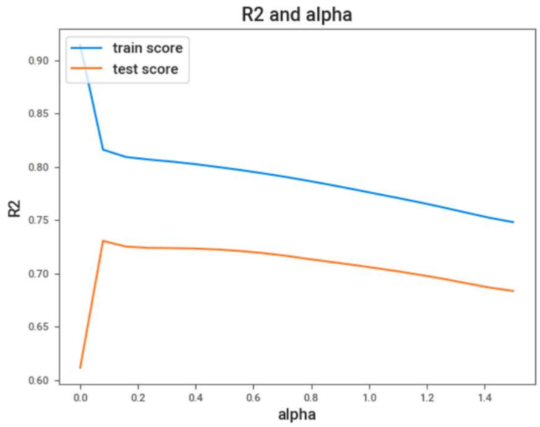
Assignment-based Subjective Questions

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?

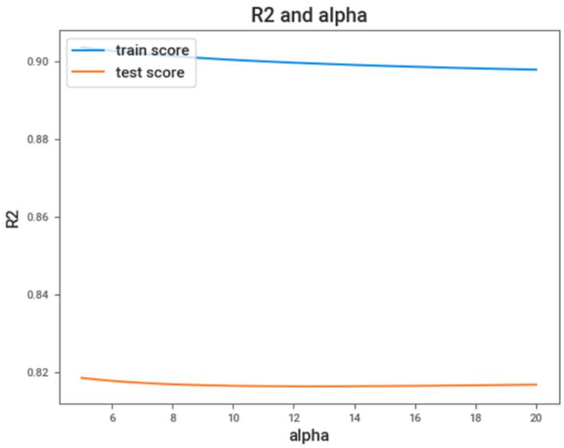
The optimal value of alpha for ridge regression is 9 and for lasso regression it is 0.5.

R2 scores for different alpha ranges in lasso regression:



If we double the value of alpha in lasso, the r2 score tend to decrease but not by huge margin. The model will underfit more.

R2 scores for different alpha ranges in ridge regression:



If we double the value of alpha in ridge, we see a similar result as in ridge. That is, the r2 score tend to decrease but not by huge margin. The model will underfit more.

The most important predictor variables after we double the value of alpha in ridge regression:

alpha=9

```
Variables: ['GarageCars', 'CentralAir', 'BsmtFullBath', 'OverallQual'
, 'PoolQC']
Coefficients: [0.0509669  0.05141762 0.05821884 0.06628209 0.23701869
]
```

alpha=18

```
Variables: ['OverallCond', 'GarageCars', 'BsmtFullBath', 'OverallQual'
, 'PoolQC']
Coefficients: [0.04726188 0.04901178 0.05526987 0.06632917 0.13135277]
```

The most important predictor variables after we double the value of alpha in lasso regression:

alpha=0.5

```
Variables: ['GarageArea', 'MSSubClass', 'PoolArea', 'YearRemodAdd', 'Y
earBuilt']

Coefficients: [0.00033836 0.00033934 0.00126965 0.0022709  0.00290887]
```

alpha=1.0

```
Variables: ['GarageArea', 'MSSubClass', 'PoolArea', 'YearRemodAdd', 'Y
earBuilt']
Coefficients: [0.00033836 0.00033934 0.00126965 0.0022709  0.00290887]
```

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?

I will choose lasso regression over ridge for the following reasons:

- Penalty in Lasso forces some of the coefficient estimates to be exactly equal to zero, thus helping to perform variable section. The variable selection is an advantage in Lasso over Ridge.
- In Ridge, when the lambda value is increased (doubled), the R2 score improved. However, it indicates that we are over regularising and underfitting the model.
- In Lasso however, when the lambda value is increased (doubled), the R2 score more or less remained the same which also indicated that we can work with the optimal lambda (0.5) identified during the experiments.

