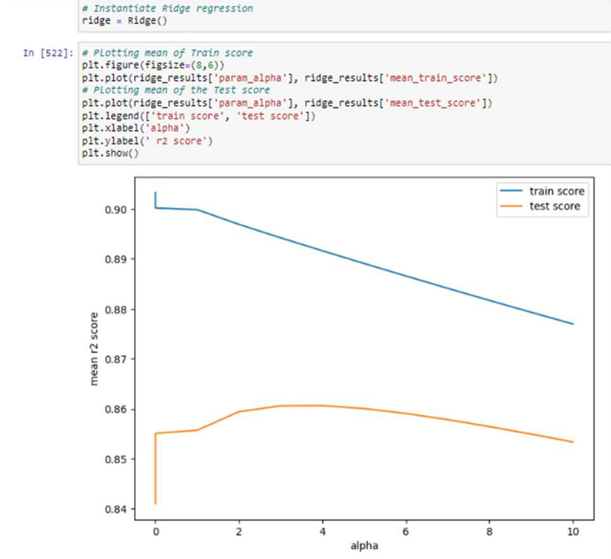
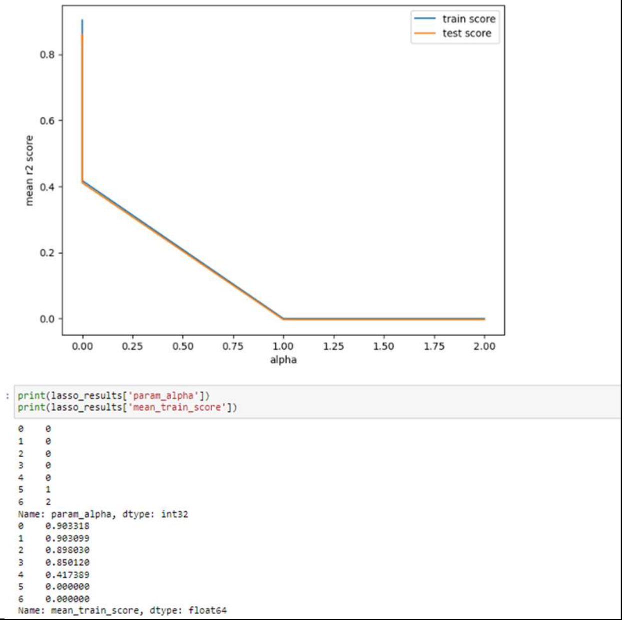
Assignment Part-II

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

Basis the assignment of house price prediction, optimal value of alpha for ridge and lasso regression is below.

RIDGE	LASSO
	

```
# Instantiate Ridge regression
ridge = Ridge()

In [522]: # Plotting mean of Train score
plt.figure(figsize=(8,6))
plt.plot(ridge_results['param_alpha'], ridge_results['mean_train_score'])
# Plotting mean of the Test score
plt.plot(ridge_results['param_alpha'], ridge_results['mean_test_score'])
plt.legend(['train score', 'test score'])
plt.xlabel('alpha')
plt.ylabel(' r2 score')
plt.show()
```

```
print(lasso_results['param_alpha'])
print(lasso_results['mean_train_score'])
```

```
0    0
1    0
2    0
3    0
4    0
5    1
6    2
Name: param_alpha, dtype: int32
0    0.903318
1    0.903099
2    0.898030
3    0.850120
4    0.417389
5    0.000000
6    0.000000
Name: mean_train_score, dtype: float64
```

the graph says lower the value of lambda (close to 0) , higher the accuracy. here it is coincidentally same

Train Score

r2 score goes down when the lambda value increase means the error is increasing. Model is overfitting and generalised.At 0.002 (close to 0) the train set accury is highest(more than 80%).

Test Score

At alpha = 0.002 the test accuracy is highest (more than 80%).

After alpha=0.002, the r2 score started decreasing as the alpha is increasing. Select the value of aplha for which the test score peaks up. In our case at alpha=0.002, the error is less in the test set and so accuracy is more close to 80%.

So, the optimum alpha will be 0.002, this level error and generalisation of the model for makes a simpler mode

Generally, smaller values of alpha result in models that are closer to ordinary linear regression, while larger values of alpha introduce more regularization and can help prevent overfitting.

Double the value of alpha for both Ridge and Lasso, the following changes could occur:

Ridge Regression:

The coefficients would be further shrunk towards zero, resulting in a more regularized model.

The model's complexity would decrease, and it would be less prone to overfitting.

