Outsatnding-Project-1

By:
Name: Jatin Jindal
E-mail Id: jatinjindal1199@gmail.com

Problem Statement 2

Q1: 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?

Ans:

Ridge Regression

```
[106]  1 from sklearn.model_selection import GridSearchCV
       2 rd=Ridge(random_state=0)
       3 para={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55,100]}
       4 print(para)
       5 grid_rig=GridSearchCV(estimator=rd,
       6                       param_grid=para,scoring='neg_mean_squared_error',
       7                       cv=7,n_jobs=-1)
       8 grid_rig=grid_rig.fit(x_train_sc,y_train_sc)
       9 print('Best score= ',grid_rig.best_score_)
      10 print('Best parameter= ',grid_rig.best_params_)

  {'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.01, 1, 5, 10, 20, 30, 35, 40, 45, 50, 55, 100]}
  Best score=  -0.147313430387144
  Best parameter=  {'alpha': 100}
```

Therefore the best/optimal value of the Parameter i.e. **alpha is 100**

When alpha is 100(best parameter)

```
1 #Model training
2 rd=Ridge(alpha=100,random_state=0)
3 rd.fit(x_train_sc,y_train)
4 print(rd.score(x_test_sc,y_test))
5 rd.coef_

0.6689229839919186
array([[-3048.00706915,  -633.61982414,   758.87555606,  5098.66500945,
         2333.36919994,  -811.40646444,  1444.97867656, -1013.87772001,
          261.33654598,   170.94018532,   602.15336011, -1624.8558183 ,
        -5819.31798168, -1372.86841195, -1545.68720009, 11976.71828491,
         5155.19562748,  3998.07134353,  1816.92173716,  2044.11515645,
         2418.55831077, -2377.20339486,  1354.13063804,  3494.67410842,
         6595.70099945, -6211.81024196,   492.46664609,  1735.5522855 ,
        -7409.12403275,  1401.67783542, -2744.84880767,    52.12478705,
         6492.91278612,  1009.1544451 ,  1950.96337967, -1259.16303858,
         6282.33787243,  -349.54039811, -1449.96460026,   202.43885417,
         -716.44080032,  7823.26401736,  7700.72808695,  -415.11152155,
        12213.87938843,  1386.46532505,   310.23694634,  2656.57486218,
         2326.84217667, -2424.29467972, -3670.29409953, -5977.70973825,
         5847.61332841,  3757.6385649 ,  3664.32365719,   512.63276155,
          597.79783718,  -273.27736685,  2136.06218166,  3609.85090441,
         -503.82174006,  1345.73806933,   133.56923432,  1762.57387528,
          -93.66515322,   417.73168674,   612.28757262,  1520.71230466,
          738.9583497 ,  1040.87056182, -1191.90452529, -1449.29678181,
        -1003.86141122,  3701.52662245]])
```

When alpha is 200(best parameter is doubled)

```
[145]  1 # Model training for Double alpha value
       2 rd=Ridge(alpha=200,random_state=0)
       3 rd.fit(x_train_sc,y_train)
       4 print(rd.score(x_test_sc,y_test))
       5 rd.coef_
```

```
0.6905778045575908
array([[-2525.48744165,  -553.38226371,   983.9553837 ,  4689.17040184,
         2198.19393702,  -956.596697  ,  1184.4469458 ,  -955.92278592,
          115.4548865 ,   292.88245943,   679.5876436 , -1407.26445007,
        -4980.93914879, -1606.0452403 , -1251.41128565, 11226.01702561,
         4550.4999441 ,  2987.96487471,  2213.76351408,  2148.62070563,
         2456.01804127, -1817.4513545 ,   906.54575219,  3195.62195229,
         6388.75228024, -6183.80899955,   452.90496455,  1805.32001026,
        -7118.50894356,  1263.29163155, -2613.17672391,  -183.44971117,
         6143.73699888,   842.45191171,  1647.19464407,  -850.90207134,
         6229.08818156,  -219.15837516, -1660.8202558 ,   483.84106276,
         -578.26049837,  7236.33398042,  6748.26807478,  -302.87510846,
        10986.41028769,  1625.1416286 ,   311.63794178,  3137.5253377 ,
         2507.4737319 , -1694.57238218, -3439.62558985, -5996.73579239,
         5748.63570592,  3470.40537149,  3963.39824653,   298.19568935,
          719.4740466 ,  -688.77061097,  2668.8029821 ,  3790.92027037,
         -460.73554151,  1154.27725806,   393.8713564 ,  1865.95454764,
          256.00661105,   407.60473995,   570.64345023,  1516.11864924,
          678.44220837,   782.82455458, -1050.85922987, -1277.11400078,
         -955.55212804,  3449.73692067]])
```

