## Question 1

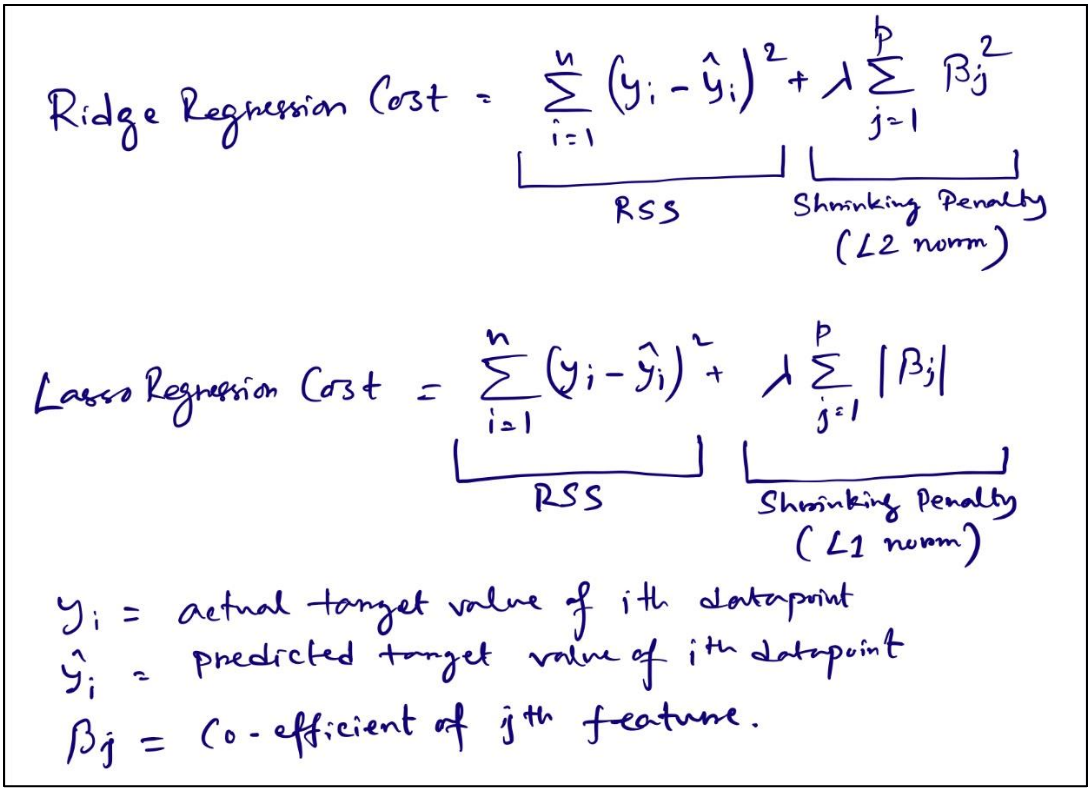
### What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

**Optimal value of alpha:**

- Optimal alpha (lambda) value for Ridge Regression model is: 8
- Optimal alpha (lambda) value for Lasso Regression model is: .0006

**Effect of choosing double the value of optimal alpha:**

Before explaining the second part of the question, let's see the cost functions of Ridge and Lasso.



$$\text{Ridge Regression Cost} = \underbrace{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}_{\text{RSS}} + \underbrace{\lambda\sum_{j=1}^{p}\beta_j^2}_{\substack{\text{Shrinking Penalty}\\(L2\text{ norm})}}$$

$$\text{Lasso Regression Cost} = \underbrace{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}_{\text{RSS}} + \underbrace{\lambda\sum_{j=1}^{p}|\beta_j|}_{\substack{\text{Shrinking Penalty}\\(L1\text{ norm})}}$$

$y_i$ = actual target value of ith datapoint

$\hat{y}_i$ = predicted target value of ith datapoint

$\beta_j$ = Co-efficient of jth feature.

