Assignment - Advanced Regression

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?

From Lasso regression or L1 regularization the optimal value of alpha is found using best_params_ as (Alpha= 0.001) for which the training accuracy is 0.90 and testing accuracy is 0.78.
From Ridge regression or L2 regularization the optimal value of alpha is found using best_params_ as (Alpha= 100) for which the training accuracy is 0.90 and testing accuracy is 0.79.

When we double the value of alpha the neg_mean_absolute error will become more negative and the training accuracy will further drop down.

For Lasso regression on doubling the value of Alpha in model we see a drop in training and testing accuracy to 0.89 and 0.76 respectively.

And the most important predictor variables after the change are **MSZoning, HouseStyle, YearRemodAdd, GarageQual and Electrical.**

For Ridge regression on doubling the value of Alpha in model we see drop in training and testing accuracy to 0.88 and 0.76 respectively.

The most important predictor variables after the change are **MSZoning, YearRemodAdd, HouseStyle, Fireplaces and GarageQual.**

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?

Out of L1(Lasso) and L2(Ridge) regression model ran in the assignment, we are getting the values of test accuracy as nearly same with value for Ridge regression model higher than Lasso regression.

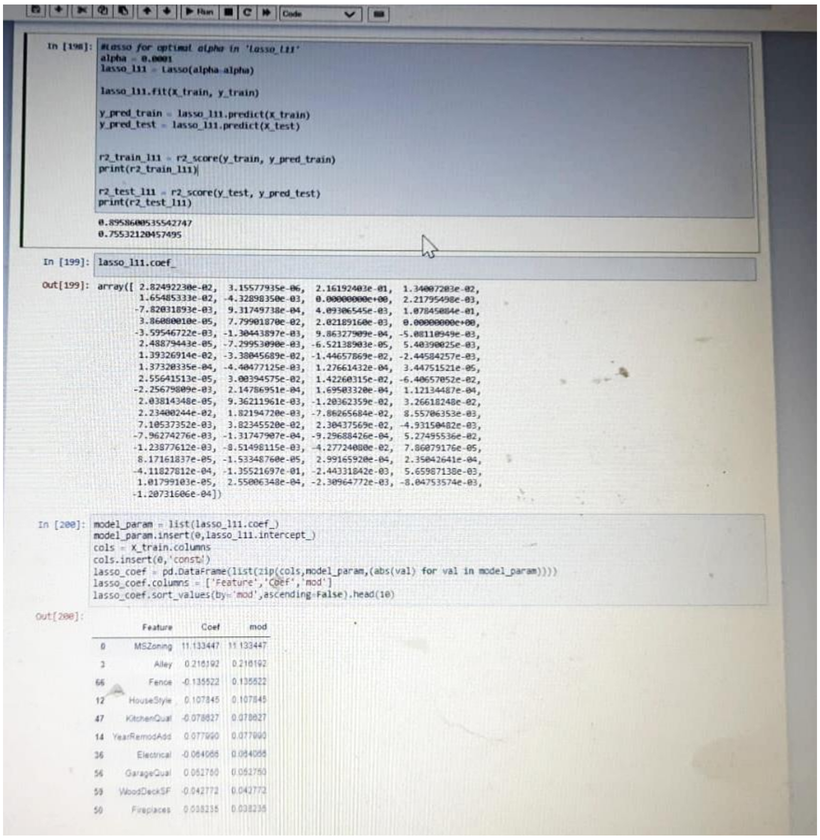
However if we look at the most important predictor variables they are almost same for both. We can see that lasso regression eliminates a lot of feature that are related to each other and ridge regression does not, it adds bias to reach optimal model. Hence we can choose Ridge regression as it is performing better on the unseen test data and comparatively more robust.

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?