We can see that the value of coefficients decreases as the value of alpha is doubled. When the value of alpha is increased in lasso regression the value of coefficients tends to zero but it never reaches the value of zero.

Finding variable with max coefficient value

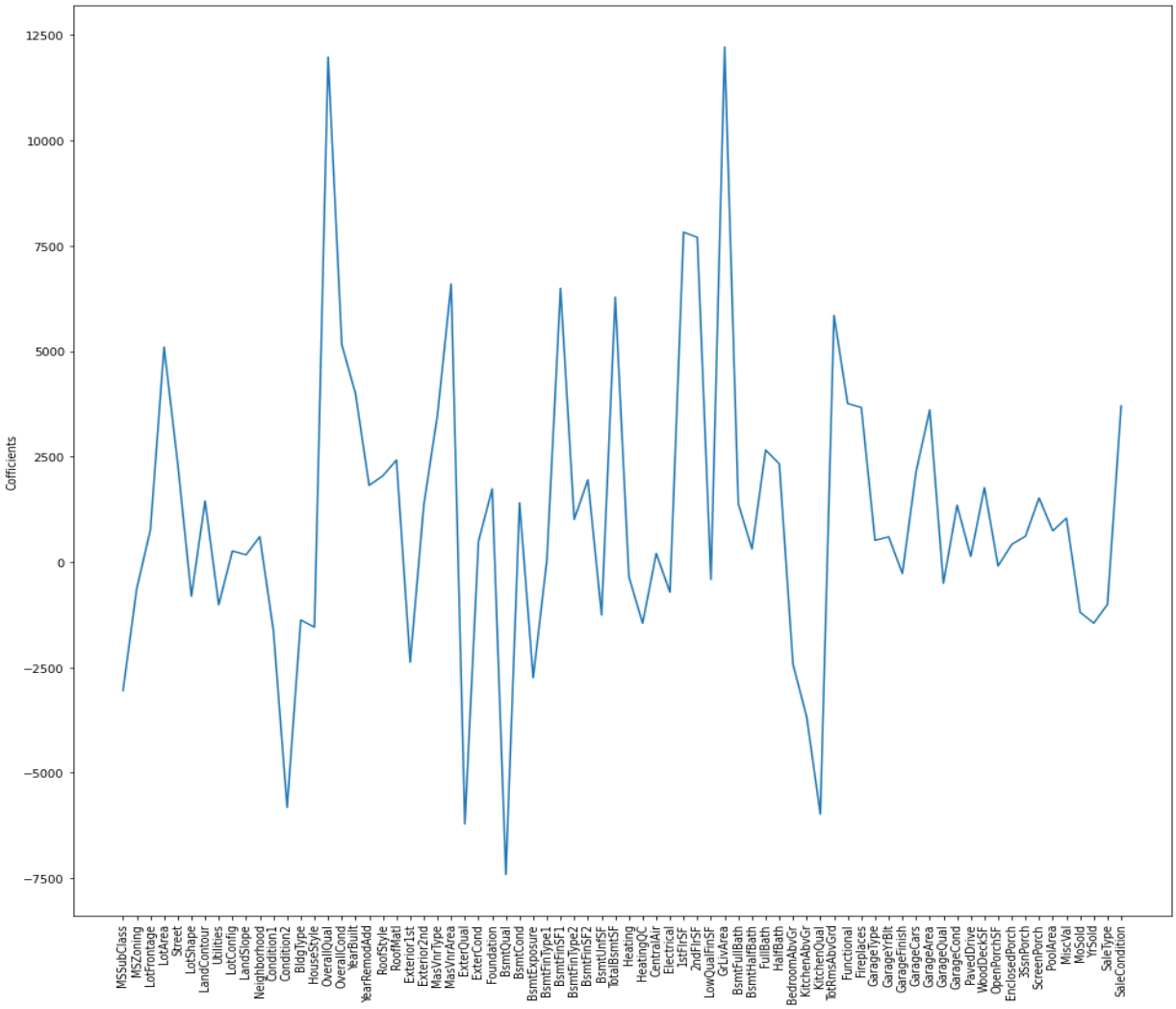
```
1 # Finding Most important Variable/Column
2 rd_coef[3]
3 m=max(rd_coef)
4 print(m)
5 for i in range(len(col)):
6     if rd_coef[i]==m:
7         imp_pred=col[i]
8 print(imp_pred)
```

```
11226.017025608819
OverallQual
```

