impact on the redgie but lasso has visible impact which is due to the above statement that lasso will force some non-zero co efficient to be zero when alpha increases.

R2 Score Vs Alpha Hyper parameter value chart Before Changing Alpha values:



R2 Score Vs Alpha Hyper parameter value chart After Changing Alpha values:



R2 Score train vs test: Before Changing alpha

Ridge:

R2 score (Train): 0.9376370090242531
R2 score (Test): 0.8881221175922376

Lasso:

R2 score of Train: 0.9206823430285135
R2 score of Test: 0.8799310752591424

R2 Score train vs test: After Changing alpha

Ridge:

R2 score (Train): 0.9321443816628079
R2 score (Test): 0.8851491675187515

Lasso :

R2 score (Train): 0.9049035810518966
R2 score (Test): 0.866625356776676

The R2 score has some reduction in both ridge and lasso regularizations doubling the alpha could have slightly reduced the model accuracy.

There would **not be an significant impact on list of top most important predictor variables (the ones with high co -efficient)** but the ones with least coefficient value may shrink even further in ridge and could become zero in lasso.

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:

Based on the optimal values of lambda obtained during the assignment, the optimal value for Ridge regression was found to be 8, while for Lasso regression it was 0.0011.

The choice between Ridge and Lasso regression depends on the specific requirements and goals of the analysis. Here are some factors to consider when deciding which regression technique to apply:

Model Interpretability: If interpretability of the model is important, Lasso regression may be preferred. Lasso tends to shrink less important coefficients to exactly zero, effectively performing feature selection. This allows for a more concise and interpretable model, as it highlights the most relevant features.

Handling Multicollinearity: Ridge regression is effective in dealing with multicollinearity, which occurs when predictors are highly correlated. By applying Ridge regression, the coefficients of correlated predictors are shrunk together, reducing their impact on the model. If multicollinearity is a concern, Ridge regression may be the better choice.

Performance on Test Data: It's essential to evaluate the performance of the models on test data to ensure they generalize well and avoid overfitting. You can assess metrics such as R-squared, mean squared error, or cross-validated performance to compare the predictive accuracy of the models.

Ultimately, the choice between Ridge and Lasso regression depends on the specific characteristics of your dataset, the importance of interpretability, the presence of multicollinearity, and the desired predictive performance. It's recommended to experiment with both approaches and compare their performance on validation or test data to make an informed decision.

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?

Answer:

Removed the top 5 predictors and build a new model without them adding the observed changes below:

Ridge:

Actual top 5 Predictors:

```
print("R2 score (Test): ", ridge2.score(X_test, y_test))


#checking the top 5 features identified through ridge regression
ridge_features = pd.DataFrame({'Variables': X_train.columns, 'Coefficient': ridge2.coef_})
top5ridge= ridge_features.sort_values(by='Coefficient', ascending=False).head(5)


top5ridge
```

```
R2 score (Train):  0.9376370090242531
R2 score (Test):  0.8881221175922376
```

	Variables	Coefficient
109	OverallQual_Excellent	0.113754
69	Neighborhood_Crawfor	0.082984
118	OverallCond_Excellent	0.078760
114	OverallQual_Very Good	0.073423
10	GrLivArea	0.072326

```
exl=list(top5ridge['Variables'])
```

```
exl
```

```
['OverallQual_Excellent',
 'Neighborhood_Crawfor',
 'OverallCond_Excellent',
 'OverallQual_Very Good',
 'GrLivArea']
```

```
X_train_1=X_train.drop(exl,axis=1)
X_test_1=X_test.drop(exl,axis=1)
```

```
##
```

Removed Top 5 Predictors and build the model (New Top 5 Predictors):

```
X_train_1=X_train.drop(exl,axis=1)
X_test_1=X_test.drop(exl,axis=1)
```

```
##

#building the ridge model with best  hyperparameter alpha
ridge2 = Ridge(alpha=8.0)
ridge2.fit(X_train_1, y_train)

#printing r2 score
print("R2 score (Train): ", ridge2.score(X_train_1, y_train))
print("R2 score (Test): ", ridge2.score(X_test_1, y_test))


#checking the top 5 features identified through ridge regression
ridge_features = pd.DataFrame({'Variables': X_train_1.columns, 'Coefficient': ridge2.coef_})
top5ridge= ridge_features.sort_values(by='Coefficient', ascending=False).head(5)
```

```
R2 score (Train):  0.9324869806434816
R2 score (Test):  0.8822630421636617
```

```
top5ridge
```

