Question 1

a. What is the optimal value of alpha for ridge and lasso regression?
b. What will be the changes in the model if you choose to double the value of alpha for both ridge and lasso?
c. What will be the most important predictor variables after the change is implemented?

Answer 1

First, we removed the columns with very high proportion of missing values (***PoolQC, MiscFeature, Alley, FireplaceQu, Fence***) & ***ID*** and created dummies for all categorical features. We then started with 537 features.

a)

We used GridSearchCV to test iteratively multiple values of alpha amongst these {'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50, 100, 500, 1000].

When we use all the available features, the alpha that was found optimal was:

Ridge: 10.0

```
In [96]: # Printing the best hyperparameter alpha
         print(model_cv.best_params_)

         {'alpha': 10.0}
```

Lasso: 100

```
In [105]: # Printing the best hyperparameter alpha
          print(model_cv.best_params_)

          {'alpha': 100}
```

b)

For Ridge:

1. The R2 Score for Training data goes down, but the performance gap with Test data is narrowed.
2. The number of non-zero feature coefficients is unchanged from **527 to 527**
3. The coefficients values themselves get smaller
4. The order of features importance changes, ***GrLivArea*** & ***OverallQual_10*** is still the most important but ***Neighborhood_NoRidge*** and ***2ndFlrSF*** order is changed

Model performance scores **Alpha = 10**

```
R2 Score for Training:   0.8976285067496429
R2 Score for Test:       0.8573988115842437

RMSE for Training:   25193.607219872214
RMSE for Test:       30293.5318921134
```

```
]: Df_1
]:
```

	features	coef	coef_abs
118	OverallQual_10	39225.890400	39225.890400
10	GrLivArea	38527.573952	38527.573952
8	2ndFlrSF	38457.629995	38457.629995
74	Neighborhood_NoRidge	33951.233401	33951.233401
117	OverallQual_9	31827.799490	31827.799490
...	...	...	...
305	RoofMatl_Metal	-23.629556	23.629556
77	Neighborhood_SWISU	-21.899838	21.899838
257	YearRemodAdd_1970	-15.228507	15.228507
435	GarageYrBlt_1949.0	-10.341084	10.341084
224	YearBuilt_1997	4.299254	4.299254

527 rows × 3 columns

Model performance scores **Alpha = 20**

```
R2 Score for Training:   0.8798023647821736
R2 Score for Test:       0.8479592298050945

RMSE for Training:   27407.48674739061
RMSE for Test:       31280.116960466876
```

```
: Df_1
:
```

	features	coef	coef_abs
118	OverallQual_10	30376.361332	30376.361332
10	GrLivArea	30168.469156	30168.469156
74	Neighborhood_NoRidge	29175.295408	29175.295408
8	2ndFlrSF	28346.262747	28346.262747
117	OverallQual_9	26881.591256	26881.591256
...	...	...	...
510	MoSold_3	33.471273	33.471273
451	GarageYrBlt_1965.0	-31.729409	31.729409
419	GarageYrBlt_1931.0	-30.871940	30.871940
267	YearRemodAdd_1980	29.333423	29.333423
306	RoofMatl_Roll	-1.071725	1.071725

527 rows × 3 columns

For Lasso:

1. The R2 Score for Training data goes down, but the performance on Test data is nearly unchanged with doubling the alpha.
2. The number of non-zero feature coefficients drops from **134 to 96**
3. The coefficients values themselves drop down

4. The order of features importance changes, ***GrLivArea*** is still the most important but e.g., encoded dummies: ***Condition2_PosN*** (PosN: Near positive off-site feature--park, greenbelt, etc.) and ***OverallQual_10*** (10: Very Excellent) order is changed

Model performance scores **Alpha = 100**

```
R2 Score for Training:  0.9084072805988124
R2 Score for Test:  0.8563112954833194

RMSE for Training:  23924.98795613639
RMSE for Test:  30408.825919706353
```

Features and coeff. Values **Alpha = 100**:

	features	coef	coef_abs
10	GrLivArea	227767.729635	227767.729635
95	Condition2_PosN	-187732.083091	187732.083091
118	OverallQual_10	92897.014647	92897.014647
309	RoofMatl_WdShngl	75457.126656	75457.126656
117	OverallQual_9	70828.087744	70828.087744
...	...	...	...
386	Electrical_SBrkr	298.540343	298.540343
340	MasVnrType_None	257.686823	257.686823
344	ExterQual_TA	-251.156241	251.156241
348	ExterCond_TA	201.994256	201.994256
57	LotConfig_Inside	-46.294739	46.294739

Model performance scores **Alpha = 200**

```
R2 Score for Training:  0.8877810606131872
R2 Score for Test:  0.856322959758456

RMSE for Training:  26482.21645312849
RMSE for Test:  30407.59163997666
```

Features and coeff. Values **Alpha = 200**:

	features	coef	coef_abs
10	GrLivArea	211187.256712	211187.256712
118	OverallQual_10	86593.746706	86593.746706
117	OverallQual_9	73753.477970	73753.477970
95	Condition2_PosN	-63794.635208	63794.635208
19	GarageCars	45976.916731	45976.916731
...	...	...	...
372	BsmtFinType2_Unf	661.672925	661.672925
499	GarageQual_Fa	-514.620324	514.620324
335	Exterior2nd_Stucco	-477.580534	477.580534
81	Neighborhood_StoneBr	234.558611	234.558611
70	Neighborhood_Mitchel	-50.915227	50.915227

c)

Since we used scaled value of features, the absolute value of coefficient tells use about the importance of that feature.