As **OverallQual variable** has highest Coefficient value therefore it is most important variable

We can also see this from the graph plotted below for all the variables with coefficients for

Ridge Regression



Lasso Regression

```
[155]  1 #Hyperparameter tuning of Lasso Regression
       2 from sklearn.model_selection import GridSearchCV
       3 ls=Lasso(random_state=0)
       4 para={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55,100]}
       5 print(para)
       6 grid_rig=GridSearchCV(estimator=ls,
       7                       param_grid=para,scoring='neg_mean_squared_error',
       8                       cv=10,n_jobs=-1)
       9 grid_rig=grid_rig.fit(x_train_sc,y_train_sc)
      10 print('Best Score= ',grid_rig.best_score_)
      11 print('Best parameter= ',grid_rig.best_params_)
```

```
{'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.01, 1, 5, 10, 20, 30, 35, 40, 45, 50, 55, 100]}
Best Score=  -0.15083391209308183
Best parameter=  {'alpha': 0.01}
```

From the Above Hyperparameter tuning we can see that the best value of the **alpha for Lasso regression is 0.01**

When alpha is 0.01(best parameter)

```
[147]  1 # Model training
       2 ls=Lasso(alpha=0.01,random_state=0)
       3 ls.fit(x_train_sc,y_train)
       4 print(ls.score(x_test_sc,y_test))
       5 ls.coef_
```

```
0.628856519803142
array([-4.14969279e+03, -6.82184558e+02,  4.37084965e+02,  5.67946682e+03,
        2.45364283e+03, -5.35142402e+02,  1.80759627e+03, -1.06379165e+03,
        4.94929241e+02,  1.12822846e+01,  5.92438585e+02, -1.90137459e+03,
       -7.13392577e+03, -7.05402256e+02, -1.90111075e+03,  1.26853161e+04,
        6.26820335e+03,  7.52303891e+03,  8.76746106e+02,  1.86249315e+03,
        2.24730411e+03, -3.46012889e+03,  2.46571461e+03,  3.87454255e+03,
        6.70397208e+03, -6.04906866e+03,  4.43655011e+02,  1.23690399e+03,
       -7.69840296e+03,  1.51413484e+03, -2.91573497e+03,  4.43337996e+02,
        1.14721368e+04,  1.35179548e+03,  4.09338075e+03,  2.64941480e+03,
        1.85616894e+03, -5.83377757e+02, -1.14430898e+03, -4.28243024e+02,
       -8.00632291e+02,  1.75479430e+04,  1.96340274e+04,  6.75624363e+02,
        3.09104388e+03,  8.53773543e+02,  2.80554549e+02,  1.22934577e+03,
        1.53231988e+03, -3.62911694e+03, -3.88719185e+03, -5.92079759e+03,
        5.79812094e+03,  4.15091174e+03,  3.11341742e+03,  8.95790035e+02,
        2.47977738e+02,  3.26832664e+02,  1.07518722e+03,  3.45588121e+03,
       -6.80178840e+02,  1.69193547e+03, -4.96607181e+02,  1.57313034e+03,
       -6.07768331e+02,  5.83982843e+02,  6.89606768e+02,  1.54035109e+03,
        7.95407741e+02,  1.43694588e+03, -1.34368668e+03, -1.64382919e+03,
       -9.94218592e+02,  4.01371907e+03])
```

