Assignment based Subjective Questions

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?

Optimal Value of alpha for ridge: 9.0
Optimal Value of alpha for lasso: 0.001

Ridge Regression – Doubling alpha – Analysis:

Ridge	alpha = 9	double alpha = 18
R2 Train	0.94	0.93
R2 Test	0.91	0.91
RSS Train	8.87	9.49
RSS Test	3.08	3.15
MSE Train	0.01	0.01
MSE Test	0.01	0.01

In summary:

1. Doubling the alpha value in Ridge Regression led to a slight decrease in the R2 Score on the training dataset, indicating a slightly weaker fit to the training data. However, the R2 Score on the test dataset remained the same.
2. The RSS and MSE values also increased slightly, but they are still at very low levels, indicating a good model fit.

Overall, the model's performance remained quite stable, with only minor changes in the metrics.

alpha = 9			alpha = 18	
GrLivArea	0.093		GrLivArea	0.09
OverallCond_9	0.091		Neighborhood_Crawfor	0.074
Neighborhood_Crawfor	0.089		OverallQual_8	0.07
OverallQual_9	0.085		OverallQual_9	0.069
OverallQual_8	0.077		OverallCond_9	0.062
TotalBsmtSF	0.062		TotalBsmtSF	0.06
Functional_Typ	0.062		Functional_Typ	0.055
Exterior1st_BrkFace	0.054		Exterior1st_BrkFace	0.046
MSZoning_FV	0.052		OverallCond_7	0.043
Neighborhood_StoneBr	0.05		Condition1_Norm	0.042

Changing the alpha value in Ridge Regression led to variations in predictor importance. Some predictors became more important, while others became less important. GrLivArea remained the top predictor, but the **specific order and importance of other predictors shifted with the change in alpha**.

Lasso Regression – Doubling alpha – Analysis:

Lasso	alpha = 0.001	double alpha = 0.002
R2 Train	0.93	0.91
R2 Test	0.91	0.89
RSS Train	10.91	13.11
RSS Test	3.34	3.93
MSE Train	0.01	0.01
MSE Test	0.01	0.01

In summary:

1. R2 Score on both the training and test sets decreased, indicating a slight decrease in the model's ability to explain the variance in the target variable
2. RSS on both the training and test sets increased, suggesting that the model's fit to the data became less effective with the higher alpha value.
3. MSE on both the training and test sets remained the same.

Overall, **doubling the alpha in Lasso Regression led to a decrease in the model's predictive performance**, as indicated by lower R2 scores and higher RSS values. However, the MSE values remained consistent.

alpha = 0.001			alpha = 0.002	
OverallQual_9	0.133		GrLivArea	0.104
OverallQual_8	0.11		OverallQual_8	0.091
GrLivArea	0.103		OverallQual_9	0.085
Neighborhood_Crawfor	0.1		Neighborhood_Crawfor	0.073
Functional_Typ	0.062		Functional_Typ	0.059
TotalBsmtSF	0.054		TotalBsmtSF	0.056
OverallQual_7	0.052		OverallQual_7	0.042
OverallCond_9	0.048		GarageArea	0.035
Exterior1st_BrkFace	0.046		Condition1_Norm	0.034
Condition1_Norm	0.042		OverallCond_7	0.029

1. The order and importance of predictors changed after doubling the alpha. Some predictors increased in importance, while others decreased.
2. GrLivArea became the most important predictor, displacing OverallQual_9.
3. The importance of predictors such as OverallQual_8, OverallQual_7, and Exterior1st_BrkFace decreased in comparison to the original model with alpha = 0.001
4. TotalBsmtSF remained important but had a relatively consistent importance ranking
5. OverallCond_9 decreased in importance, while GarageArea and Condition1_Norm increased in importance

Overall, doubling the alpha in Lasso Regression led to **changes in predictor importance**, with some predictors gaining importance while others lost importance in predicting the target variable.

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?

Optimal value of alpha for ridge: 9.0
Optimal value of alpha for lasso: 0.001

	Metric	Linear Regression with RFE	Ridge Regression	Lasso Regression
0	R2 Score (Train)	0.893242	0.939078	0.925013
1	R2 Score (Test)	0.857192	0.914340	0.907096
2	RSS (Train)	15.539092	8.867474	10.914652
3	RSS (Test)	5.137716	3.081726	3.342348
4	MSE (Train)	0.115343	0.087132	0.096668
5	MSE (Test)	0.132646	0.102732	0.106988

Looking at the summary of metrics, Ridge Regression and Lasso Regression outperform Linear Regression with RFE in terms of R2 Score (both on the training and test datasets) and have lower Mean Squared Error values.