In Ridge regression the top 10 most important predictors are:

For **Alpha = 10** Vs **Alpha = 20**

```
70]: Df_1.iloc[0:10, 0:3]
```

70]:

	features	coef	coef_abs
118	OverallQual_10	39225.890400	39225.890400
10	GrLivArea	38527.573952	38527.573952
8	2ndFlrSF	38457.629995	38457.629995
74	Neighborhood_NoRidge	33951.233401	33951.233401
117	OverallQual_9	31827.799490	31827.799490
13	FullBath	31162.606637	31162.606637
19	GarageCars	29580.414934	29580.414934
7	1stFlrSF	28246.730736	28246.730736
17	TotRmsAbvGrd	27759.047343	27759.047343
309	RoofMatl_WdShngl	27556.614557	27556.614557

```
Df_1.iloc[0:10, 0:3]
```

	features	coef	coef_abs
118	OverallQual_10	30376.361332	30376.361332
10	GrLivArea	30168.469156	30168.469156
74	Neighborhood_NoRidge	29175.295408	29175.295408
8	2ndFlrSF	28346.262747	28346.262747
117	OverallQual_9	26881.591256	26881.591256
13	FullBath	26618.910710	26618.910710
19	GarageCars	25339.432612	25339.432612
17	TotRmsAbvGrd	25302.281537	25302.281537
7	1stFlrSF	22907.488289	22907.488289
18	Fireplaces	21730.396145	21730.396145

As we can see 9/10 features are still common amongst the two models, just that the order of importance has been changed.

The uncommon features are last in each image above ***RoofMatl_WdShngl*** & **Fireplaces**

In lasso regression the top 10 most important predictors are:

For **Alpha = 100** Vs **Alpha = 200**

```
In [163]: Df_1.iloc[0:10,0:3]
```

Out[163]:

	features	coef	coef_abs
10	GrLivArea	227767.729635	227767.729635
95	Condition2_PosN	-187732.083091	187732.083091
118	OverallQual_10	92897.014647	92897.014647
309	RoofMatl_WdShngl	75457.126656	75457.126656
117	OverallQual_9	70828.087744	70828.087744
19	GarageCars	43266.357603	43266.357603
74	Neighborhood_NoRidge	36632.773399	36632.773399
16	KitchenAbvGr	-34203.132561	34203.132561
116	OverallQual_8	32298.432193	32298.432193
8	2ndFlrSF	25592.452164	25592.452164

```
In [158]: Df_1.iloc[0:10,0:3]
```

Out[158]:

	features	coef	coef_abs
10	GrLivArea	211187.256712	211187.256712
118	OverallQual_10	86593.746706	86593.746706
117	OverallQual_9	73753.477970	73753.477970
95	Condition2_PosN	-63794.635208	63794.635208
19	GarageCars	45976.916731	45976.916731
309	RoofMatl_WdShngl	41147.912978	41147.912978
74	Neighborhood_NoRidge	38388.821187	38388.821187
116	OverallQual_8	34342.139291	34342.139291
16	KitchenAbvGr	-26058.773649	26058.773649
75	Neighborhood_NridgHt	23292.522125	23292.522125

We see that 9/10 features are still common amongst the two models, just that the order of importance has been changed.

The uncommon featres are ***2ndFlrSF*** & ***Neighborhood_NoRidge*** *last in the image above.*

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 2

In terms of model performance scores and no. of features.

the Lasso model with **lambda = 200,** seems the best because:

1. The drop from training to test data scores is minimal. Which means that the model can avoid overfit and is learning the underlying general patterns in the data well. This model has lower variance than one with lambda = 100.
2. The use of less features is also better in my opinion. Ideally lesser number of features should be used in a good model. So, the more we move towards that ideal the better. We have only 96 features in this Lasso Model.

The ridge model with Lambda = 20 is also close second basis these same justifications.