When alpha is 0.02(best parameter is doubled)

```
[148] 1 # Model training for Double alpha value
      2 ls=Lasso(alpha=0.02,random_state=0)
      3 ls.fit(x_train_sc,y_train)
      4 print(ls.score(x_test_sc,y_test))
      5 ls.coef_
```

```
0.6288573859278459
array([-4.14970871e+03, -6.82180777e+02,  4.37072981e+02,  5.67946454e+03,
        2.45363124e+03, -5.35134684e+02,  1.80757326e+03, -1.06377823e+03,
        4.94909740e+02,  1.12719321e+01,  5.92425202e+02, -1.90136901e+03,
       -7.13389588e+03, -7.05387436e+02, -1.90107573e+03,  1.26853489e+04,
        6.26816951e+03,  7.52296535e+03,  8.76753382e+02,  1.86248370e+03,
        2.24729600e+03, -3.46003956e+03,  2.46564220e+03,  3.87453593e+03,
        6.70395586e+03, -6.04907786e+03,  4.43640249e+02,  1.23689627e+03,
       -7.69840985e+03,  1.51412799e+03, -2.91572439e+03,  4.43310645e+02,
        1.14708947e+04,  1.35176202e+03,  4.09289371e+03,  2.64816145e+03,
        1.85733233e+03, -5.83349484e+02, -1.14430575e+03, -4.28201009e+02,
       -8.00614434e+02,  1.75473913e+04,  1.96333493e+04,  6.75540331e+02,
        3.09183530e+03,  8.53741492e+02,  2.80527401e+02,  1.22932177e+03,
        1.53230607e+03, -3.62907559e+03, -3.88716267e+03, -5.92080569e+03,
        5.79806867e+03,  4.15090523e+03,  3.11340476e+03,  8.95767034e+02,
        2.47958971e+02,  3.26797144e+02,  1.07518981e+03,  3.45588582e+03,
       -6.80152711e+02,  1.69190156e+03, -4.96580241e+02,  1.57312252e+03,
       -6.07744507e+02,  5.83962269e+02,  6.89593534e+02,  1.54033835e+03,
        7.95393734e+02,  1.43692810e+03, -1.34367664e+03, -1.64381745e+03,
       -9.94204048e+02,  4.01370881e+03])
```

As we see that there is almost negligible effect in the coefficients value as even after doubling the change in the value of the alpha is very small.

But as the value of alpha increases the coefficients values goes on decreasing and the ultimately it will reach the value of zero in case of Lasso regression.

