**Introduction**

Welcome to the module on Advanced Regression.

In the previous module, we had learnt the concepts of model selection for the various learning algorithms. Now moving ahead, we want to implement the application of model selection in the regression framework to reduce the complexity of regression models. Below is a list of what is covered in this module.

* **Build the Conceptual Foundation for Generalized Linear Regression:**
  + The meaning of "linearity" in this context and why it is desirable.
  + Understand the usage of an appropriate family of functions for a particular regression task.
  + Build geometric intuition about regression in general.

**Understand how non-linear regression is different from linear regression and circumstances:**

* Be able to identify when non-linearity might be required.
* Be able to use GLM implementations such as *glm* in R to solve regression problems.

**Evaluation / Comparison of Ridge and Lasso regression solutions:**

* Be able to evaluate and assess any given regression solution for a given dataset.
* Be able to objectively compare regression solutions and make a choice based on tuning factor "lambda".
* Learn how to read the summary of a model produced by implementations like glm in R.

We will learn primarily about these two broad concepts of Advanced Regression in this module:

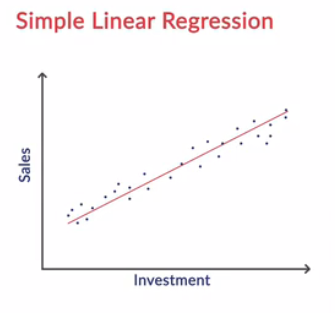
1. Generalized Linear Regression
2. Ridge and Lasso Regression

## In this session:

We will explore regression a little further and build models based on what we have learnt from the first course on Predictive Analytics. We will deliberately explore regression in a somewhat more generalised form than what you may have been exposed to until now. This would expand the scope of regression significantly while making regression considerably more robust.

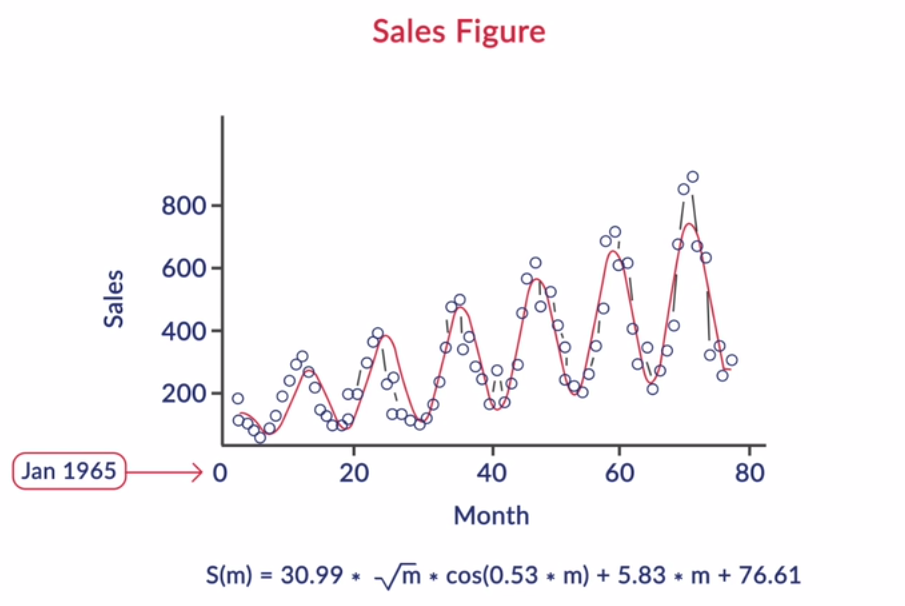
**Downloads**

Download the *Electric consumption data*(used throughout the session) from below. The data set contains two attributes namely "Year"  and "Total electricity consumption".