Lasso models

Lambda = 200 **Vs** **Lambda =100**

```
R2 Score for Training:  0.8877810606131872
R2 Score for Test:  0.856322959758456

RSS for Training:  716035251824.0266
RSS for Test:  404984273652.48737

MSE for Training:  701307788.2703493
MSE for Test:  924621629.3435785

RMSE for Training:  26482.21645312849
RMSE for Test:  30407.59163997666
```

```
R2 Score for Training:  0.9084072805988124
R2 Score for Test:  0.8563112954833194

RSS for Training:  584425554723.998
RSS for Test:  405017151890.9721

MSE for Training:  572405048.7012713
MSE for Test:  924696693.8150048

RMSE for Training:  23924.98795613639
RMSE for Test:  30408.825919706353
```

Ridge Models

Lambda = 20 **Vs** **Lambda =10**

```
R2 Score for Training:  0.8798023647821736
R2 Score for Test:      0.8479592298050945

RSS for Training:  766944906734.368
RSS for Test:      428559224072.4935

MSE for Training:  751170329.8083918
MSE for Test:      978445717.0604875

RMSE for Training:  27407.48674739061
RMSE for Test:      31280.116960466876
```

```
R2 Score for Training:  0.8976285067496429
R2 Score for Test:      0.8573988115842437

RSS for Training:  653201664083.2285
RSS for Test:      401951756630.33936

MSE for Training:  639766566.1931719
MSE for Test:      917698074.4984916

RMSE for Training:  25293.607219872214
RMSE for Test:      30293.5318921134
```

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 3

When we scale the features, it brings them all to a 0 – 1 scale, which we did in our cases their Beta coefficients reflect their importance in final model

For the Lasso Model with all features and alpha = 100, top 10 most significant variables and model scores are:

	features	coef	coef_abs
10	GrLivArea	227767.729635	227767.729635
95	Condition2_PosN	-187732.083091	187732.083091
118	OverallQual_10	92897.014647	92897.014647
309	RoofMatl_WdShngl	75457.126656	75457.126656
117	OverallQual_9	70828.087744	70828.087744
19	GarageCars	43266.357603	43266.357603
74	Neighborhood_NoRidge	36632.773399	36632.773399
16	KitchenAbvGr	-34203.132561	34203.132561
116	OverallQual_8	32298.432193	32298.432193
8	2ndFlrSF	25592.452164	25592.452164

```
R2 Score for Training:  0.9084072805988124
R2 Score for Test:  0.8563112954833194

RSS for Training:  584425554723.998
RSS for Test:  405017151890.9721

MSE for Training:  572405048.7012713
MSE for Test:  924696693.8150048

RMSE for Training:  23924.98795613639
RMSE for Test:  30408.825919706353
```

Now we will remove the top five: ***GrLivArea, Condition2_PosN, OverallQual_10, RoofMatl_WdShngl, OverallQual_9*** and create another model to see what the changes are.

The new Lasso model with alpha = 100 and the 5 features dropped has he below characteristics:

	features	coef	coef_abs
7	1stFlrSF	170995.421709	170995.421709
8	2ndFlrSF	115218.079334	115218.079334
18	GarageCars	45153.857263	45153.857263
73	Neighborhood_NoRidge	44451.943993	44451.943993
15	KitchenAbvGr	-39009.784089	39009.784089
110	OverallQual_4	-37862.116360	37862.116360
111	OverallQual_5	-35253.238259	35253.238259
109	OverallQual_3	-33727.284731	33727.284731
74	Neighborhood_NridgHt	33561.322412	33561.322412

```
R2 Score for Training:  0.8911237383721647
R2 Score for Test:  0.8560631510837218

RSS for Training:  694706631860.2798
RSS for Test:  405716599619.45447

MSE for Training:  680417856.8660918
MSE for Test:  926293606.4371107

RMSE for Training:  26084.820430014308
RMSE for Test:  30435.07198015327
```

The changes are:

1. The top 2 features are now: ***1stFlrSF& 2ndFlrSF***, where earlier ***2ndFlrSF*** was # 10 in importance earlier.

2. With ***OverallQual_9*** being dropped the other features for overall quality all become more important ***OverallQual_4, OverallQual_5 & OverallQual_3*** are numbers #7, 8 & 9 in the new model.
3. ***GarageCars*** has improved in importance from # 6 earlier to #3 in new model
4. The number of non-zero feature coefficients increased from **134 to 145,** meaning with some of the higher importance variables taken out it was possible for other feature coefficients to increase in importance
5. The model performance score R2 score is down but by a very small amount
6. The RMS errors are higher, but the gap between Training & test has narrowed
7. The test performance scores are virtually the same, just that the gap has narrowed

Question 4

a. How can you make sure that a model is robust and generalisable?
b. What are the implications of the same for the accuracy of the model and why?

Answer 4

a. To ensure that a model is Robust & generalisable, we should:

- **A.** Ensure that the gap in model scores is minimal between Training and test data. High gap points to a high model variance, and that should not be if we want a Robust & generalisable model.
- **B.** Ensure that we that we use models for Interpolation and not extrapolation (especially linear models). If the values are ways out of what the model learned in training, it cannot give an accurate answer.
- **C.** We should reduce the overall number of features in the model as it ensures that confounding effects and multicollinearity issues are minimal.

b.

To ensure a more robust and generalisable model, we may have to choose models that have lower accuracy. Here we give more importance to narrowing the gap between the Training and test performance, rather than trying to achieve highest accuracy in training.

A narrow gap between test/train performance hints that model is learning the general patterns and less affected by noise in the data i.e., it is more robust/generalisable.