# Assignment-based Subjective Questions

**Ques 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:** Below are the optimal value of alpha for Ridge and Lasso regression:

- Lasso: 20
- Ridge: 3

```
R2 score coming via Lasso, alpha=20:  [0.9448188165213283, 0.9052585356760134]
R2 score coming via Ridge, alpha=3:  [0.9448091485839297, 0.9031409398691665]
R2 score coming via Lasso, alpha=40:  [0.9448188165213283, 0.9052585356760134]
R2 score coming via Ridge, alpha=6:  [0.9398976077360915, 0.9015605082234307]
```

**Seems R2 score is remain same in Lasso but in Ridge Train and Test both R2 score is little bit decreased**

**Below are the predictor variables:**

- LotArea — Lot size in square feet
- OverallQual — Rates the overall material and finish of the house
- OverallCond — Rates the overall condition of the house
- YearBuilt — Original construction date
- BsmtFinSF1 — Type 1 finished square feet
- TotalBsmtSF — Total square feet of basement area
- GrLivArea — Above grade (ground) living area square feet
- TotRmsAbvGrd — Total rooms above grade (does not include bathrooms)
- Street_Pave — Pave road access to property
- RoofMatl_Metal — Roof material_Metal

Refer the below artifact for R2 score while double the alpha:

### Changing the values of alpha to double:

In [200]:
```python
# Lasso
alpha = 40
lasso = Lasso(alpha=alpha)
lasso.fit(X_train, y_train)
```

Out[200]:
```
Lasso(alpha=40)
```

In [206]:
```python
# Calculating R2 score from Lasso

y_pred_train = lasso.predict(X_train)
y_pred_test = lasso.predict(X_test)

metric3 = []
r2_train_lr = r2_score(y_train, y_pred_train)
print('R2 Score:', r2_train_lr)
metric3.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print('          ', r2_test_lr)
metric3.append(r2_test_lr)
```

```
R2 Score: 0.9448188165213283
          0.9052585356760134
```

In [203]:
```python
# Ridge
alpha = 6
ridge = Ridge(alpha=alpha)
ridge.fit(X_train, y_train)
```

Out[203]:
```
Ridge(alpha=6)
```

In [204]:
```python
# Calculating R2 score from Ridge

y_pred_train = ridge.predict(X_train)
y_pred_test = ridge.predict(X_test)

metric4 = []
r2_train_lr = r2_score(y_train, y_pred_train)
print('R2 Score:', r2_train_lr)
metric4.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print('          ', r2_test_lr)
metric4.append(r2_test_lr)
```

```
R2 Score: 0.9398976077360915
          0.9015605082234307
```

In [205]:
```python
print('R2 score coming via Lasso, alpha=20: ' , metric1)
print('R2 score coming via Ridge, alpha=3: ' , metric2)
print('R2 score coming via Lasso, alpha=40: ' , metric3)
print('R2 score coming via Ridge, alpha=6: ' , metric4)
```

```
R2 score coming via Lasso, alpha=20:  [0.9448188165213283, 0.9052585356760134]
R2 score coming via Ridge, alpha=3:  [0.9448091485839297, 0.9031409398691665]
R2 score coming via Lasso, alpha=40:  [0.9448188165213283, 0.9052585356760134]
R2 score coming via Ridge, alpha=6:  [0.9398976077360915, 0.9015605082234307]
```

**Seems R2 score is remain same in Lasso but in Ridge Train and Test both R2 score is little bit decreased**

**Ques 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 R2 square value of Lasso is slightly higher than Ridge for the test dataset so we will choose Lasso regression to solve this problem.

- Ridge regression, uses a tuning parameter called lambda as the penalty is square of magnitude of coefficients which is identified by cross validation.

- Lasso regression, uses a tuning parameter called lambda as the penalty is absolute value of magnitude of coefficients which is identified by cross validation. Lasso also does variable selection.

**Ques 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:** Those 5 most important predictor variables that will be excluded are:

- GrLivArea
- TotalBsmtSF
- BsmtFinSF1
- GarageArea
- TotRmsAbvGrd

In [217]:
```
X_train.columns
```
Out[217]:
```
Index(['MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2',
       ...
       'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New', 'SaleType_Oth',
       'SaleType_WD', 'SaleCondition_AdjLand', 'SaleCondition_Alloca',
       'SaleCondition_Family', 'SaleCondition_Normal',
       'SaleCondition_Partial'],
      dtype='object', length=242)
```

In [218]:
```
X_train2 = X_train.drop(['MSSubClass','LotFrontage','LotArea','OverallQual','OverallCond'],axis=1)
X_test2 = X_test.drop(['MSSubClass','LotFrontage','LotArea','OverallQual','OverallCond'],axis=1)
```

In [219]:
```
# Lasso
alpha = 20
lasso5 = Lasso(alpha=alpha)
lasso5.fit(X_train2, y_train)
```
Out[219]:
```
Lasso(alpha=20)
```

In [221]:
```
# Important predictor variables
betas = pd.DataFrame(index=X_train2.columns)
betas.rows = X_train2.columns
betas['lasso5'] = lasso5.coef_
pd.set_option('display.max_rows', None)
betas.head(68)
```
Out[221]:

| | lasso5 |
|---|---|
| **YearBuilt** | 25252.821771 |
| **YearRemodAdd** | 17544.732173 |
| **MasVnrArea** | 14221.868906 |
| **BsmtFinSF1** | 29683.638606 |
| **BsmtFinSF2** | 0.000000 |
| **BsmtUnfSF** | 0.000000 |
| **TotalBsmtSF** | 70980.521577 |
| **1stFlrSF** | 8572.464983 |
| **2ndFlrSF** | 0.000000 |
| **LowQualFinSF** | 0.000000 |
| **GrLivArea** | 166499.766208 |

**Ques 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:** The model should be accurate for datasets other than the ones which were used during training. It also should be as simple as possible, though its accuracy will decrease but it will be more robust and generalisable. Too much importance should not give to the outliers so that the accuracy predicted by the model is high. The simpler be the model the more the bias but less variance and more generalizable. Its implication in terms of accuracy is that a robust and generalisable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data. The outliers which it does not make sense to keep must be removed from the dataset. If the model is not robust, It cannot be trusted for predictive analysis.