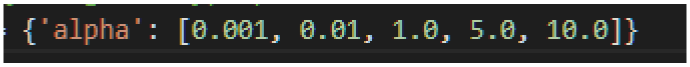
Question-1:

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?

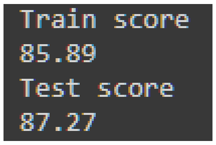
Answer:

Alpha range



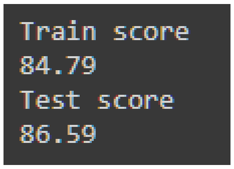
```
{'alpha': [0.001, 0.01, 1.0, 5.0, 10.0]}
```

As the range of alpha value increases the bias in the model also increases. The optimal value for Lasso is 0.001 as alpha where the test and the train score does not have huge difference and the test score is at highest value as below



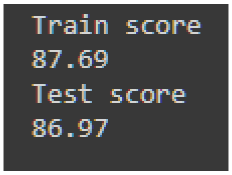
```
Train score
85.89
Test score
87.27
```

When we double the value of alpha as 0.002, we achieve a comparatively lower test score as the modal adds more bias



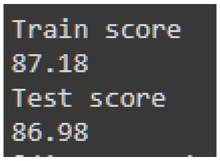
```
Train score
84.79
Test score
86.59
```

As the range of alpha value increases the bias in the model also increases. The optimal value for Ridge is 1 as alpha where the test and the train score does not have huge difference and the test score is at highest value as below



```
Train score
87.69
Test score
86.97
```

When we double the value of alpha as 2, we achieve a similar test score. The model shows decrease in performance as the value increases after alpha 3



```
Train score
87.18
Test score
86.98
```

The important predictor variables after making the changes for ridge and lasso are as follows

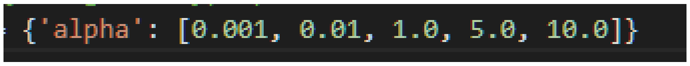
Lasso:
GrLivArea
TotalBsmtSF
Age
GarageArea
TtlBath

Ridge:
GrLivArea
TotalBsmtSF
Age
MSZoning_FV
Exterior1st_BrkComm

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?

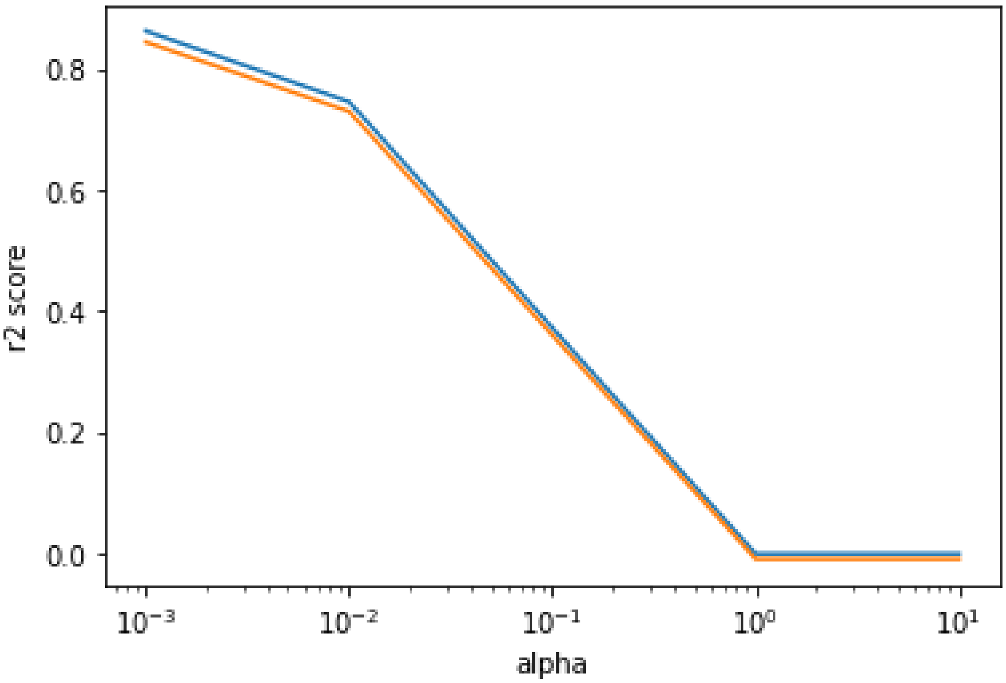
Answer:

Alpha range



```
{'alpha': [0.001, 0.01, 1.0, 5.0, 10.0]}
```

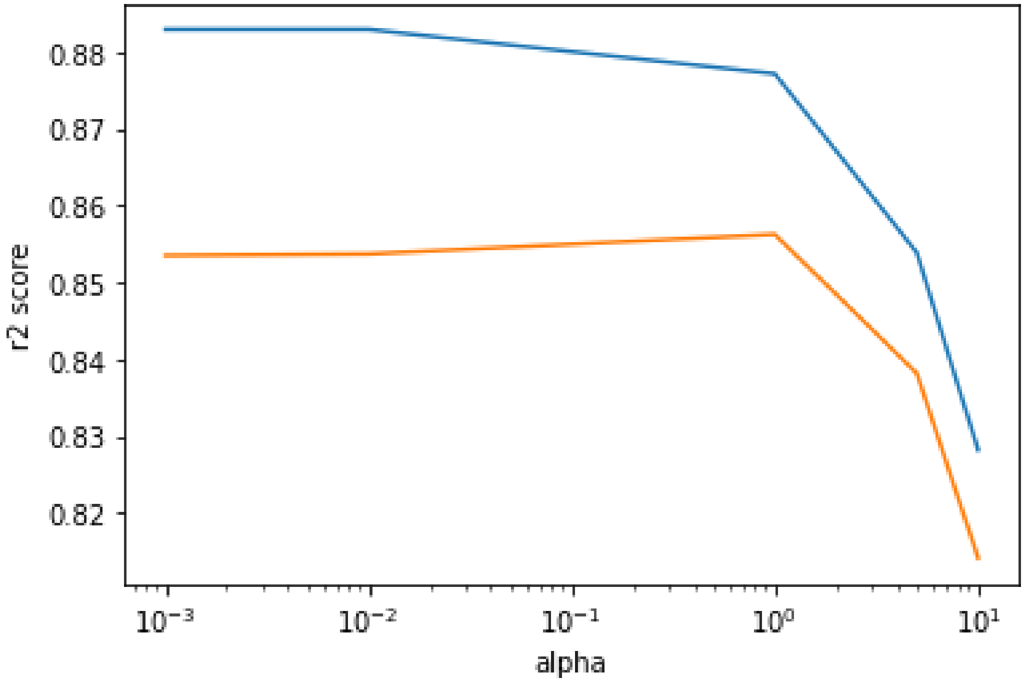
Lasso



The optimal value of alpha for Lasso is determined to be 0.001 which in the graph shows the highest train and test score without overfitting (Note: here test is just the validation data which is part of train data, the actual test scores would be different)

a point to note here is that as we increase the value of alpha/lambda the model becomes more general and the bias is very high but on the other hand the variance is getting low. We need to maintain the simplicity so that the model is not too naïve to predict test set.

Ridge



The optimal value of alpha for Ridge is determined to be 1 which in the graph shows the highest train and test score without overfitting (Note: here test is just the validation data which is part of train data, the actual test scores would be different). The R2 score is similar around alpha 1,2 and 3 but after that it starts decreasing.

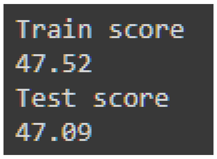
a point to note here is that as we increase the value of alpha/lambda the model becomes more general and the bias is very high but on the other hand the variance is getting low. We need to maintain the simplicity so that the model is not too naïve to predict test set.

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?