These results suggest that Ridge and Lasso Regression are providing better fit and better generalization to the data compared to the basic Linear Regression model.

Additionally, **Ridge Regression has the lowest RSS on the test dataset, indicating good model fit. Hence, I will go with Ridge**.

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?

Lasso Regression

```
top5 = ['OverallQual_9', 'OverallQual_8', 'GrLivArea', 'Neighborhood_Crawfor', 'Functional_Typ']
X_train_new = X_train.drop(top5, axis=1)
X_test_new = X_test.drop(top5, axis=1)
```

```
lasso = Lasso()

# cross validation
model_cv = GridSearchCV(estimator = lasso,
                        param_grid = params,
                        scoring= 'neg_mean_absolute_error',
                        cv = folds,
                        return_train_score=True,
                        verbose = 1)

model_cv.fit(X_train_new, y_train)
```

```
Fitting 5 folds for each of 28 candidates, totalling 140 fits
```

▸ **GridSearchCV**
▸ **estimator: Lasso**
▸ Lasso

```
print(model_cv.best_params_)
```

```
{'alpha': 0.0001}
```

Since the best alpha is around 0.001, in the graph we can see that the error stabilizes around this.

```
lasso = Lasso(alpha=0.0001)

lasso.fit(X_train_new, y_train)
```

▾ Lasso
Lasso(alpha=0.0001)

```
model_parameters = list(lasso.coef_ )
model_parameters.insert(0, lasso.intercept_)
model_parameters = [round(x, 3) for x in model_parameters]
cols = X.columns
cols = cols.insert(0, "constant")
list(zip(cols, model_parameters))
```

```
[('constant', 12.044),
 ('LotFrontage', 0.006),
 ('LotArea', 0.028),
 ('MasVnrArea', -0.004),
 ('BsmtFinSF1', 0.011),
 ('BsmtFinSF2', -0.001),
 ('BsmtUnfSF', -0.012),
 ('TotalBsmtSF', 0.101),
 ('2ndFlrSF', 0.088),
 ('GrLivArea', 0.012),
 ('BsmtFullBath', -0.002),
 ('BsmtHalfBath', 0.023),
 ('FullBath', 0.02),
 ('HalfBath', 0.009),
 ('BedroomAbvGr', 0.016),
 ('Fireplaces', 0.036),
 ('GarageArea', 0.012),
 ('WoodDeckSF', 0.007),
 ('OpenPorchSF', 0.002),
```

After removing the top 5 predictors and building the lasso model again, **alpha changed to 0.0001**. Thereby changing the list of top 10 predictors as well.

	Variable	Coeff
0	constant	12.044
111	OverallCond_5	0.179
234	Fence_MnPrv	0.167
126	Exterior1st_HdBoard	0.130
7	TotalBsmtSF	0.101
8	2ndFlrSF	0.088
37	MSSubClass_190	0.087
119	RoofStyle_Mansard	0.081
75	Neighborhood_SawyerW	0.081
109	OverallCond_3	0.081
110	OverallCond_4	0.079

So now the **new top 5 predictors** are:

1. OverallCond_5
2. Fence_MnPrv
3. Exterior1st_HdBoard
4. TotalBsmtSF
5. 2ndFlrSF

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?

To ensure model is robust and generalizable:

1. We must train our model on diverse range of data sets; it helps model to learn patterns that work across wide range of scenarios.
2. We must use cross-validation techniques and test our model on different parts of data. This ensures our model is not just remembering training data but can make good predictions.
3. We must make sure our model isn't overfitting. We must apply regularization techniques like Lasso or Ridge to penalize large coefficients which can lead to overfitting.
4. We must always test on our model on data that it has never seen before.
5. We must handle outliers, ensure model isn't too sensitive to strange or extreme data points. Robust model should give predictions that make sense even with unusual data.
6. We must do feature selection carefully by understanding the domain properly.
7. Finally, we must also tune hyperparameters. Small changes shouldn't drastically affect our model's performance.

Implications on accuracy:
A robust model may not have the highest accuracy on our training data, but it's more likely to work well in real life. Simpler models are more robust and does not change significantly even if the training data points undergo changes.