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 To find the optimal value of alpha for ridge and lasso regression we tune the hyperparameters using KFold and GridSearchCV.

KFold splits the data (train/test) to k consecutive folds where k-1 folds can be used for training set and 1 set can be used for validation

```
# set up cross validation scheme
folds = KFold(n_splits = 5, shuffle = True, random_state = 4)
```

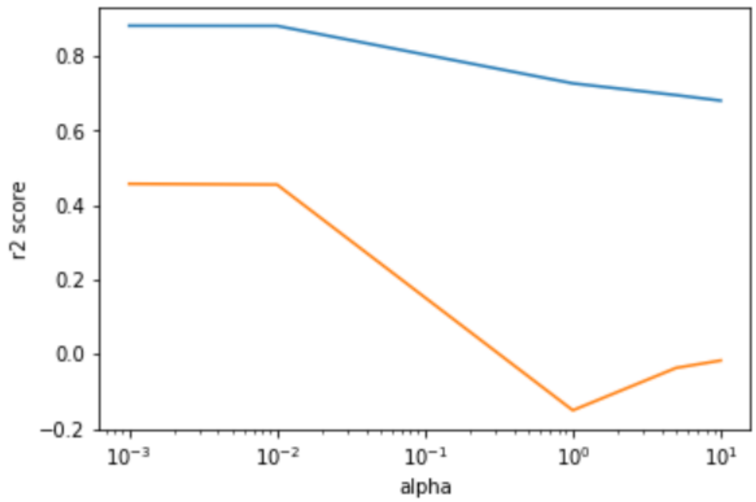
GridSearchCV is used to tune the hyperparameters

```
# specify range of hyperparameters
params = {'alpha': [0.001, 0.01, 1.0, 5.0, 10.0]}

# grid search
# lasso model
model = Lasso()
model_cv = GridSearchCV(estimator = model, param_grid = params,
                        scoring= 'r2',
                        cv = folds,
                        return_train_score=True, verbose = 1)
model_cv.fit(X_train, y_train)
```

In gridsearchcv we pass the different values of the hyperparameter to tune. In the attached image we are passing different values of alpha as 0.001, 0.01, 1, 5, 10
By plotting the results of params and mean score of the test and train data

```
# plot
cv_results['param_alpha'] = cv_results['param_alpha'].astype('float32')
plt.plot(cv_results['param_alpha'], cv_results['mean_train_score'])
plt.plot(cv_results['param_alpha'], cv_results['mean_test_score'])
plt.xlabel('alpha')
plt.ylabel('r2 score')
plt.xscale('log')
plt.show()
```



We determine the optimal value of alpha as the value for which the train data increases first.
In the attached image the optimal value of alpha is between 0.001 and 0.01 as the graph line is constant between these values and then it takes a dip.

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?

A2 Choosing Lasso over Ridge regression would be better option. Since lasso performs the feature selection by making the coefficients to 0.

Lasso stands for Least Absolute Shrinkage Selector Operator. It adds the penalty of absolute

Value of the magnitude of coefficients and shrinkage the not required features.

A small change in the value of alpha can reduce the magnitude of coefficients.

Q3. 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?

A3. If using lasso the five important predictor variables are selected and are not available in the incoming data, then selecting the other most five predictor variable we can use the recursive feature selection (RFE)
Steps to select features using RFE

Step 8: Feature Selection Using RFE

[40]:
```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
```

[41]:
```
from sklearn.feature_selection import RFE
rfe = RFE(logreg, 15)             # running RFE with 13 variables as output
rfe = rfe.fit(X_train, y_train)
```

[42]:
```
rfe.support_
```

[42]:
```
array([ True,  True,  True, False,  True,  True, False, False,  True,
        True,  True, False,  True, False,  True,  True,  True,  True,
       False, False,  True,  True, False])
```

Q4. 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?

A4. Robustness means that the model tested on training sample and on a similar sample the performance in both the sample is same.

By making the model more robustness and gernalisable the accuracy of the model on both the training and test data would be close and there won't be much difference.
But by making the model more complex the model will perform good on the training data and while executing the model on the testing data the results accuracy won't be close to the accuracy obtained on the training data.

Since by making the model complex the model overfits on the training data and it will have more accuracy.