**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?

As a part of my regression analysis, initially observed alpha to be 500 for both ridge and lasso (obtained using model_cv.best_params_).

**Initial important predictors for Ridge**

```
In [173]: betas.sort_values(by=['Ridge'],ascending=False)
Out[173]:
```

| | Linear | Ridge | Lasso |
|---|---|---|---|
| OverallQual_E | 8.037500e+03 | 6833.417532 | 1.194531e+04 |
| GrLivArea | 1.892547e+14 | 6369.089244 | 2.381268e+04 |
| 1stFlrSF | -1.444207e+14 | 5417.135493 | 2.294866e+03 |
| GarageCars_3 | -5.430214e+14 | 4788.167631 | 6.179995e+03 |
| OverallQual_VG | 7.685789e+03 | 4608.239673 | 9.280963e+03 |
| Neighborhood_NridgHt | 2.143438e+03 | 4486.767522 | 4.958708e+03 |
| Neighborhood_StoneBr | 4.010438e+03 | 3879.143703 | 4.353733e+03 |
| TotalBsmtSF | 1.730196e+15 | 3712.188843 | 2.927173e+03 |
| LotArea | 8.185915e+03 | 3475.948261 | 4.069002e+03 |
| GarageArea | 2.619875e+03 | 3296.591811 | 2.071801e+03 |
| Neighborhood_NoRidge | 2.100006e+03 | 3075.793097 | 2.493298e+03 |

**Alpha doubled- predictors for Ridge**

```
In [179]: betas.sort_values(by=['Ridge'],ascending=False)
Out[179]:
```

| | Linear | Ridge | Lasso |
|---|---|---|---|
| OverallQual_E | 8.037500e+03 | 6833.417532 | 1.194531e+04 |
| GrLivArea | 1.892547e+14 | 6369.089244 | 2.381268e+04 |
| 1stFlrSF | -1.444207e+14 | 5417.135493 | 2.294866e+03 |
| GarageCars_3 | -5.430214e+14 | 4788.167631 | 6.179995e+03 |
| OverallQual_VG | 7.685789e+03 | 4608.239673 | 9.280963e+03 |
| Neighborhood_NridgHt | 2.143438e+03 | 4486.767522 | 4.958708e+03 |
| Neighborhood_StoneBr | 4.010438e+03 | 3879.143703 | 4.353733e+03 |
| TotalBsmtSF | 1.730196e+15 | 3712.188843 | 2.927173e+03 |
| LotArea | 8.185915e+03 | 3475.948261 | 4.069002e+03 |
| GarageArea | 2.619875e+03 | 3296.591811 | 2.071801e+03 |
| Neighborhood_NoRidge | 2.100006e+03 | 3075.793097 | 2.493298e+03 |

**Initial important predictors for Lasso**

```
In [175]: betas.sort_values(by=['Lasso'],ascending=False)
Out[175]:
```

| | Linear | Ridge | Lasso |
|---|---|---|---|
| GrLivArea | 1.892547e+14 | 6369.089244 | 2.381268e+04 |
| OverallQual_E | 8.037500e+03 | 6833.417532 | 1.194531e+04 |
| OverallQual_VG | 7.685789e+03 | 4608.239673 | 9.280963e+03 |
| GarageCars_3 | -5.430214e+14 | 4788.167631 | 6.179995e+03 |
| SaleType_New | 1.060568e+04 | 2176.303507 | 5.936681e+03 |
| OverallQual_VE | 6.341688e+03 | 2737.315827 | 5.417590e+03 |
| Neighborhood_NridgHt | 2.143438e+03 | 4486.767522 | 4.958708e+03 |
| Neighborhood_StoneBr | 4.010438e+03 | 3879.143703 | 4.353733e+03 |
| LotArea | 8.185915e+03 | 3475.948261 | 4.069002e+03 |
| BsmtFullBath_1 | 2.094883e+03 | 2957.392012 | 4.038498e+03 |
| BsmtExposure_Gd | 2.689062e+03 | 2899.712563 | 3.998006e+03 |

**Alpha doubled- predictors for Lasso**

```
In [180]: betas.sort_values(by=['Lasso'],ascending=False)
Out[180]:
```

| | Linear | Ridge | Lasso |
|---|---|---|---|
| GrLivArea | 1.892547e+14 | 6369.089244 | 2.381268e+04 |
| OverallQual_E | 8.037500e+03 | 6833.417532 | 1.194531e+04 |
| OverallQual_VG | 7.685789e+03 | 4608.239673 | 9.280963e+03 |
| GarageCars_3 | -5.430214e+14 | 4788.167631 | 6.179995e+03 |
| SaleType_New | 1.060568e+04 | 2176.303507 | 5.936681e+03 |
| OverallQual_VE | 6.341688e+03 | 2737.315827 | 5.417590e+03 |
| Neighborhood_NridgHt | 2.143438e+03 | 4486.767522 | 4.958708e+03 |
| Neighborhood_StoneBr | 4.010438e+03 | 3879.143703 | 4.353733e+03 |
| LotArea | 8.185915e+03 | 3475.948261 | 4.069002e+03 |
| BsmtFullBath_1 | 2.094883e+03 | 2957.392012 | 4.038498e+03 |
| BsmtExposure_Gd | 2.689062e+03 | 2899.712563 | 3.998006e+03 |

**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?**

Lasso shows a better R2 score on test and also it eliminates coefficients and provides us with better explainability of the model through 164 coefficients.