Finding variable with max coefficient value

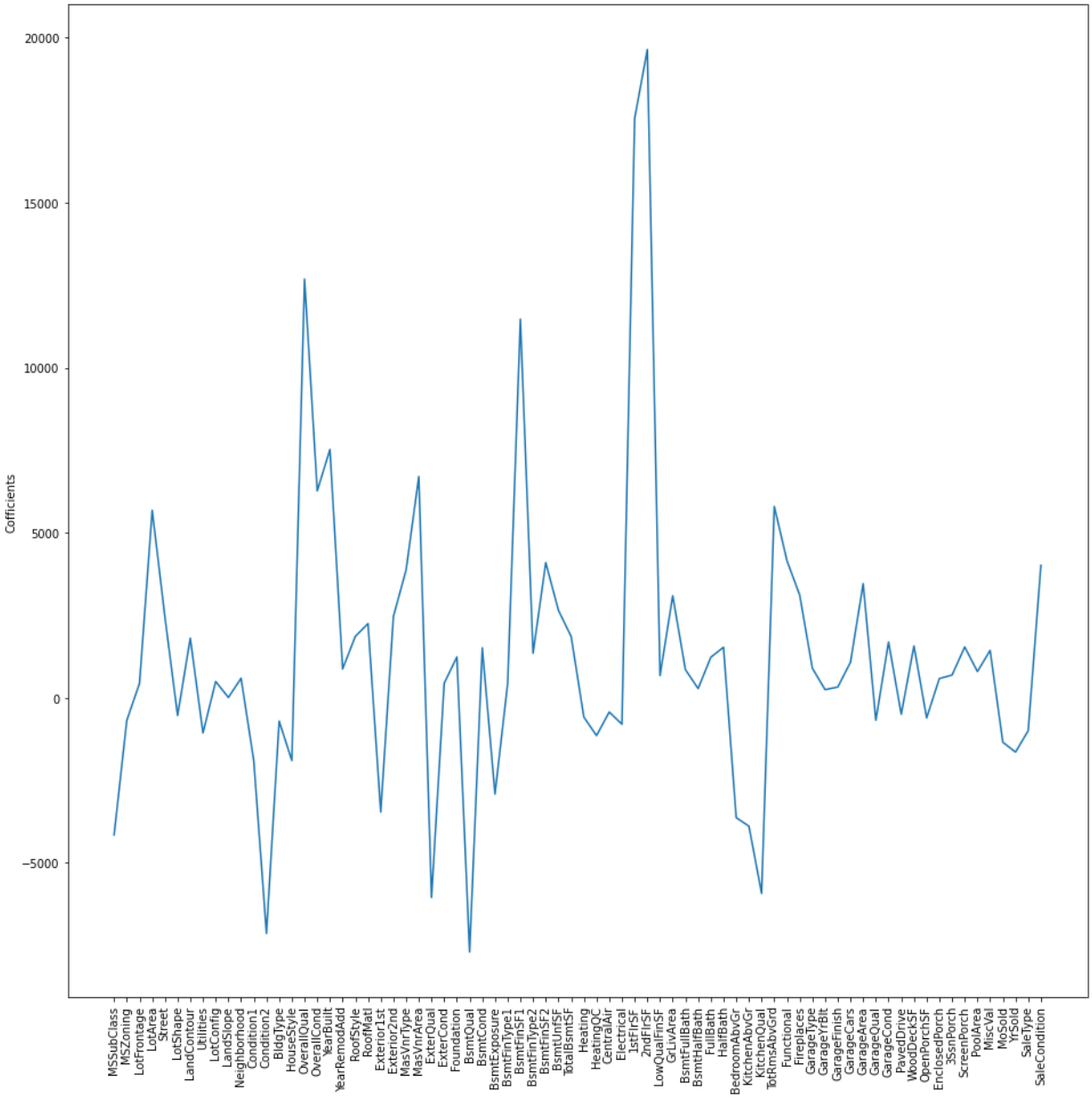
```
1 # Finding Most important Variable/Column
2 m=max(ls_coef)
3 print(m)
4 for i in range(len(col)):
5     if ls_coef[i]==m:
6         imp_pred=col[i]
7 print(imp_pred)
```

```
19633.34932598228
2ndFlrSF
```

There fore the **2ndFlrSE variable** is having the highest value for the coefficient's so it is most important variable in Lasso regression.

We can also see this from the graph plotted below for all the variables with coefficients for

Lasso Regression



Q2: 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?