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?*

The five most important predictor variables in the lasso model at lambda=0.5 are:

```
['GarageArea', 'MSSubClass', 'PoolArea', 'YearRemodAdd', 'YearBuilt']
```

After excluding the above and rebuilding the Lasso model, the five most important predictor variables and the corresponding coefficient values are found to be as below:

```
['WoodDeckSF', 'ScreenPorch', '1stFlrSF', '2ndFlrSF', 'GarageYrBlt']
[0.00027839 0.00028584 0.00028596 0.00031108 0.00381989]
```

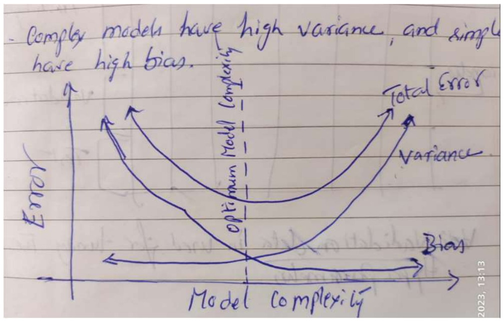
While the R2 for training dataset dropped, it improved for the test dataset.

```
Training R2
0.6956606101336146
Testing R2
0.715407678109186
```

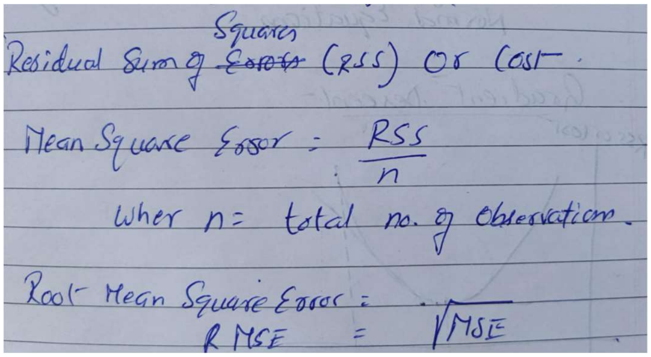
Question 4

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?

The central issue in all of machine learning is how to extrapolate the learnings from a finite amount of available data to all possible inputs of the same kind. Training data is always finite and the challenge is for the model to learn all about the task/usecase from it and perform well on the unseen data.

- As a first principle we should use Simpler models since they are usually “more generic” and require fewer training samples.
- We should avoid overfitting wherein a model does extremely well on training data but fails on the unseen data.
- We need to look at the trade-off between Bias and Variance while building the model.
- Complex models have high variance and simple models have high bias. We need to balance this.
- 

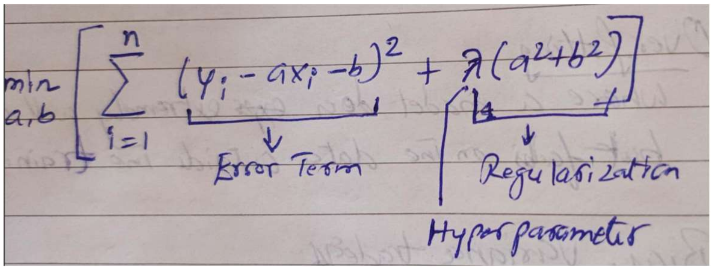
- Complex models have high variance, and simpl
have high bias.

- Below are metrics to assess to model performance and provides guidance to tune the models to be more general and robust:
- 

  Residual Sum of ~~Errors~~ Squares (RSS) Or Cost.

  Mean Square Error : $\frac{RSS}{n}$

  Wher n= total no. of Observation.

  Root Mean Square Error = RMSE = $\sqrt{MSE}$
- All these metrics give us a sense of overall error in the model and lower the values of these the better.
- We use Regularization and Hyperparameters to prevent the model from becoming complex. Regularization is part of the learning algorithm and Hyperparameters are the parameters we pass onto the learning algorithm to control the complexity of the final model.
- Input to the learning algorithm are Hyperparameters and class off algorithm. Output of the learning algorithm are the model parameters.
- Mathematical representation below:
- 

  $$\min_{a,b}\left[\underbrace{\sum_{i=1}^{n}(y_i - ax_i - b)^2}_{\text{Error Term}} + \underbrace{\lambda}_{\text{Hyperparameter}}\underbrace{(a^2+b^2)}_{\text{Regularization}}\right]$$
- There is always a trade-off between the model accuracy and the model complexity and robustness. High accuracy of the model on training data may not necessarily result in high accuracy on the test data due to overfitting. The above levers ensure we check how well model has learnt all the behaviour in the data such that it can generalize and predict the behaviour of the unseen data.