	Variables	Coefficient
8	2ndFlrSF	0.111649
7	1stFlrSF	0.093750
82	Neighborhood_Somerst	0.075466
83	Neighborhood_StoneBr	0.073953
215	Functional_Typ	0.071225

Lasso:

Actual top 5 Predictors:

```
### Lasso


####### Lassso

#building the lasso model with best parameters
lasso = Lasso(alpha=0.0011)
lasso.fit(X_train, y_train)

#printing the training and testing R2 score
print("R2 score (Train): ", lasso.score(X_train, y_train))
print("R2 score (Test): ", lasso.score(X_test, y_test))

lasso_features = pd.DataFrame({'Variables': X_train.columns, 'Coefficient': lasso.coef_})
top5lasso=lasso_features.sort_values(by='Coefficient', ascending=False).head(5)
top5lasso
```

```
R2 score (Train):  0.9206823430285135
R2 score (Test):  0.8799310752591424
```

	Variables	Coefficient
109	OverallQual_Excellent	0.134314
10	GrLivArea	0.095133
69	Neighborhood_Crawfor	0.085153
84	Neighborhood_Somerst	0.080214
114	OverallQual_Very Good	0.079067

```
exl=list(top5lasso['Variables'])
```

```
X_train_1=X_train.drop(exl,axis=1)
X_test_1=X_test.drop(exl,axis=1)
```

Removed Top 5 Predictors and build the model (New Top 5 Predictors):

```
###### Lassso

#building the lasso model with best parameters
lasso = Lasso(alpha=0.0011)
lasso.fit(X_train_1, y_train)

#printing the training and testing R2 score
print("R2 score (Train): ", lasso.score(X_train_1, y_train))
print("R2 score (Test): ", lasso.score(X_test_1, y_test))

lasso_features = pd.DataFrame({'Variables': X_train_1.columns, 'Coefficient': lasso.coef_})
top5lasso=lasso_features.sort_values(by='Coefficient', ascending=False).head(5)
top5lasso
```

```
R2 score (Train):  0.9139450660381777
R2 score (Test):  0.8753674251789061
```

	Variables	Coefficient
8	2ndFlrSF	0.122027
7	1stFlrSF	0.103963
45	MSZoning_FV	0.094424
215	Functional_Typ	0.072082
126	Exterior1st_BrkFace	0.068497

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:

Ensuring that a model is robust and generalizable is crucial to its effectiveness and reliability. Here are some key considerations and practices to achieve robustness and generalizability:

Data Quality and Preprocessing: Start by ensuring the quality and cleanliness of your data. This includes handling missing values, outliers, and any data inconsistencies. Additionally, preprocessing steps such as feature scaling, normalization, and encoding categorical variables should be applied consistently to both the training and test data.

Train-Test Split: Split your dataset into training and test sets. The training set is used to train the model, while the test set is used to evaluate its performance on unseen data. It is important to keep the test set completely separate from the training process and only use it for final evaluation.

Cross-Validation: Use cross-validation techniques, such as k-fold cross-validation, to obtain a more robust estimate of the model's performance. This helps to evaluate the model on multiple subsets of the data and provides a more reliable estimate of its generalizability.

Regularization Techniques: Regularization methods like Ridge and Lasso regression can improve the robustness of the model by reducing overfitting and handling multicollinearity. These techniques add a penalty to the loss function, preventing the model from becoming too complex and increasing its generalizability.

Feature Selection: Consider using feature selection techniques to identify the most relevant and informative features for the model. This helps to reduce the complexity of the model, improve interpretability, and potentially enhance its generalizability.

Outliers and Anomaly Detection: Identify and handle outliers in the data appropriately. Outliers can significantly impact the model's performance and generalizability. You can consider techniques such as outlier detection algorithms or data transformations to address outliers.

Implications for Model Accuracy:
Ensuring robustness and generalizability of a model may have implications for its accuracy. By applying practices such as cross-validation and regularization, the model aims to strike a balance between bias and variance. The model may exhibit slightly higher bias (lower training accuracy) to achieve lower variance and better generalization to unseen data. This trade-off can lead to slightly lower accuracy on the training data but better accuracy on the test or validation data, ultimately improving the model's overall performance and reliability.

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?

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?

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?