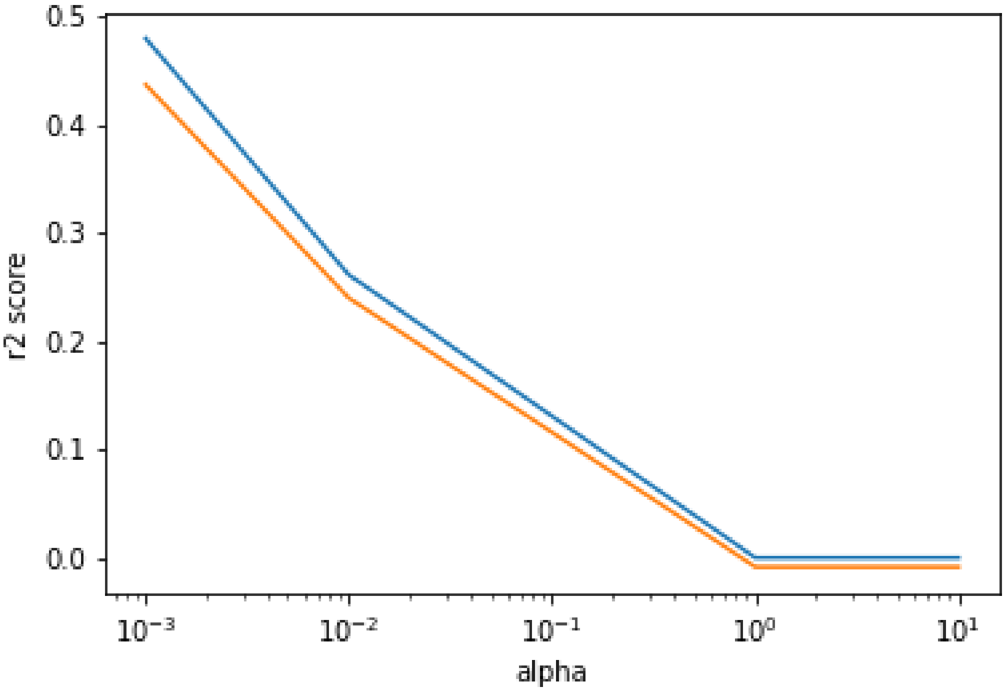
Answer:

After Removing the 5 most important variable which were GrLivArea, TotalBsmtSF, Age, GarageArea, TtlBath the model was built with again the same alpha value 0.001. The train and the test scores had drastic decrease as follows



```
Train score
47.52
Test score
47.09
```

This is expected because the variable that explains the variation of the model most were excluded.It still has a similar pattern for alpha values



The new 5 most important variables are as below

LotArea

TtlPorchArea

Neighborhood_NridgHt

MSZoning_FV

Neighborhood_StoneBr

Note:- few of these new important variables are features created out of raw variables for example TtlPorchArea which is Total Porch area is a sum of OpenPorchSF+ EnclosedPorch+ 3SsnPorch+ ScreenPorch

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?

Answer:
Ideally, you want to select a model at the sweet spot between underfitting and overfitting. This is the goal, but is very difficult to do in practice. To understand this goal, we can look at the performance of a machine learning algorithm over time as it is learning a training data. We can plot both the skill on the training data and the skill on a test dataset we have held back from the training process. Over time, as the algorithm learns, the error for the model on the training data goes down and so does the error on the test dataset. If we train for too long, the performance on the training dataset may continue to decrease because the model is overfitting and learning the irrelevant detail and noise in the training dataset. At the same time the error for the test set starts to rise again as the model's ability to generalize decreases.

The sweet spot is the point just before the error on the test dataset starts to increase where the model has good skill on both the training dataset and the unseen test dataset. You can perform this experiment with your favourite machine learning algorithms. This is often not useful technique in practice, because by choosing the stopping point for training using the skill on the test dataset it means that the test set is no longer "unseen" or a standalone objective measure. Some knowledge (a lot of useful knowledge) about that data has leaked into the training procedure.

There are two additional techniques you can use to help find the sweet spot in practice: resampling methods and a validation dataset. Both overfitting and underfitting can lead to poor model performance. But by far the most common problem in applied machine learning is overfitting.

Overfitting is such a problem because the evaluation of machine learning algorithms on training data is different from the evaluation we actually care the most about, namely how well the algorithm performs on unseen data.

There are two important techniques that you can use when evaluating machine learning algorithms to limit overfitting:

Use a resampling technique to estimate model accuracy.
Hold back a validation dataset.
The most popular resampling technique is k-fold cross validation. It allows you to train and test your model k-times on different subsets of training data and build up an estimate of the performance of a machine learning model on unseen data.

A validation dataset is simply a subset of your training data that you hold back from your machine learning algorithms until the very end of your project. After you have selected and tuned your machine learning algorithms on your training dataset you can evaluate the learned models on the validation dataset to get a final objective idea of how the models might perform on unseen data.

Using cross validation is a gold standard in applied machine learning for estimating model accuracy on unseen data. If you have the data, using a validation dataset is also an excellent practice.
The Advance regression techniques such as Lasso and ridge helps to generalize the model