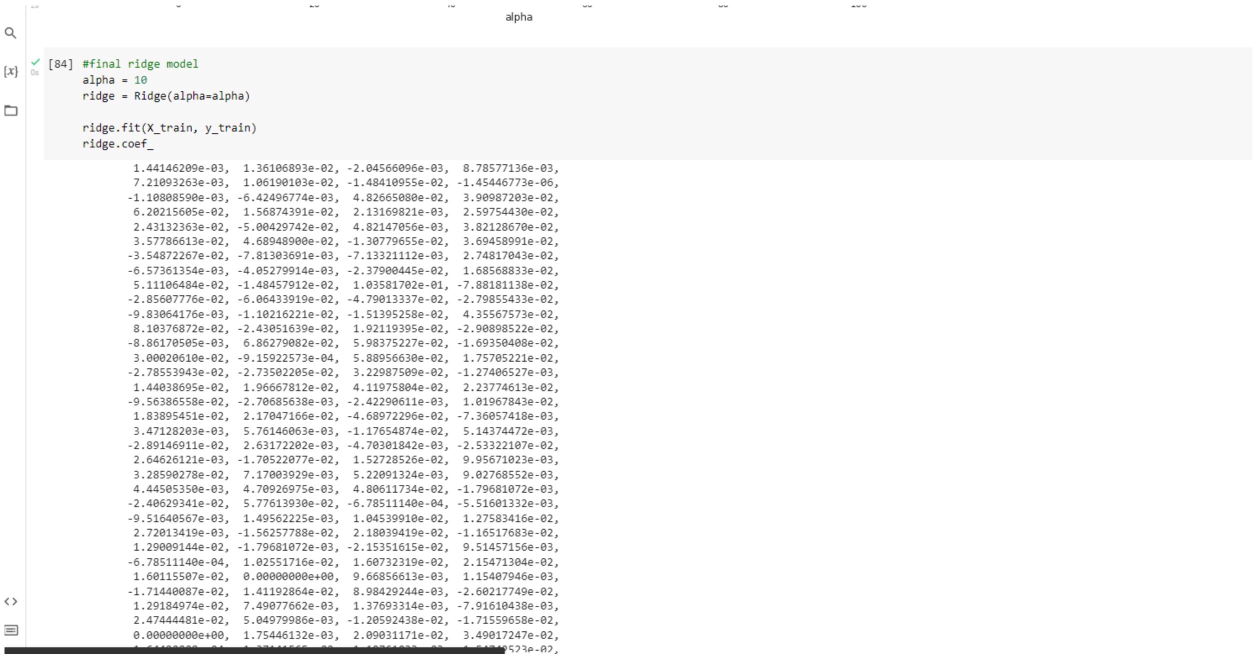
1. What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer -

