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?

i) What is the optimal value of alpha for ridge and lasso regression?

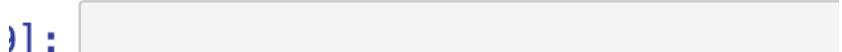
Ans : The optimal value for alpha for ridge Regression is 10 and for Lasso Regression is .0002 , Please find below screenshot for reference

Ridge Regression ¶

- Optimal lambda value : 10
- R2 Score Train : .9227 ,
- R2 Test Score : .8917
- RMSE test is .133596

Lasso Regression

- Optimal lambda value : .0002
- R2 Score Train : .926494
- R2 Test Score : .8945
- RMSE test is .131860



ii) What will be the changes in the model if you choose double the value of alpha for both ridge and lasso?

Ans : if we choose to double the value of alpha

For Ridge Regression : alpha 20

For Lasso Regression : alpha .0004

Ridge Regression :

R2 Score Train will decrease from .922 to .919

R2 Score Test will decrease from .8917 to .8912

RMSE value will increase from .1335 to .1338

Lasso Regression :

R2 Score Train will decrease from .926494 to .921956

R2 Score Test will decrease from .894 to .893

In [107]:
```
# Compare the different metrics for Linear Regression, Ridge Regression and Lassoregression
metric_column = pd.Series(["R2 Score (train)","R2 Score (test)","RMSE (train)","RMSE (test)"], name = "Metric")
Linear_Metric = pd.Series(LinearRegMetric, name = "LinearRegression")
ridge_metric = pd.Series(ridgemetric, name = "RidgeRegression")
lasso_metric = pd.Series(lassometric, name = "LassoRegression")
ridge_metric_doublealplha = pd.Series(ridgemetric_updatedalpha, name = "RidgeRegression_doublealpha")
lasso_metric_doublealplha = pd.Series(lassometric_updated, name = "LassoRegression_doublealpha")
finalmetric = pd.concat([metric_column,Linear_Metric,ridge_metric,ridge_metric_doublealplha,lasso_metric,lasso_metri
finalmetric
```

Out[107]:

	Metric	LinearRegression	RidgeRegression	RidgeRegression_doublealpha	LassoRegression	LassoRegression_doublealpha
0	R2 Score (train)	0.900911	0.922734	0.919564	0.926494	0.921956
1	R2 Score (test)	0.874180	0.891708	0.891272	0.894503	0.893740
2	RMSE (train)	0.124784	0.110190	0.112427	0.107475	0.110743
3	RMSE (test)	0.144002	0.133596	0.133865	0.131860	0.132337

RMSE value will increase from .1318 to .1323

iii) What will be the most important predictor variables after the change is implemented?

Answer : Observation Top features as well as coefficients are changed

Ridge Regression top 10 Predictors

Params	Coef
GrLivArea	0.078987
OverallQual	0.077281
Neighborhood_Crawfor	0.066905
TotalBsmtSF	0.054827
OverallCond	0.051468
Neighborhood_Somerst	0.050939
Exterior1st_BrkFace	0.050807
Neighborhood_NridgHt	0.047474
MSZoning_FV	0.044708
SaleCondition_Partial	0.041179
SaleCondition_Normal	0.040688

Lasso Regression top 10 Predictors

Params	Coef
GrLivArea	0.102074
Neighborhood_Crawfor	0.099553
Exterior1st_BrkFace	0.088063
MSZoning_FV	0.083078
Neighborhood_NridgHt	0.080935
Neighborhood_Somerst	0.080603
MSZoning_RL	0.078364
Neighborhood_StoneBr	0.077625
OverallQual	0.074078
Neighborhood_NoRidge	0.065654
Neighborhood_ClearCr	0.058840

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 : Optimal value of alpha for Ridge and Lasso Regression are determined based on the GridSearchCv and KFold Cross Validation , we have used Hyper Parameter Tuning in order to

extract these values Also we have Plotted the Graph between alpha and mean r2 score to analyse the values by validating Cross Validation results

The optimal value for alpha for ridge Regression is 10 and for Lasso Regression is .0002

```
#checking the value of optimum number of parameters
print(model_cv.best_params_)
print(model_cv.best_score_)
```

