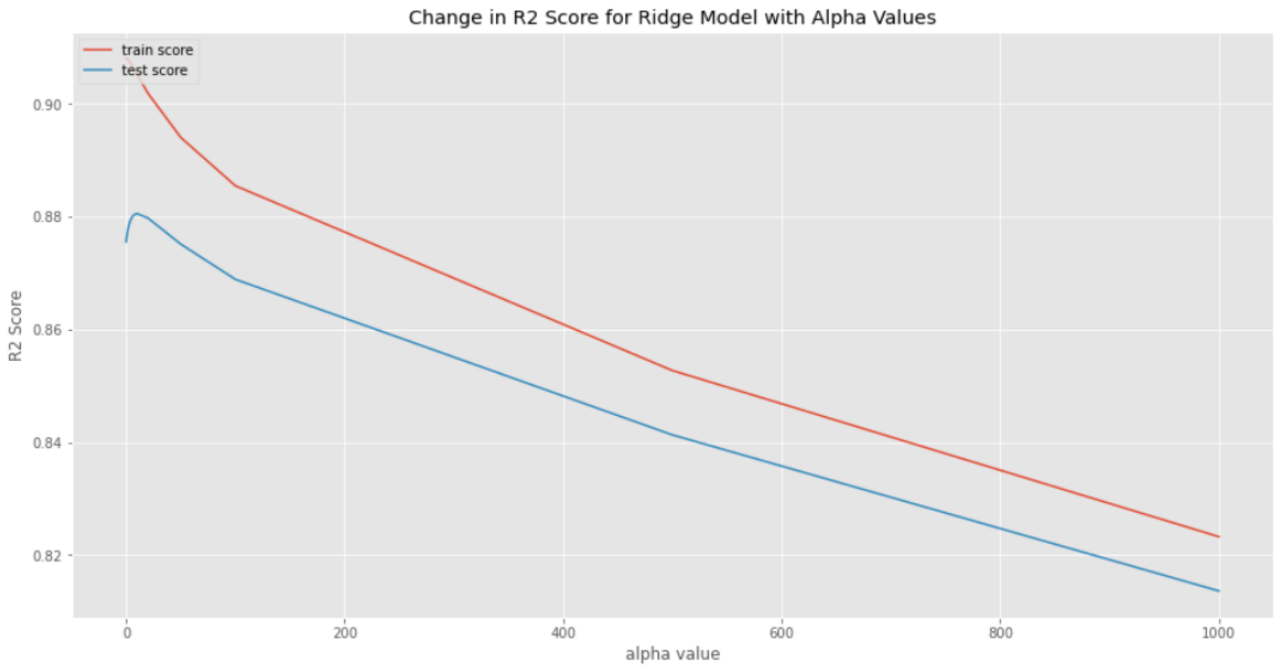
upGrad

Assignment-based Subjective Questions

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?

A. Optimal Value of Alpha for Ridge: 10




```
# Retrieving the best Alpha value
model_cv.best_params_
```

```
{'alpha': 10.0}
```