The optimal value of alpha for ridge and lasso regression

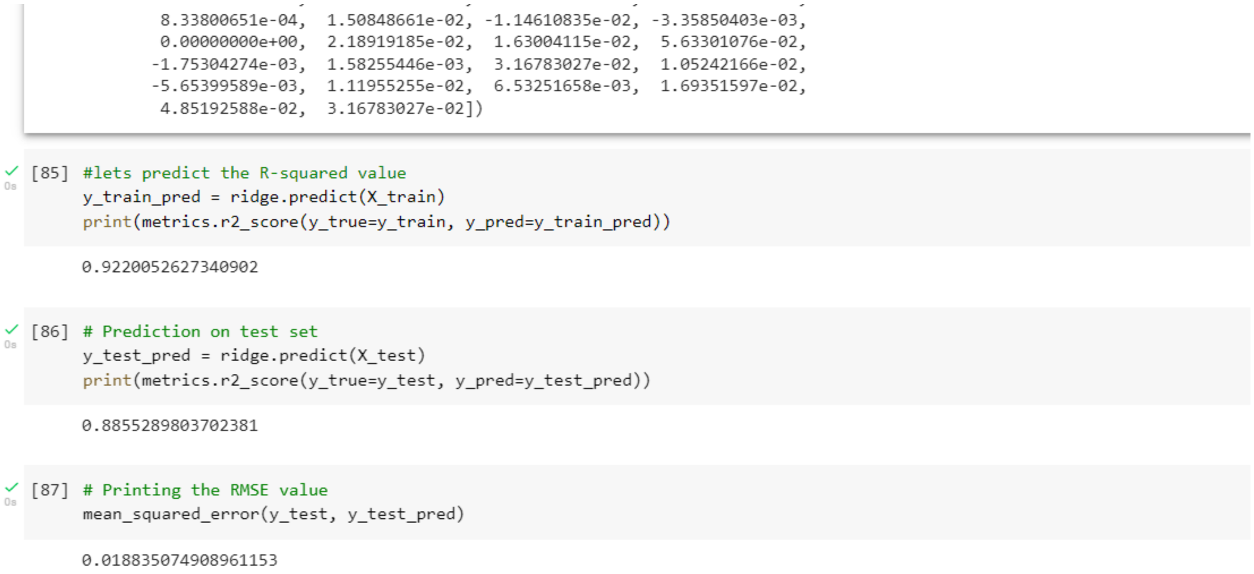
Ridge Alpha 1

lasso Alpha 10



```
[84] #final ridge model
     alpha = 10
     ridge = Ridge(alpha=alpha)

     ridge.fit(X_train, y_train)
     ridge.coef_
```

```
        1.44146209e-03,  1.36106893e-02, -2.04566096e-03,  8.78577136e-03,
        7.21093263e-03,  1.06190103e-02, -1.48410955e-02, -1.45446773e-06,
       -1.10808590e-03, -6.42496774e-03,  4.82665080e-02,  3.90987203e-02,
        6.20215605e-02,  1.56874391e-02,  2.13169821e-03,  2.59754430e-02,
        2.43132363e-02, -5.00429742e-02,  4.82147056e-03,  3.82128670e-02,
        3.57786613e-02,  4.68948900e-02, -1.30779655e-02,  3.69458991e-02,
       -3.54872267e-02, -7.81303691e-03, -7.13321112e-03,  2.74817043e-02,
       -6.57361354e-03, -4.05279914e-03, -2.37900445e-02,  1.68568833e-02,
        5.11106484e-02, -1.48457912e-02,  1.03581702e-01, -7.88181138e-02,
       -2.85607776e-02, -6.06433919e-02, -4.79013337e-02, -2.79855433e-02,
       -9.83064176e-03, -1.10216221e-02, -1.51395258e-02,  4.35567573e-02,
        8.10376872e-02, -2.43051639e-02,  1.92119395e-02, -2.90898522e-02,
       -8.86170505e-03,  6.86279082e-02,  5.98375227e-02, -1.69350408e-02,
        3.00020610e-02, -9.15922573e-04,  5.88956630e-02,  1.75705221e-02,
       -2.78553943e-02, -2.73502205e-02,  3.22987509e-02, -1.27406527e-03,
        1.44038695e-02,  1.96667812e-02,  4.11975804e-02,  2.23774613e-02,
       -9.56386558e-02, -2.70685638e-03, -2.42290611e-03,  1.01967843e-02,
        1.83895451e-02,  2.17047166e-02, -4.68972296e-02, -7.36057418e-03,
        3.47128203e-03,  5.76146063e-03, -1.17654874e-02,  5.14374472e-03,
       -2.89146911e-02,  2.63172202e-03, -4.70301842e-03, -2.53322107e-02,
        2.64626121e-03, -1.70522077e-02,  1.52728526e-02,  9.95671023e-03,
        3.28590278e-02,  7.17003929e-03,  5.22091324e-03,  9.02768552e-03,
        4.44505350e-03,  4.70926975e-03,  4.80611734e-02, -1.79681072e-03,
       -2.40629341e-02,  5.77613930e-02, -6.78511140e-04, -5.51601332e-03,
       -9.51640567e-03,  1.49562225e-03,  1.04539910e-02,  1.27583416e-02,
        2.72013419e-03, -1.56257788e-02,  2.18039419e-02, -1.16517683e-02,
        1.29009144e-02, -1.79681072e-03, -2.15351615e-02,  9.51457156e-03,
       -6.78511140e-04,  1.02551716e-02,  1.60732319e-02,  2.15471304e-02,
        1.60115507e-02,  0.00000000e+00,  9.66856613e-03,  1.15407946e-03,
       -1.71440087e-02,  1.41192864e-02,  8.98429244e-03, -2.60217749e-02,
        1.29184974e-02,  7.49077662e-03,  1.37693314e-03, -7.91610438e-03,
        2.47444481e-02,  5.04979986e-03, -1.20592438e-02, -1.71559658e-02,
        0.00000000e+00,  1.75446132e-03,  2.09031171e-02,  3.49017247e-02,
```