**Ridge Scores – Alpha 500**

```
MSE Test 421285670.6334385
MAE Score Test 13531.069539602102
R2 Score Test 0.9186504930448596
RSS Score Test 489112663605.4221

MSE Test 931193918.8226645
MAE Score Test 17447.215027218248
R2 Score Test 0.8291537265012339
RSS Score Test 270977430377.39536
```

**Lasso Scores – Alpha 500**

```
MSE Test 379607919.8270465
MAE Score Test 13036.244042934175
R2 Score Test 0.9266983919301963
RSS Score Test 440724794919.201

MSE Test 1142487630.2857523
MAE Score Test 17884.72063882557
R2 Score Test 0.7903876408476326
RSS Score Test 332463900413.15393
```

**Ridge non zero coefficients:**

```
# Printing the best hyperparameter alpha
print(model_cv.best_params_ , "Number of non-zero Coefficients {}".format(np.sum(ridge.coef_!=0)))
```

```
{'alpha': 500} Number of non-zero Coefficients 329
```

**Lasso non zero coefficients:**

```
# Printing the best hyperparameter alpha
print(model_cv.best_params_ , "Number of non-zero Coefficients {}".format(np.sum(lasso.coef_!=0)))
```

```
{'alpha': 500} Number of non-zero Coefficients 164
```

**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?**

The top 5 predictors 'OverallQual','GrLivArea','1stFlrSF','GarageCars','Neighbborhood' are dropped and a new model is built. Analysing the coefficients gives the below result of new dependent variables

In [196]:
```
betas_new.sort_values(by=['Lasso'],ascending=False)
```

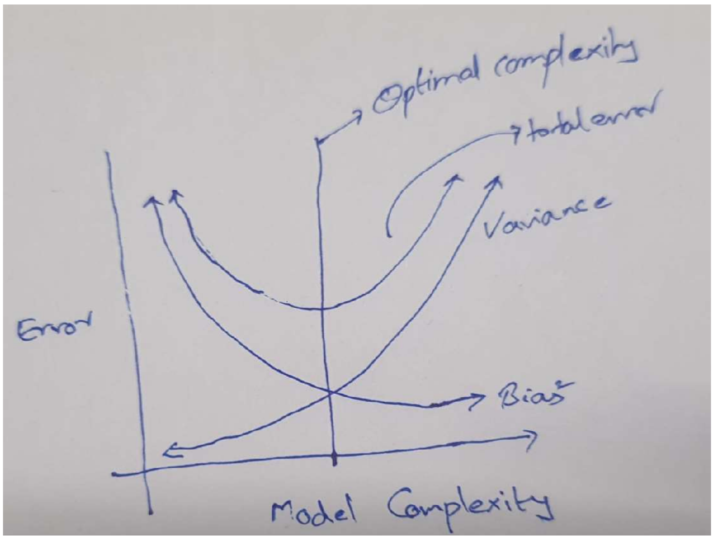
Out[196]:

| | Ridge | Lasso |
|---|---|---|
| **TotalBsmtSF** | 11416.744792 | 19498.041572 |
| **2ndFlrSF** | 8005.329322 | 12015.502370 |
| **GarageArea** | 9506.983474 | 9757.084190 |
| **MasVnrArea** | 6763.459393 | 5888.738283 |
| **LotArea** | 6470.832330 | 5393.469970 |
| **SaleType_New** | 2809.502586 | 5137.794031 |
| **BsmtFinType1_GLQ** | 3750.244233 | 4658.437106 |
| **TotRmsAbvGrd_9** | 4980.362274 | 4166.616182 |
| **RoofMatl_CompShg** | 7351.152758 | 4010.269277 |
| **TotRmsAbvGrd_10** | 5217.042047 | 3967.498619 |
| **BsmtFullBath_1** | 3897.997985 | 3819.046965 |

**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?**

Through regularization, model can be prevented from becoming extremely complex (avoid overfitting on training data). Hyperparameters are passed to learning algorithm to control the complexity of final model. When regularization is applied, penalty term is applied to model's cost function which will help bring the magnitude of model coefficients to 0. Through regularization, we compromise by allowing little bias for a significant gain in variance.




Two regularization techniques are Ridge and Lasso

Cost function for Ridge = $\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$

Cost function for Lasso = $\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$

As the alpha values ($\lambda$) increase model tends to underfit

If the alpha is 0 then there is no penalty and model overfits.