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:

Optimal value for alpha for ridge and lasso regression are as follows:

- Ridge Regularization Alpha: **5**

```
# Printing the best hyperparameter alpha
print(model_cv.best_params_)
```

```
{'alpha': 5.0}
```

- Lasso Regularization Alpha: **0.0001**

```
# Printing the best hyperparameter alpha
print(model_cv.best_params_)
```

```
{'alpha': 0.0001}
```

Doubling alpha in both cases will make the regularization more aggressive than needed leading to a decrease in r2_score.

```
# Calculate r2_score,RSS and mean_squared_error for ridge regularization with alpha =10
y_pred_train = ridge.predict(X_train)
y_pred_test = ridge.predict(X_test)

r2_train_lr = r2_score(y_train, y_pred_train)
print("r2_train:",r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print("r2_test:",r2_test_lr)
```

```
r2_train: 0.8763979567973651
r2_test: 0.8601170560096947
```

```
# Calculate r2_score,RSS and mean_squared_error for lasso regularization with alpha =0.0002
y_pred_train = lasso.predict(X_train)
y_pred_test = lasso.predict(X_test)

r2_train_lr = r2_score(y_train, y_pred_train)
print("r2_train:",r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print("r2_test:",r2_test_lr)
```

```
r2_train: 0.8868067293968984
r2_test: 0.8704235251850165
```

Important predictors variables after the change is implemented i.e doubling alpha are as follows:

```
# List top 5 important variables according to Ridge Regularized model that affect Sale Price (alpha=10):
print(betas['Ridge'].sort_values(ascending = False)[:5])
```

```
OverallQual            0.077680
GrLivArea              0.068713
2ndFlrSF               0.057081
GarageCars             0.055826
Neighborhood_NoRidge   0.050920
Name: Ridge, dtype: float64
```

```
# List top 5 important variables according to Lasso Regularized model that affect Sale Price (alpha=0.0002):
print(betas['Lasso'].sort_values(ascending = False)[:5])
```

```
GrLivArea              0.310389
OverallQual            0.132236
GarageCars             0.065671
Neighborhood_NoRidge   0.057761
Neighborhood_NridgHt   0.047924
Name: Lasso, dtype: float64
```

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:

Ridge regression seem to be overall better model as it has slightly better r2 score, RSS and MSE than lasso on test set. However, it should be noted that Lasso seems to perform almost as well and is a much simpler model as it has less number of features and can result in significant savings in production. Therefore, the choice of model is heavily dependent on the business use case and accuracy tolerance. Since the number of features in this case are very less, I would go with Ridge regularized model as its slightly more accurate.

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:

Below are the top 5 most important predictor variables in the lasso model:

Top 5 important variables according to Lasso Regularized model that affect Sale Price:

```
: # List top 5 important variables according to Lasso Regularized model that affect Sale Price:
print(betas['Lasso'].sort_values(ascending = False)[:5])
```

```
GrLivArea           0.327169
OverallQual         0.120991
LotArea             0.071714
RoofMatl_WdShngl    0.067915
GarageCars          0.067223
Name: Lasso, dtype: float64
```

If 'GrLivArea', 'OverallQual'.'LotArea','RoofMatl_wdShngl' and 'GarageCars' is not available in incoming data then we will use the next 5 most important variables. Dropping above variables and building a new model, we get below list of next top 5 important variables:

```
# List top 5 important variables according to Lasso Regularized model that affect
# Sale Price (after dropping previous top 5 variables):
print(betas['Lasso'].sort_values(ascending = False)[:5])
```

```
TotalBsmtSF             0.231637
2ndFlrSF                0.168807
GarageCars              0.087821
Neighborhood_NoRidge    0.064074
Neighborhood_StoneBr    0.060083
Name: Lasso, dtype: float64
```

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:

A model can perform well on training data but may not perform so well on the test data. Such model is said to have high variance (Overfitting). In this case the model is very complex and not generalized. On the other had a too simple model may fail to recognize patterns in the data and may suffer from high bias (underfitting). A model is said to be robust and generalized when it has low bias and low variance i.e., its effectively able to understand patterns in the data and can predict correctly on unseen data. For an accurate model, we need to manage model complexity such that its neither too high and neither too low (bias/variance trade-off).

This can be achieved by using regularization. Regularization helps manage model complexity by shrinking model coefficient estimates towards zero. This discourages models from becoming overly complex without hindering the learning process.