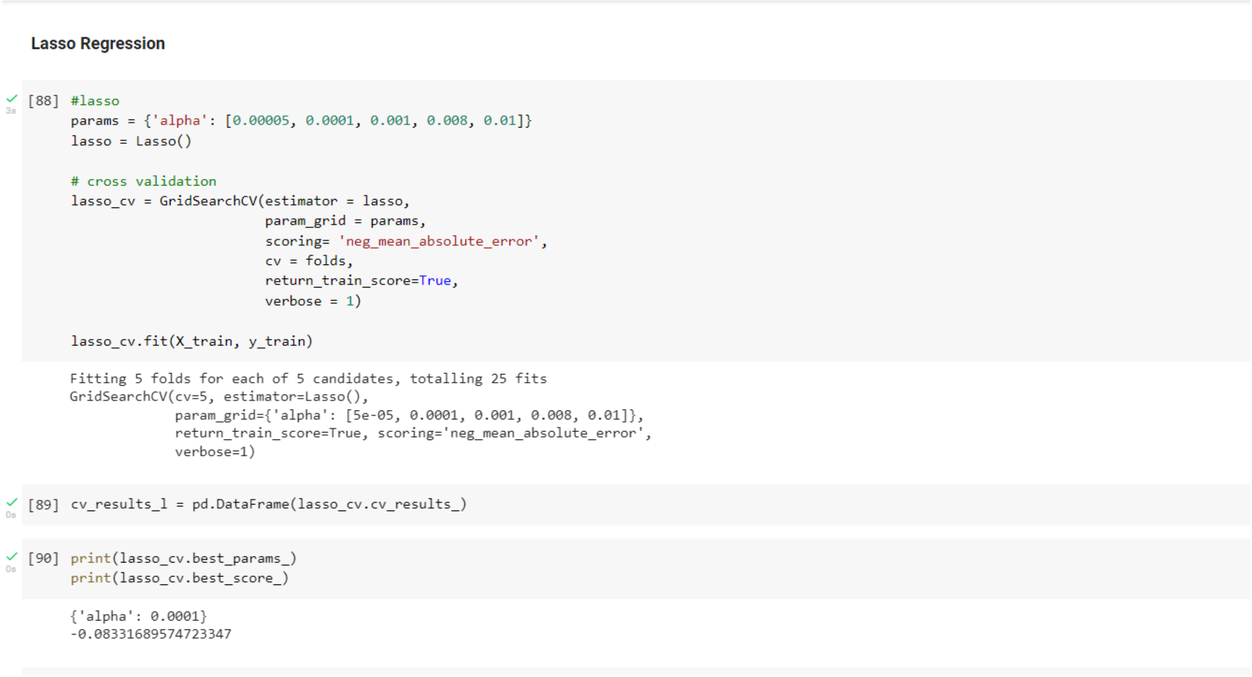


```
             8.33800651e-04,  1.50848661e-02, -1.14610835e-02, -3.35850403e-03,
             0.00000000e+00,  2.18919185e-02,  1.63004115e-02,  5.63301076e-02,
            -1.75304274e-03,  1.58255446e-03,  3.16783027e-02,  1.05242166e-02,
            -5.65399589e-03,  1.11955255e-02,  6.53251658e-03,  1.69351597e-02,
             4.85192588e-02,  3.16783027e-02])
```

```
[85] #lets predict the R-squared value
     y_train_pred = ridge.predict(X_train)
     print(metrics.r2_score(y_true=y_train, y_pred=y_train_pred))

     0.9220052627340902
```

```
[86] # Prediction on test set
     y_test_pred = ridge.predict(X_test)
     print(metrics.r2_score(y_true=y_test, y_pred=y_test_pred))

     0.8855289803702381
```

```
[87] # Printing the RMSE value
     mean_squared_error(y_test, y_test_pred)

     0.018835074908961153
```



Lasso Regression

```
[88] #lasso
     params = {'alpha': [0.00005, 0.0001, 0.001, 0.008, 0.01]}
     lasso = Lasso()

     # cross validation
     lasso_cv = GridSearchCV(estimator = lasso,
                             param_grid = params,
                             scoring= 'neg_mean_absolute_error',
                             cv = folds,
                             return_train_score=True,
                             verbose = 1)

     lasso_cv.fit(X_train, y_train)

     Fitting 5 folds for each of 5 candidates, totalling 25 fits
     GridSearchCV(cv=5, estimator=Lasso(),
                  param_grid={'alpha': [5e-05, 0.0001, 0.001, 0.008, 0.01]},
                  return_train_score=True, scoring='neg_mean_absolute_error',
                  verbose=1)
```

```
[89] cv_results_l = pd.DataFrame(lasso_cv.cv_results_)
```

```
[90] print(lasso_cv.best_params_)
     print(lasso_cv.best_score_)

     {'alpha': 0.0001}
     -0.08331689574723347
```