After building the model if most important top 5 predictors in the lasso model are not available in incoming data then we need create another model without these predictors which are **MSubClass, BldgType, OverallCond, YearBuilt and GarageArea.**

After recreating the lasso model as done in the python code below the most important predictors now are **MSZoning, Alley, Fence, HouseStyle and KitchenQual.**



```
In [198]: #Lasso for optimal alpha in 'lasso_l11'
          alpha = 0.0001
          lasso_l11 = Lasso(alpha=alpha)

          lasso_l11.fit(X_train, y_train)

          y_pred_train = lasso_l11.predict(X_train)
          y_pred_test = lasso_l11.predict(X_test)


          r2_train_l11 = r2_score(y_train, y_pred_train)
          print(r2_train_l11)

          r2_test_l11 = r2_score(y_test, y_pred_test)
          print(r2_test_l11)

          0.8958600535542747
          0.75532120457495

In [199]: lasso_l11.coef_

Out[199]: array([ 2.82492230e-02,  3.15577935e-06,  2.16192403e-01,  1.34007203e-02,
                  1.65485333e-02, -4.32898350e-03,  0.00000000e+00,  2.21795498e-03,
                 -7.82031893e-03,  9.31749738e-04,  4.09306545e-03,  1.07845084e-01,
                  3.86080010e-05,  7.79901870e-02,  2.02189160e-03,  0.00000000e+00,
                 -3.59546722e-03, -1.30443897e-03,  9.86327909e-04, -5.08110949e-03,
                  2.48879443e-05, -7.29953090e-03, -6.52138903e-05,  5.40390025e-03,
                  1.39326914e-02, -3.38045689e-02, -1.44657869e-02, -2.44584257e-03,
                  1.37320335e-04, -4.40477125e-03,  1.27661432e-04,  3.44751521e-05,
                  2.55641513e-05,  3.00394575e-02,  1.42260315e-02, -6.40657052e-02,
                 -2.25679809e-03,  2.14786951e-04,  1.69503320e-04,  1.12134487e-04,
                  2.03814348e-05,  9.36211961e-03, -1.20362359e-02,  3.26618248e-02,
                  2.23400244e-02,  1.82194720e-03, -7.86265684e-02,  8.55706353e-03,
                  7.10537352e-03,  3.82345520e-02,  2.30437569e-02, -4.93150482e-03,
                 -7.96274276e-03, -1.31747907e-04, -9.29688426e-04,  5.27495536e-02,
                 -1.23877612e-03, -8.51498115e-03, -4.27724080e-02,  7.86079176e-05,
                  8.17161837e-05, -1.53348760e-05,  2.99165920e-04,  2.35042641e-04,
                 -4.11827812e-04, -1.35521697e-01, -2.44331842e-03,  5.65987138e-03,
                  1.01799103e-05,  2.55006348e-04, -2.30964772e-03, -8.04753574e-03,
                 -1.20731606e-04])

In [200]: model_param = list(lasso_l11.coef_)
          model_param.insert(0,lasso_l11.intercept_)
          cols = X_train.columns
          cols.insert(0,'const')
          lasso_coef = pd.DataFrame(list(zip(cols,model_param,(abs(val) for val in model_param))))
          lasso_coef.columns = ['Feature','Coef','mod']
          lasso_coef.sort_values(by='mod',ascending=False).head(10)

Out[200]:
```

	Feature	Coef	mod
0	MSZoning	11.133447	11.133447
3	Alley	0.216192	0.216192
66	Fence	-0.135522	0.135522
12	HouseStyle	0.107845	0.107845
47	KitchenQual	-0.078627	0.078627
14	YearRemodAdd	0.077990	0.077990
36	Electrical	-0.064066	0.064066
56	GarageQual	0.052750	0.052750
59	WoodDeckSF	-0.042772	0.042772
50	Fireplaces	0.038235	0.038235

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?

A robust and generalizable machine learning model tends to be better working on unseen or test data.
One way is that we can simply split your original data set into 3: training, validation and test; use training and validation for the model development and keep test data unseen for a final check after cross validation. This will check your model's generalizability.

We can also make our machine learning model more robust and generalizable by using the following options:

Regularization: With the help of regularization we can reduce over-fitting and variance. This will give us a model which will work well on unseen data using the optimal values of hyperparameter.

Also we can choose a more practical error metric for eg Mean absolute error or Huber loss.

Data preparation: When the data is heavily skewed for eg target variable in our House prediction dataset is heavily positive skewed we can transform the data into Log values to get it near to normal distribution as machine learning models tend to work much better with normally distributed target variables.

In case there are extreme values present in the dataset we can cap the data at various thresholds from top and bottom percentiles. For eg we can cap the bottom and top 5% values at 5th and 95th percentiles. This is called Winsorization.

Removing/ handling outlier values and Winsorize basically means removing information from dataset and are generally not preferred.

Choosing different models: Model selection is the best way to look for a machine learning model that is best suitable to dataset in our problem and performs better with unseen data.