Ans:

```
#Model training
rd=Ridge(alpha=100,random_state=0)
rd.fit(x_train_sc,y_train)
print(rd.score(x_test_sc,y_test))
rd.coef_
```

```
0.6689229839919186
array([[-3048.00706915,  -633.61982414,   758.87555606,  5098.66500945,
         2333.36919994,  -811.40646444,  1444.97867656, -1013.87772001,
          261.33654598,   170.94018532,   602.15336011, -1624.8558183 ,
        -5819.31798168, -1372.86841195, -1545.68720009, 11976.71828491,
         5155.19562748,  3998.07134353,  1816.92173716,  2044.11515645,
         2418.55831077, -2377.20339486,  1354.13063804,  3494.67410842,
         6595.70099945, -6211.81024196,   492.46664609,  1735.5522855 ,
        -7409.12403275,  1401.67783542, -2744.84880767,    52.12478705,
         6492.91278612,  1009.1544451 ,  1950.96337967, -1259.16303858,
         6282.33787243,  -349.54039811, -1449.96460026,   202.43885417,
         -716.44080032,  7823.26401736,  7700.72808695,  -415.11152155,
        12213.87938843,  1386.46532505,   310.23694634,  2656.57486218,
         2326.84217667, -2424.29467972, -3670.29409953, -5977.70973825,
         5847.61332841,  3757.6385649 ,  3664.32365719,   512.63276155,
          597.79783718,  -273.27736685,  2136.06218166,  3609.85090441,
         -503.82174006,  1345.73806933,   133.56923432,  1762.57387528,
          -93.66515322,   417.73168674,   612.28757262,  1520.71230466,
          738.9583497 ,  1040.87056182, -1191.90452529, -1449.29678181,
        -1003.86141122,  3701.52662245]])
```

```
[147] # Model training
ls=Lasso(alpha=0.01,random_state=0)
ls.fit(x_train_sc,y_train)
print(ls.score(x_test_sc,y_test))
ls.coef_
```

```
0.628856519803142
array([-4.14969279e+03, -6.82184558e+02,  4.37084965e+02,  5.67946682e+03,
        2.45364283e+03, -5.35142402e+02,  1.80759627e+03, -1.06379165e+03,
        4.94929241e+02,  1.12822846e+01,  5.92438585e+02, -1.90137459e+03,
       -7.13392577e+03, -7.05402256e+02, -1.90111075e+03,  1.26853161e+04,
        6.26820335e+03,  7.52303891e+03,  8.76746106e+02,  1.86249315e+03,
        2.24730411e+03, -3.46012889e+03,  2.46571461e+03,  3.87454255e+03,
        6.70397208e+03, -6.04906866e+03,  4.43655011e+02,  1.23690399e+03,
       -7.69840296e+03,  1.51413484e+03, -2.91573497e+03,  4.43337996e+02,
        1.14721368e+04,  1.35179548e+03,  4.09338075e+03,  2.64941480e+03,
        1.85616894e+03, -5.83377757e+02, -1.14430898e+03, -4.28243024e+02,
       -8.00632291e+02,  1.75479430e+04,  1.96340274e+04,  6.75624363e+02,
        3.09104388e+03,  8.53773543e+02,  2.80554549e+02,  1.22934577e+03,
        1.53231988e+03, -3.62911694e+03, -3.88719185e+03, -5.92079759e+03,
        5.79812094e+03,  4.15091174e+03,  3.11341742e+03,  8.95790035e+02,
        2.47977738e+02,  3.26832664e+02,  1.07518722e+03,  3.45588121e+03,
       -6.80178840e+02,  1.69193547e+03, -4.96607181e+02,  1.57313034e+03,
       -6.07768331e+02,  5.83982843e+02,  6.89606768e+02,  1.54035109e+03,
        7.95407741e+02,  1.43694588e+03, -1.34368668e+03, -1.64382919e+03,
       -9.94218592e+02,  4.01371907e+03])
```

From both models we can see that the accuracy of the Ridge Regression is more than the accuracy of the Lasso regression.
Therefore we will apply Ridge regression during the assignment.

Q3:

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?

Ans:

As we can see below in both the average accuracy(84%) for the Model is less than the accuracy obtained from training the model above(66%) Which implies either the model is underfitted or the relation B/W Dependent and Independent variable iss not linear As we know that some models like Random forest and Decision Tree Regressions were more robust and Generalised.
However we can make Lasso and Ridge more robust and generalised by removing outliers but this may not guarantee that the model will become more robust and generalised as the relation B/W dependent and independent variables is may not be Linear.

As we can see below in both the average accuracy(84%) for the Model is less than the accuracy obtained from training the model above(66%) Which implies either the model is underfitted or the relation B/W Dependent and Independent variable is not linear.

Cross Validation of Ridge Regression

```
[ ]  1 from sklearn.model_selection import cross_val_score
     2 accuracies_ridge= cross_val_score(estimator = rd, X = x_train_sc, y = y_train_sc, cv = 10)
     3 print('Accuracy={:f} and Standard Deviation={:f}'.format(accuracies_ridge.mean()*100,accuracies_ridge.std()*100))
     4 accuracies_ridge

Accuracy=84.582197 and Standard Deviation=8.776072
array([0.86830144, 0.87185901, 0.59075023, 0.8688077 , 0.91949336,
       0.87578847, 0.85688663, 0.83544484, 0.90204007, 0.86884796])
```

Cross Validation of Lasso Regression

```
     1 from sklearn.model_selection import cross_val_score
     2 accuracies_lasso= cross_val_score(estimator = ls, X = x_train_sc, y = y_train_sc, cv = 10)
     3 print('Accuracy={:f} and Standard Deviation={:f}'.format(accuracies_lasso.mean()*100,accuracies_lasso.std()*100))
     4 accuracies_lasso

Accuracy=84.209085 and Standard Deviation=9.577517
array([0.86677455, 0.87081993, 0.5603117 , 0.86211344, 0.91608695,
       0.88066694, 0.85828795, 0.84273466, 0.89028769, 0.87282465])
```