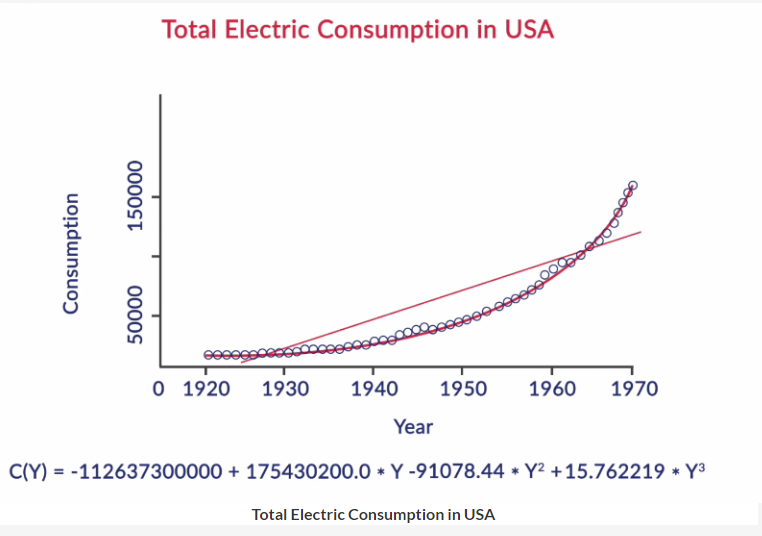


In the earlier module on linear regression, we have taught ways to fit the linear curve to explain variability in data points. Now we have learnt that in many scenarios, the explanatory and response variables do not vary in a linear manner. Let us evaluate each of these scenarios to understand the type of patterns these variables follow:

1. In the first example, you may have noticed that the data points oscillate and follow some kind of sine or cosine type pattern.



1. In the second example of electric consumption, you have seen that the data points gradually increase which clearly indicate the polynomial fit. In this case, you can see the 3rd order polynomial would be the best fit.



**Total Electric Consumption in USA**

You may recall from your higher secondary studies that "quadratic function" is another name for our formulated regression function. Nonetheless, you'll often hear statisticians referring to the quadratic model as a second-order model if the highest power of the explanatory variables term is 2.

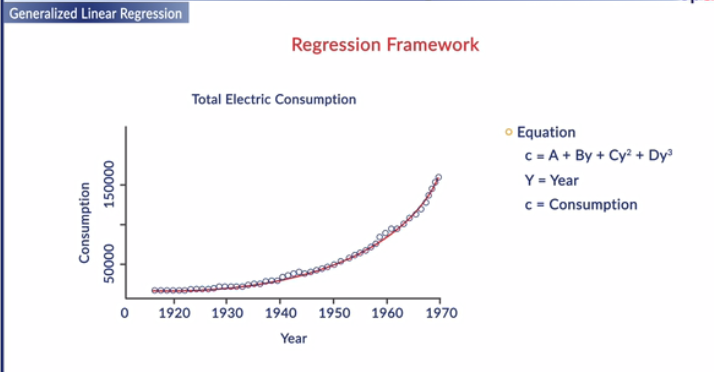
So how can we take the decision of fitting the polynomial model or sine curve or any other model by just looking at the scatter plot? Let's try to find out these answers in the next segment.

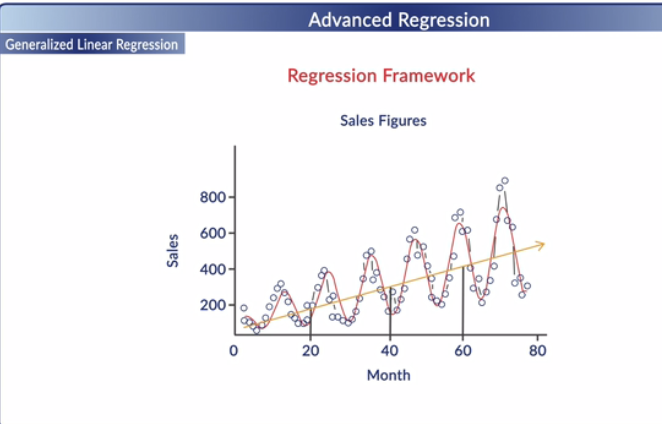
**Generalized Regression Framework-1**

In the general regression problem, we learnt how to build a linear curve that has the best fit to a series of data points. We will now learn how to extend the general regression framework onto examples, wherein the data does not follow a linear pattern.

We should follow these two steps while building any best fit model in order to explain the data:

1. Carry out exploratory data analysis by examining scatter plots of explanatory variables and the dependent variable.
2. Choose an appropriate set of function to define the scatter plot with reasonable precision.



