Advanced Regression Subjective Questions

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:

The Optimal value of alpha for ridge and lasso regression is obtained by fetching the "best_params_" dictionary attribute from sklearn.model_selection.GridSearchCV object. In this dict attribute, we look for the value of "alpha" key, this gives the optimal alpha.In the assignment, the optimal value of lambda for ridge is 8 and for the Lasso is 50.

Using these optimal values of alpha, and if we choose to double the value of optimal alpha, we can observe the reduced performance metric of r2 score.

a) Lasso regression
   a. optimal alpha value is 50

```
print('Best alpha value obtained after cross validation:', gridsearch_cv_model.best_params_)
lasso_model = Lasso(alpha=50)
lasso_model.fit(X_train, y_train)

y_train_pred = lasso_model.predict(X_train)
print("R2 for Lasso Model on training set:\t", r2_score(y_true=y_train, y_pred=y_train_pred))

y_test_pred = lasso_model.predict(X_test)
print("R2 for Lasso Model on test set:\t\t", r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
Best alpha value obtained after cross validation: {'alpha': 50}
R2 for Lasso Model on training set:      0.9408202494114437
R2 for Lasso Model on test set:          0.9204604878066895
```

   b. Double value of alpha is 100

```
lasso_model = Lasso(alpha=100)
lasso_model.fit(X_train, y_train)

y_train_pred = lasso_model.predict(X_train)
print("R2 for Lasso Model on training set:\t", r2_score(y_true=y_train, y_pred=y_train_pred))

y_test_pred = lasso_model.predict(X_test)
print("R2 for Lasso Model on test set:\t\t", r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
Best alpha value obtained after cross validation: {'alpha': 8.0}
R2 for Lasso Model on training set:      0.9335208626260618
R2 for Lasso Model on test set:          0.9165159421687553
```

So, both training and test score dropped by about 0.01

b) Ridge Regression

a. Optimal alpha value is 8

```
print('Best alpha value obtained in ridge regression after cross validation:', gridsearch_cv_model.best_params_)
ridge_model = Ridge(alpha=8)
ridge_model.fit(X_train, y_train)

y_train_pred = ridge_model.predict(X_train)
print("R2 for Ridge Model on training set:\t", r2_score(y_true=y_train, y_pred=y_train_pred))

y_test_pred = ridge_model.predict(X_test)
print("R2 for Ridge Model on test set:\t\t", r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
Best alpha value obtained in ridge regression after cross validation: {'alpha': 8.0}
R2 for Ridge Model on training set:      0.9388006509940007
R2 for Ridge Model on test set:          0.9180081572031796
```

b. Double value of alpha is 16

```
ridge_model = Ridge(alpha=16)
ridge_model.fit(X_train, y_train)

y_train_pred = ridge_model.predict(X_train)
print("R2 for Ridge Model on training set:\t", r2_score(y_true=y_train, y_pred=y_train_pred))

y_test_pred = ridge_model.predict(X_test)
print("R2 for Ridge Model on test set:\t\t", r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
Best alpha value obtained in ridge regression after cross validation: {'alpha': 8.0}
R2 for Ridge Model on training set:      0.9334577751394615
R2 for Ridge Model on test set:          0.9145321897745896
```

So, although the performance of the ridge model also dropped but it just fell by about approx. 0.005.

The most important predictor variables are the following, after we changed the alpha value to double of the most optimal value for both Lasso & Ridge.

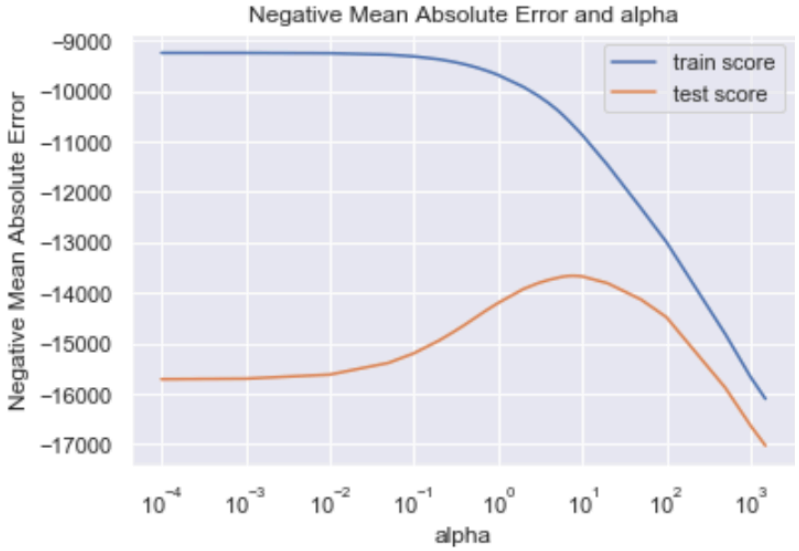
1. Neighborhood: Houses near Crawford, Northridge Heights and Northridge area are expensive.
2. Functional: Homes with typical functionality are high in prices
3. OverallQual: Overall material and finish of the house increases the price
4. KitchenQual: Overall kitchen quality
5. BsmtQual: Evaluates the height of the basement

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:

We would decide which optimal value of lambda for ridge and lasso regression to choose, by plotting the curve between alpha values and Negative Mean Absolute Error, and where we get the best r2 score on training and test datasets.