Lasso Regression:

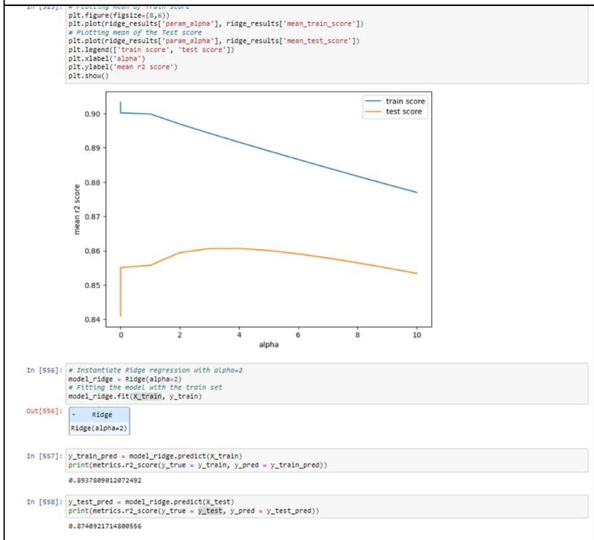
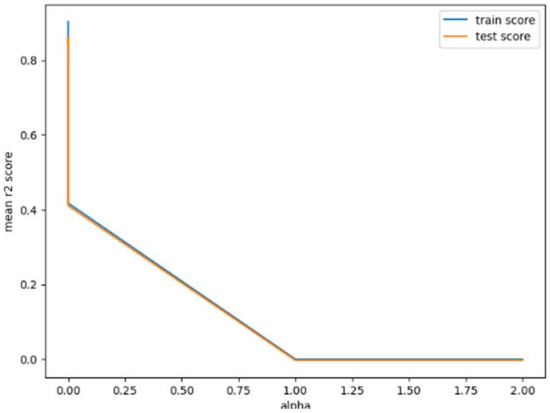
More coefficients would be driven to exactly zero, leading to even sparser models.

### Top 10 features of Ridge regression and coefficients	### Lasso Top 10 features and Co-Efficients
{'OverallQual': 0.1495, 'GrLivArea': 0.1407, 'OverallCond': 0.1008, 'GarageArea': 0.0838, 'FullBath': 0.0646, 'BsmtFullBath': 0.0477, 'Fireplaces': 0.0462, 'BedroomAbvGr': 0.0444, 'd_BsmtQual': 0.041, 'Neighborhood_StoneBr': 0.0373}	{'OverallQual': 0.2628, 'GarageArea': 0.0822, 'GrLivArea': 0.077, 'FullBath': 0.0565, 'Fireplaces': 0.0535, 'd_GarageFinish': 0.0423, 'd_HeatingQC': 0.0313, 'MSZoning_RL': 0.0295, 'd_BsmtFinType1': 0.0292, 'd_BsmtQual': 0.0235}¶

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?

RIDGE	LASSO


`In [525]: # Plotting mean of Train score`
`plt.figure(figsize=(8,6))`
`plt.plot(ridge_results['param_alpha'], ridge_results['mean_train_score'])`
`# Plotting mean of the Test score`
`plt.plot(ridge_results['param_alpha'], ridge_results['mean_test_score'])`
`plt.legend(['train score', 'test score'])`
`plt.xlabel('alpha')`
`plt.ylabel('mean r2 score')`
`plt.show()`
 train score
test score
0.90
0.89
0.88
0.87
0.86
0.85
0.84
mean r2 score
0
2
4
6
8
10
alpha
`In [556]: # Instantiate Ridge regression with alpha=2`
`model_ridge = Ridge(alpha=2)`
`# Fitting the model with the train set`
`model_ridge.fit(X_train, y_train)`
`Out[556]: Ridge`
`Ridge(alpha=2)`
`In [557]: y_train_pred = model_ridge.predict(X_train)`
`print(metrics.r2_score(y_true = y_train, y_pred = y_train_pred))`
`0.8937809012072492`
`In [558]: y_test_pred = model_ridge.predict(X_test)`
`print(metrics.r2_score(y_true = y_test, y_pred = y_test_pred))`
`0.8740921714800556`
	

`Ridge Regression Model:`
 	
`Train data:`
`R2 score: 0.8937809012072492`
`RMSE score: 0.03958866748353199`
 	
`Test data:`
`R2 score: 0.8740921714800556`
`RMSE score: 0.04417766528950032`
 	
Conclusion We have a good train score 85% and good test score as well 83%. That means what the model learnt in the train set it performed well in the test set.
	`Lasso Regression Model`
	
	`Train data:`
	`R2 score: 0.8069457722085366`
	`RMSE score: 0.05337147880821713`
	
	`Test data :`
	`R2 score: 0.800881671942512`
	`RMSE score: 0.05555613339120312`

- The R2 test score on the Ridge Regression Model is better than Lasso Regression Model.
- RMSE on the Ridge Regression Model is better than Lasso Regression Model.