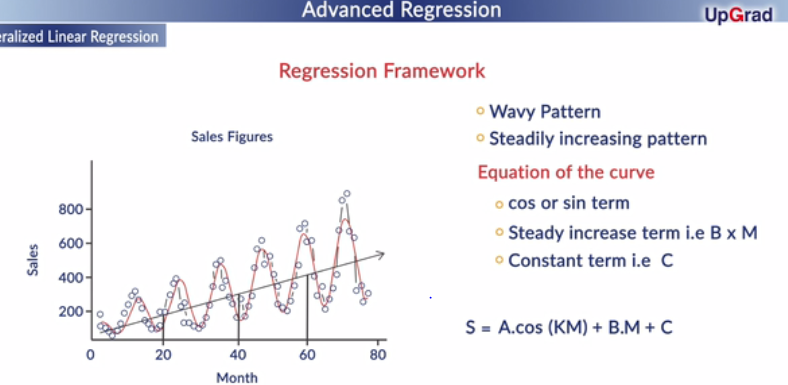
 **Sales Figure**

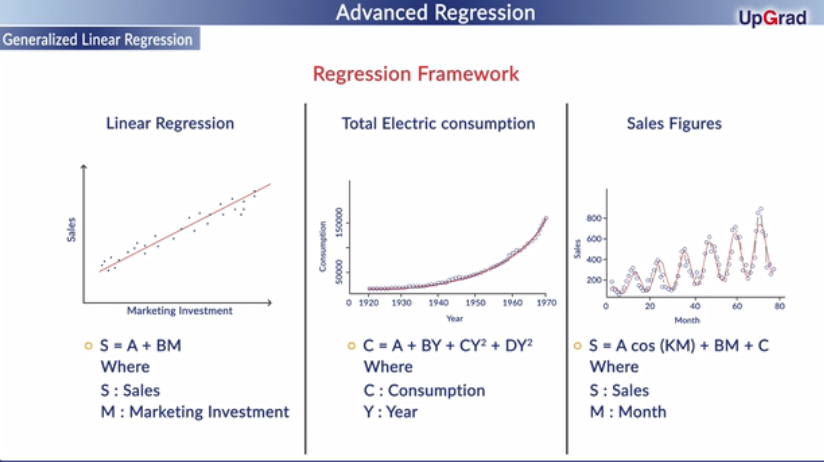
The sales graph (discussed in the lecture) shows a wavy nature. Select the appropriate function which could possibly depict wavy nature.

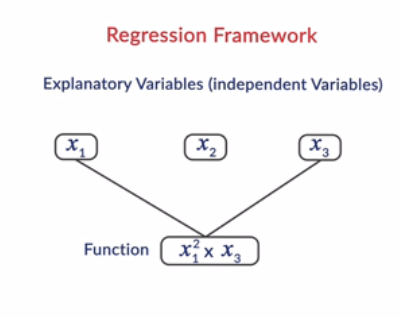


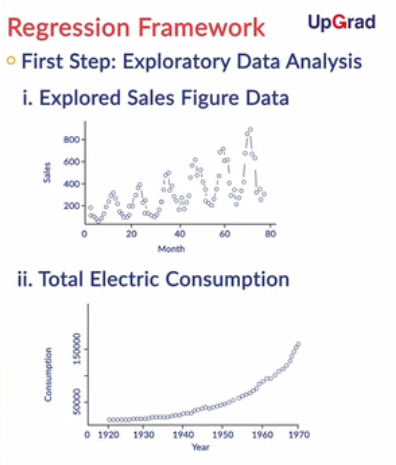
**Y ~ C cos(x)**

**Feedback :***The graph of cosine/sine function repeats itself at intervals of 360 degrees with one peak and one dip (in each 360 degree interval) giving it a wavy nature.*