For ridge regression, we observe the best performance on alpha value of 8



And the r2 score on training and test is as follows:

```
Best alpha value obtained in ridge regression after cross validation: {'alpha': 8.0}
R2 for Ridge Model on training set:      0.9388006509940007
R2 for Ridge Model on test set:          0.9180081572031796
```

When compared with Lasso, we observe in the following r2 score statistics, and obtain the best performance using optimal value of alpha as 50 for Lasso regression for both training and test datasets. It's better compared to Ridge regression model stated above.

```
print('Best alpha value obtained after cross validation:', gridsearch_cv_model.best_params_)
lasso_model = Lasso(alpha=50)
lasso_model.fit(X_train, y_train)

y_train_pred = lasso_model.predict(X_train)
print("R2 for Lasso Model on training set:\t", r2_score(y_true=y_train, y_pred=y_train_pred))

y_test_pred = lasso_model.predict(X_test)
print("R2 for Lasso Model on test set:\t\t", r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
Best alpha value obtained after cross validation: {'alpha': 50}
R2 for Lasso Model on training set:      0.9408202494114437
R2 for Lasso Model on test set:          0.9204604878066895
```

Also, the plot between alpha value and the negative mean absolute error also converges to the best at the optimal alpha value of 50.

Thus, I would apply lambda value equal to 50 along with Lasso regression model due to following reasons:

a) Lasso regression helps in feature elimination by making the coefficients of insignificant variables as zero, thus giving only those variables which seem to be significant in determining the sale price.
b) Lasso regression also makes the model simpler by reducing the number of significant features.
c) Also, on the basis of plots between alpha value and Negative mean absolute error where we have good r2 score on the training as well as on the test dataset.

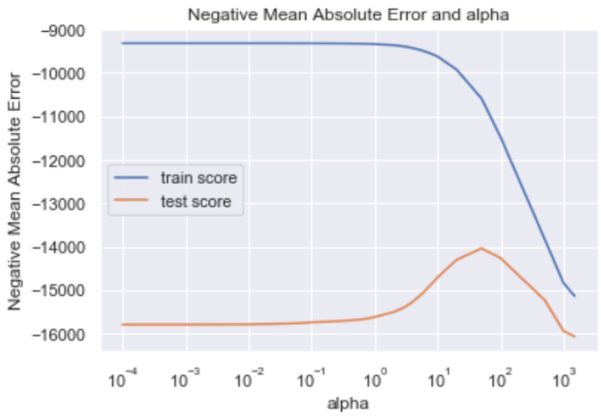
3. After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Answer:

Using the optimal value of alpha as 50 in the Lasso regression model, we initially obtained the following most important predictor variables.

a) Neighborhood_Crawfor: Crawford area
b) Functional_Typ: Typical home functionality
c) Neighborhood_NridgHt: Northridge Heights area
d) Neighborhood_NoRidge: Northridge area
e) GarageQual_Gd: Good garage quality

Now, excluding these above-mentioned predictor variables and building a new Lasso regression model, we get following model in place having this relation between alpha values and negative mean absolute error



Model performance is as follows:

```
Best alpha value obtained after cross validation: {'alpha': 50}
R2 for Lasso Model on training set:      0.9383345945203589
R2 for Lasso Model on test set:          0.9191222859189445
```

Finally, the five most important predictor variables obtained after rebuilding the Lasso regression model as above, are the following:

a) SaleType_CWD: Type of sale as Warranty Deed of cash
b) Exterior1st_BrkFace: Exterior covering on house as brick face
c) SaleCondition_Partial: Condition of sale as Home was not completed when last assessed (associated with New Homes)
d) LandContour_HLS: Flatness of the property as Hillside - Significant slope from side to side
e) Condition1_Norm: Proximity to various conditions is Normal

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 is considered to be robust if the model is stable, i.e. does not change drastically upon changing the training set. Models are trained on a set of training data, but their efficiency is determined by the ability to perform well on the unseen test data.

Extremely complex models don’t generalize well since they are prone to change with small changes in the input data. Extremely simple models are likely to fail in predicting hence are prone to make errors and less accurate.

Hence, we should always select a model which is just complex enough to understand the variance in the data without much inaccuracy at the same time not too complex to overfit.

To make sure that the model is robust and generalized we can achieve this using regularization, so that our model is as simple as possible and shouldn't use redundant features.

Its implication in terms of accuracy is that a robust and generalizable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data.