So, here it can be seen that in both the cases penalty term increases with higher value of beta co-efficient. Ridge imposes more aggressive penalty as it uses sum of square of all beta coefficients (L2 norm) as shrinking penalty. Where Lasso uses sum of absolute values of all beta coefficients (L1 norm) as shrinking penalty. In both equations these norms are multiplied by lambda or alpha. This alpha is a hyperparameter and its optimal value can be obtained by performing cross validation. Value of alpha can be any number >= 0.

If we increase the value of alpha then shrinking penalty will be higher, so Ridge and Lasso both will try to shrink values of beta coefficients towards zero, so our model will be simpler. That means it will increase the bias where variance will be reduced. If we increase the value of alpha to a very large number, then all coefficients of Lasso become 0 and for Ridge coefficients become close to zero (as they cannot be exact 0 in Ridge). That means the model will have very high bias and low variance and it may result in underfitting. That means model will fail to learn the underlying data pattern in training dataset.

If we reduce the value of alpha then shrinking penalty will be lower, so model bias will reduce, and variance will increase. Now if we put value of alpha as 0, then the cost function of both Ridge and Lasso become OLS cost function (i.e., RSS) and we will get exact same model as we get using OLS. So, reducing value of alpha will reduce the effect of shrinking penalty may lead to possible overfitting for very low or close to zero value of alpha.

So, we need to find the optimal value of alpha by performing hyperparameter tunning.

Top 10 features with beta coefficient values obtained from Ridge after using alpha= 16

```
OverallQual             0.205139
GrLivArea               0.145057
OverallCond             0.111117
GarageArea              0.106036
1stFlrSF                0.103177
2ndFlrSF                0.101840
FullBath                0.091148
Neighborhood_StoneBr    0.087765
MSSubClass_30          -0.081856
Exterior1st_BrkFace     0.078220
```

Top 10 features with beta coefficient values obtained from Lasso after using alpha= .0012

```
GrLivArea               0.368980
OverallQual             0.364970
GarageArea              0.148863
OverallCond             0.130027
Neighborhood_StoneBr    0.086031
Exterior1st_BrkFace     0.081740
Neighborhood_NridgHt    0.080808
MSSubClass_30          -0.080712
YearRemodAdd            0.076909
CentralAir              0.076685
```

**So, after Doubling value of alpha the most important variable:**

In Ridge model: **OverallQual** (Rates the overall material and finish of the house)<br>
In Lasso model: **GrLivArea** (Above grade (ground) living area square feet)

## 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?

As per Occam's Razor a model should not be unnecessary complex.

Model complexity depends on two main things: **No. of features or independent variables** and **Magnitude of beta coefficients.** Normalization (Ridge and Lasso) already shrinks beta coefficients towards zero.

Now, Lasso and Ridge both have similar r2 score and MAE on test dataset. But Lasso has eliminated 110 features and final no. of features in Lasso Regression model is 116. Where Ridge has all 226 features. So, the Lasso model is simpler than Ridge with having similar r2 score and MAE.

Ridge:

```
r2 score on testing dataset: 0.8911807696767164
MSE on testing dataset: 0.018419435953924413
RMSE on testing dataset: 0.135718222630288
MAE on testing dataset: 0.09344961892214859
```

Lasso:

```
r2 score on testing dataset: 0.8947392213072709
MSE on testing dataset: 0.01781710976847528
RMSE on testing dataset: 0.13348074680820182
MAE on testing dataset: 0.09142208307508749
```

As these two models shows almost similar performance on test dataset, we should choose the simpler model. So, I will choose Lasso as my final model

## Question 3

## After building the model, you realized 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?

Initially top 5 features in Lasso model are as below:

```
GrLivArea                0.377281      (Above grade (ground) living area square feet)
OverallQual              0.309806      (Rates the overall material and finish of the house)
OverallCond              0.144188      (Rates the overall condition of the house)
Neighborhood_StoneBr     0.136858      (Dummy variable of Neighborhood = Stone Brook)
GarageArea               0.134759      (Size of garage in square feet)
```

