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 to double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer:

The optimal value of alpha:

Regression	Alpha
Ridge	2.0
Lasso	0.0001

When the model is again regressed with double alpha values, there is no significant changes in the accuracy (r2_score) of the models for both the Ridge and Lasso regression.

However, the number of variables selected by Lasso drops from **145** to **116**.

The accuracy of Lasso model is little better than Ridge model. After the change, following are the new predictor variables:

1. GrLivArea
2. TotalBsmtSF
3. OverallQual_9
4. GarageArea
5. OverallCond_9

Code:

```
# Doubling the alpha for Ridge, see the jupyter notebook for the complete
code.
ridge = Ridge(alpha=2 * ridge_alpha)
fit_and_predict_regression(ridge)
```

```
0.9458876448967827
0.8933899762780211
```

```
# Doubling the alpha for Lasso
lasso = Lasso(alpha=2 * lasso_alpha)
fit_and_predict_regression(lasso)
```

```
0.9386239514600085
0.8942786744641913
```

```
df_coef = variables_with_coefficent(lasso)
df_coef.shape
```

```
(116, 3)
```

```
df_coef.head(5)
```

	index	Variable	Coef
0	8	GrLivArea	0.2746
1	5	TotalBsmtSF	0.1191
2	91	OverallQual_9	0.0592
3	9	GarageArea	0.0499
4	100	OverallCond_9	0.0407

Questions-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:

The optimal value of alpha:

Regression	Alpha
Ridge	2.0
Lasso	0.0001

When we started with the regression, we had 251 variables which is quite a large number of variables. I will be selecting Lasso regression to apply as it has removed the less significant variables thus leaving behind only the significant variables. Also, the r2_score of the both the regression is almost same.

Regression	r2_score
Ridge	0.8919
Lasso	0.8931

*Thus, **Lasso regression** will be my final selection.*

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:

As per the selected model, following are the selected variables

	index	Variable	Coef
0	8	GrLivArea	0.2830
1	5	TotalBsmtSF	0.1094
2	91	OverallQual_9	0.0626
3	28	MSZoning_RH	0.0546
4	92	OverallQual_10	0.0482

Now, we need to retrain the model after dropping these variables from the training and test data.

```
# Updating X_train and X_test to remove the previous significant
variables, see the jupyter notebook for the complete code.
X_train = housing_model_train.drop([
    'GrLivArea', 'TotalBsmtSF', 'OverallQual_9', 'MSZoning_RH',
    'OverallQual_10'
],
                                   axis=1)

housing_model_test = housing_model_test.drop([
    'GrLivArea', 'TotalBsmtSF', 'OverallQual_9', 'MSZoning_RH',
    'OverallQual_10'
],
                                  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, y_train)

lasso_alpha = model_cv.best_params_['alpha']
```

```
print('alpha: ',lasso_alpha)

lasso = Lasso(alpha=lasso_alpha)
fit_and_predict_regression(lasso)
df_coef = variables_with_coefficent(lasso)
df_coef.shape
```

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


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent
workers.


alpha:  0.0001
0.9424654547527702
0.8932079018443114

[Parallel(n_jobs=1)]: Done 140 out of 140 | elapsed:    0.9s finished

(148, 3)
```

After updating the model following are the new variables:

1. 1stFlrSF
2. 2ndFlrSF
3. BsmtFinSF1
4. LotArea
5. OverallCond_9

We can see that the variables which corresponds to the area of the property are the most important predictors for the property price.

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:

Robustness of a model: It is the property of a model which is tested on a training sample and a test sample. The performance on both the samples should be as close as possible.

Generalising a model: It is the property of the model which dictates how well the model explain the data which it has not seen.

In order to make a model generalizable, we need to make sure that the model should not underfit/overfit the data. If the model remembers all the data, then it will not be able to identify the pattern thus will perform poorly on the test data.

Following are the ways we can make our model robut and generatlizable

1. During EDA, we can identify how the independent variables impact the dependent variable.
2. We can perform feature engineering to combine multiple variable or extract variable from the existing variables which explain the data better
3. In order to avoid overfitting of data, we can use variour techniques like cross validation
4. We can perform outlier treatment so that they do not dominate the model behaviour.
5. We can use more robut error metric for example switching from mean squared error to mean absolute difference reduces the influence of outliers
6. We can also change the scale of the independent variables so that the new values are simpler to relate to the dependent variable.

Implications of these on the accuracy of the model and why If the model is overfitting, it will perform very good on the training data but will perform very bad on the test data thus resulting into a very poor accuracy. If the outliers are not treated properly, they will impact the model such that it does not perfom well when tested on real data where the outliers are infrequent.

If we perform EDA properly, we can identify the pattern of the independent variables against the dependent variable which in turn can result in a simpler model. Performing outlier treatment will result into a better test accuracy on real data.