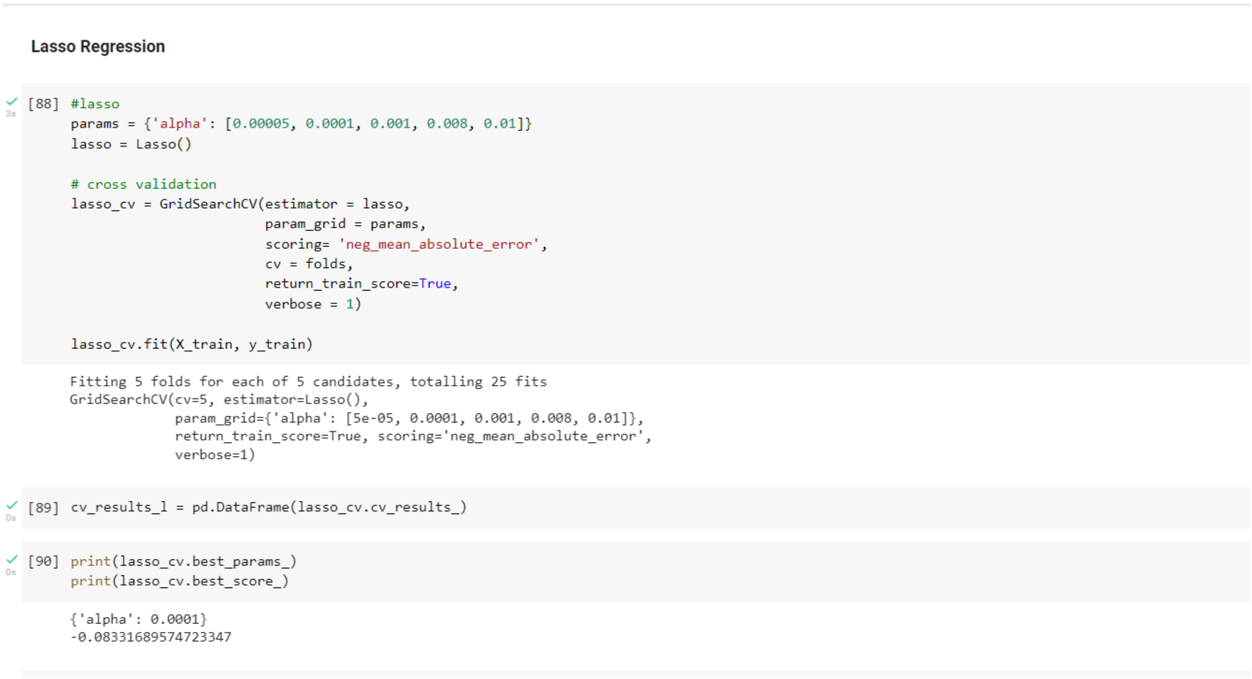
2. You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answer -

The r2_score of lasso is slightly higher than lasso for the test dataset so we will choose lasso regression to solve this problem.

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 -



Lasso Regression

```
[88] #lasso
     params = {'alpha': [0.00005, 0.0001, 0.001, 0.008, 0.01]}
     lasso = Lasso()

     # cross validation
     lasso_cv = GridSearchCV(estimator = lasso,
                             param_grid = params,
                             scoring= 'neg_mean_absolute_error',
                             cv = folds,
                             return_train_score=True,
                             verbose = 1)

     lasso_cv.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
GridSearchCV(cv=5, estimator=Lasso(),
             param_grid={'alpha': [5e-05, 0.0001, 0.001, 0.008, 0.01]},
             return_train_score=True, scoring='neg_mean_absolute_error',
             verbose=1)
```

```
[89] cv_results_l = pd.DataFrame(lasso_cv.cv_results_)
```

```
[90] print(lasso_cv.best_params_)
     print(lasso_cv.best_score_)
```

```
{'alpha': 0.0001}
-0.08331689574723347
```

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 -
The model should be generalized so that the test accuracy is not lesser than the training score. The model should be accurate for datasets other than the ones which were used during training. Too much importance should not given to the outliers so that the accuracy predicted by the model is high. To ensure that this is not the case, the outliers analysis needs to be done and only those which are relevant to the dataset need to be retained. Those outliers which it does not make sense to keep must be removed from the dataset. If the model is not robust, It cannot be trusted for predictive analysis.