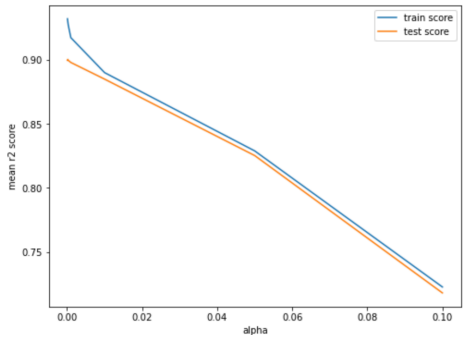
```
#result of cross validation
lasso_cv_res = pd.DataFrame(model_cv.cv_results_)
lasso_cv_res['param_alpha'] =
lasso_cv_res['param_alpha'].astype('float32')
lasso_cv_res.head(5)
```

t[84]:

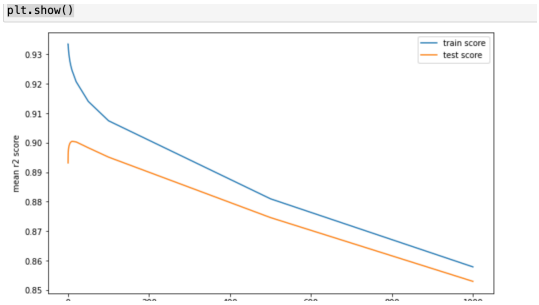
	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_alpha	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score
0	0.007287	0.002360	0.002578	0.000257	0.0001	{'alpha': 0.0001}	0.905235	0.873014	0.867147	0.906739
1	0.005051	0.000150	0.002314	0.000368	0.001	{'alpha': 0.001}	0.905257	0.873020	0.867157	0.906755
2	0.004430	0.000269	0.002153	0.000302	0.01	{'alpha': 0.01}	0.905458	0.873078	0.867248	0.906926
3	0.004681	0.000124	0.002258	0.000280	0.05	{'alpha': 0.05}	0.906193	0.873296	0.867522	0.907813
4	0.004611	0.000254	0.002180	0.000181	0.1	{'alpha': 0.1}	0.906900	0.873510	0.867685	0.908935

[85]:
```
ridge_cv_res['param_alpha'] = ridge_cv_res['param_alpha'].astype('float32')
```

Ridge Regression



Lasso Regression



However all the models are showing almost close R2_score but can see the model R2_Score for lasso regression is slightly better so we will go with it

	Metric	LinearRegression	RidgeRegression	LassoRegression
0	R2 Score (train)	0.900911	0.922734	0.926494
1	R2 Score (test)	0.874180	0.891708	0.894503
2	RMSE (train)	0.124784	0.110190	0.107475
3	RMSE (test)	0.144002	0.133596	0.131860

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?

5 imp predictors variables in the original lasso model are

MSZoning_FV,

MSZoning_RL

MSZoning_RH

MSZoning_RM

Neighborhood_Crawfor

After Dropping above 5 variables the five more import predictor variables now are

Lasso

In [125]:
```
# List of Top 5 Lasso Regression Coeffiecints

lasso_params_df = pd.DataFrame({'Params':cols,'Coef':lasso.coef_})
(lasso_params_df.sort_values('Coef',ascending=False)).head(6)
```

Out[125]:

	Params	Coef
92	Exterior1st_BrkFace	0.098240
15	GrLivArea	0.096572
71	Neighborhood_StoneBr	0.090722
70	Neighborhood_Somerst	0.088600
65	Neighborhood_NridgHt	0.082192
2	OverallQual	0.070930

Ridge

Params	Coef
GrLivArea	0.084374
OverallQual	0.074766
Neighborhood_Somerst	0.069907
Exterior1st_BrkFace	0.068312
Neighborhood_NridgHt	0.062062

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 : A model is considered to be robust and generalisable when the model is not overfitting and underfitting

Overfitting : High variance Low bias

In Linear Regression we may get the good accuracy on the model on training data but it may be overfitting hence it won't be accurate on the testing set and we can observer error in training and testing score . If the model is not robust and generalisable the accuracy of the model won't be good and we can't rely on the model

To avoid such overfitting issues we will be using Regularisations so that it will reduce the variance by compromising little bias

which will make model robust and score for training and test data will be near