Question 3

After building the model, you realized 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?

five most important predictor variables in the current lasso model

1. 'OverallQual': 0.2628,
2. 'GarageArea': 0.0822,
3. 'GrLivArea': 0.077,
4. 'FullBath': 0.0565,
5. 'Fireplaces': 0.0535,

Dropped five most important predictor variables and performed lasso regression

```
R2 Score -test dataset 0.8320680460799884
 MSE - test dataset is 0.0026030716848734364
```

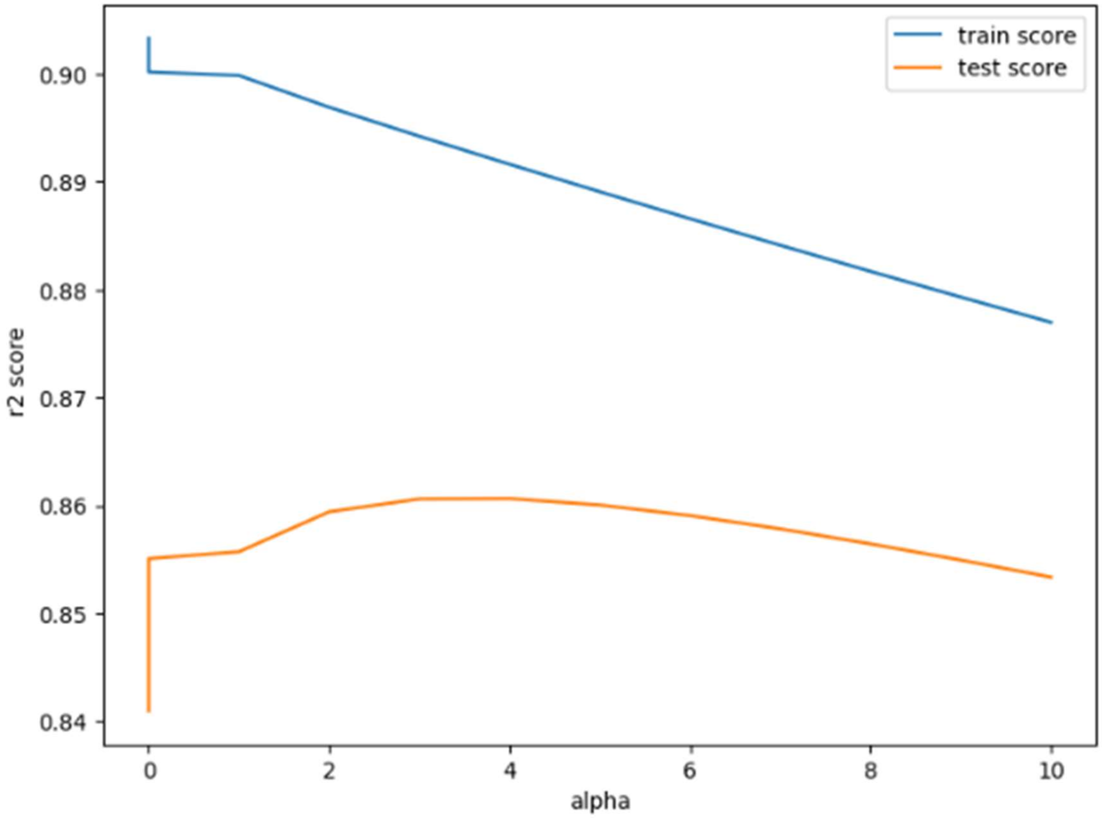
The next set of 5 predictor variables are:

	lassoR_coef
TotalBsmtSF	0.337070
2ndFlrSF	0.168409
OverallCond	0.136727
LotArea	0.074614
Neighborhood_StoneBr	0.064120

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?

1. **Regularization**: Apply regularization techniques like Lasso and Ridge to prevent overfitting. Regularization helps the model capture underlying patterns without fitting to noise in the training data.
2. **Hyperparameter Tuning**: Tune hyperparameters using techniques like grid search or random search. Avoid tuning based solely on the test set performance to prevent overfitting to the test set.



In our model, the r2 score started decreasing as the alpha is increasing after the value 2 and model accuracy started decreasing.

So, we pick the value of alpha =2, the error is much less in the test set and accuracy is good nearly 81%.

So, the optimum value for alpha could be 2, that will have the correct balance between the error and the generalization, to create a simple model.