As Neighborhood_StoneBr is a dummy variable, dropping entire Neighborhood feature.
After dropping GrLivArea, OverallQual, OverallCond, GarageArea, Neighborhood features, rebuilt the Lasso model again with rest of the features, now 5 most important predictor variables are as below.

```
1stFlrSF                  0.402011  (First Floor square feet)
2ndFlrSF                  0.369645  (Second floor square feet)
GarageType_Not Present   -0.136456  (Dummy variable of GarageType= No Garage)
KitchenQual_TA           -0.132718  (Dummy variable of Kitchen quality Typical/Average)
Exterior1st_BrkFace       0.130630  (Dummy variable of Exterior covering on house is Brick Face)
```

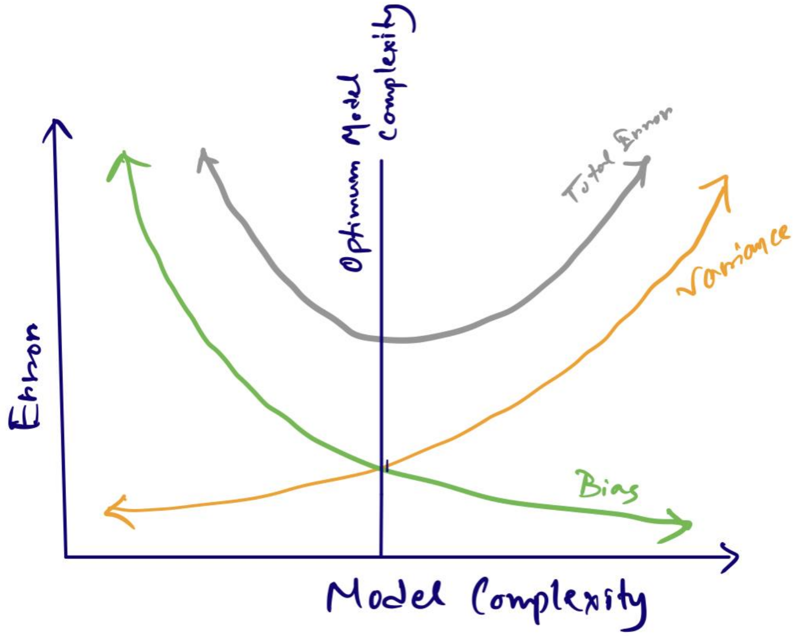
## Question 4
## How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

A model should be complex enough so that it learns the data patters in the training dataset but not too complex that it also learns noises in the training dataset. The model should be generalized enough and not so complex that it memorizes all datapoints in training dataset.

An underfitting model usually has high bias and low variance. It fails to understand data pattern in training dataset, so it performs bad both on training and testing dataset. Whereas an overfitting model usually has low bias and high variance. It performs good on training dataset but performs bad on testing dataset or unseen data.

A scenario of overfitting can be identified easily by comparing model performance in training and testing dataset. If there is a significant difference in model performance (r2 score, model accuracy, MAE, RMSE, Confusion Matrix etc. other evaluation metrics) on training and testing dataset then it’s a case of overfitting.

A robust model should have low bias and low variance and it should not suffer from underfitting and overfitting. It can be achieved by doing a trade-off between bias and variance. One of the ways to remove overfitting to create a robust and generalizable model is to reduce model complexity.



Bias - Variance Tradeoff

Model complexity depends on two main things: **Number of features or independent variables** and **Magnitude of beta coefficients.** Normalization (Ridge and Lasso) already shrinks beta coefficients towards zero. Again, Lasso also helps to reduce number of features by shrinking some beta coefficients to exact 0. Thus, it helps to overcome overfitting. Accuracy of a robust and generalizable model should be almost same/closer on training and testing datasets.