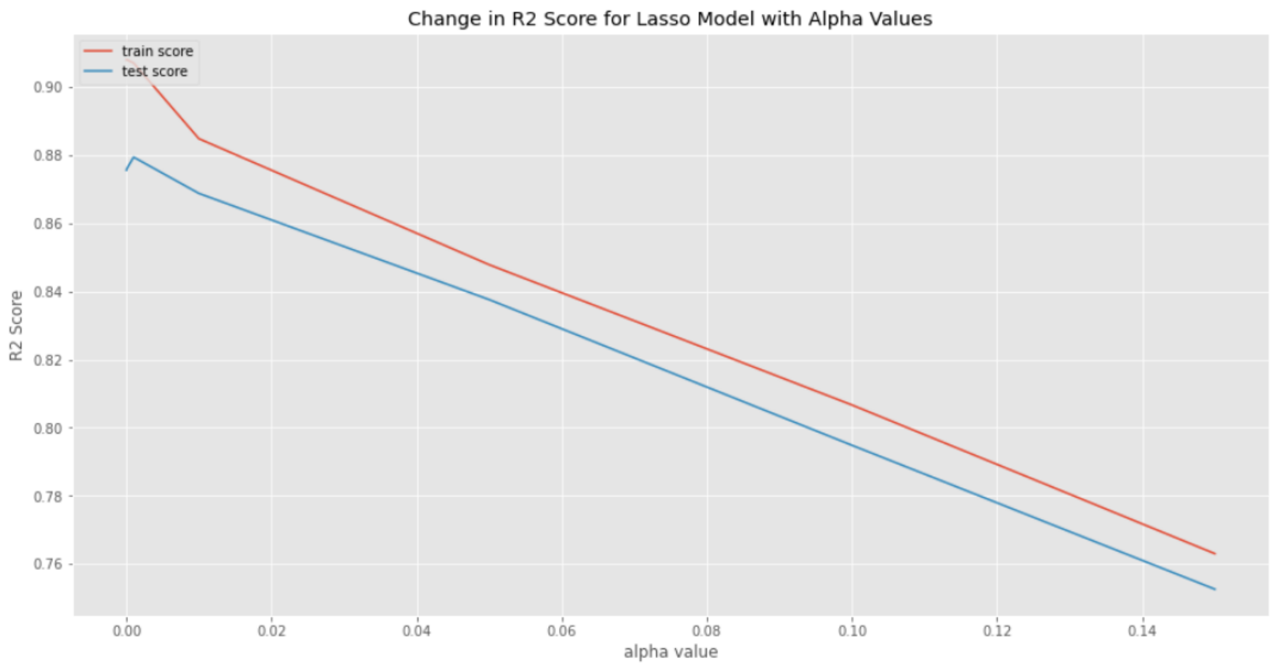
After doubling the alpha for Ridge the performance was as follows:

```
alpha = 20
ridge = Ridge(alpha=alpha)
ridge.fit(X_train, y_train)

# predict
y_train_pred = ridge.predict(X_train)
print("Train R2 Score : ", r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred_ridge = ridge.predict(X_test[final_feats])
print("Test R2 Score : ", r2_score(y_true=y_test, y_pred=y_test_pred_ridge))
```

```
Train R2 Score :  0.9022898483525895
Test R2 Score :  0.8636935680991624
```

Optimal Value of Alpha for Lasso: 0.001




```
# Retrieving the best Alpha value
model_cv.best_params_
```

```
{'alpha': 0.001}
```

After doubling the alpha for Lasso the performance was as follows:

```
alpha = 0.002
lasso = Lasso(alpha=alpha)
lasso.fit(X_train, y_train)

# predict
y_train_pred = lasso.predict(X_train)
print("Train R2 Score : ",r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred_lasso = lasso.predict(X_test[final_feats])
print("Test R2 Score : ",r2_score(y_true=y_test, y_pred=y_test_pred_lasso))
```

```
Train R2 Score :  0.9034872170659044
Test R2 Score :  0.8665120508106925
```

Even after doubling the alphas for both the models the top5 features remained the same indicating that our trained models are very robust and not overfit.

	Feaure	Coef
9	GrLivArea	0.296
41	OverallQual_9	0.277
33	Neighborhood_Crawfor	0.274
42	OverallCond_4	-0.264
39	OverallQual_7	-0.236

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?

A. I would choose to go ahead with Lasso (alpha 0.001) rather than the Ridge Model because both model's performance was very similar and comparable but Lasso had an internal benefit that it takes the coefficients of features irrelevant to zero hence leading to having lesser and more precise feature bucket.
This makes the model less complex and more interpretable.

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?

A. Retraining our model after removing the top 5 features leads us to have this:

```
final_feats.remove("GrLivArea")
final_feats.remove("OverallQual_9")
final_feats.remove("Neighborhood_Crawfor")
final_feats.remove("OverallCond_4")
final_feats.remove("OverallQual_7")
```

```
alpha = 0.001
lasso = Lasso(alpha=alpha)
lasso.fit(X_train[final_feats], y_train)

# predict
y_train_pred = lasso.predict(X_train[final_feats])
print("Train R2 Score : ",r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred_lasso = lasso.predict(X_test[final_feats])
print("Test R2 Score : ",r2_score(y_true=y_test, y_pred=y_test_pred_lasso))
```

```
Train R2 Score :  0.894456592995278
Test R2 Score :  0.8481727211332567
```

Top 5 Features now:

	Feaure	Coef
21	total_sf_area	0.547
32	Neighborhood_Edwards	-0.254
55	KitchenQual_TA	-0.233
0	constant	0.225
60	SaleCondition_Family	-0.211
54	KitchenQual_Gd	-0.196