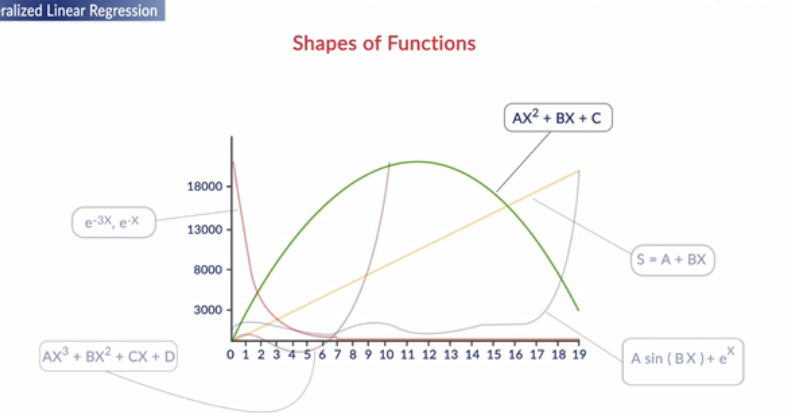






We first visualized the scatter plots and identified the shape of the data. Second, we assessed which function of the explanatory variable would explain the shape of the data. In constructing the non-linear regression model, instead of using the explanatory variables in the current form, we create some function of the explanatory variables to best explain the data points. These functions capture the non-linearity in the data.

Let's see what are the typical shapes of commonly occurring functions. Further, we will also learn how functions are created with derived features instead of raw attributes. Feature values can be raw attributes or derivations from the raw attributes.



Consider x1 and x2 as two independent variables which are used to create certain features. Which of the following is a non linear feature? Select the correct option.



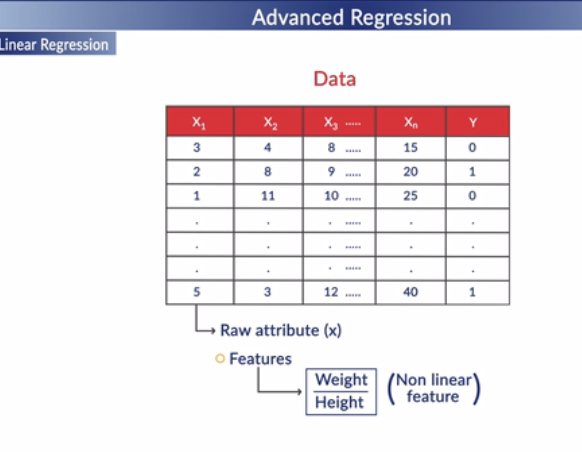
X1 \*X2



X1/X2



**Both B and C**



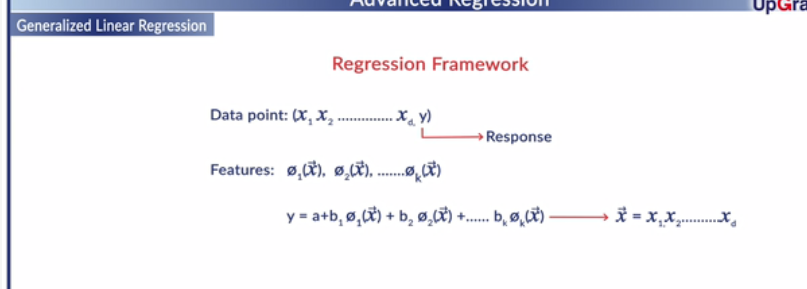
Raw attributes as they appear in the training data, may not be the ones best suited to predict the response variable value with. Recall the blood pressure example where the raw weight and height are individually not indicative of the blood pressure. The way out is to design 'features' that are more likely to describe the response variable, from the raw attributes. The features could be combinations of two or more attributes and/or tranformations of individual attributes. These combinations and transformations could in general be non-linear.

The features in the data will directly impact the predictive models we use and the results we achieve. You will get good at deciding which features to create based on empirical knowledge (like in our blood pressure example), domain knowledge and various other general approaches. Mastery of feature design comes with hands-on practice and study of best practices.

# Generalized Regression Framework-2

In general, we are given training data consisting of **n entities** with **d attributes**each (called the explanatory variables)

We believe these attributes explain a response attribute at the end. Till now, we have understood the concepts of generalized linear regression. The next step is to find out the coefficients of models mathematically. Let's see how Professor Raghavan can help us understand the derivation and the formula in general form of regression.



**Linearity**

Which of the following cannot be qualified as a linear regression model?



**y = ae^(bx) + c**

**Feedback :***In this model, if you check if the function (rhs) is ‘linear’ in the parameters treating everything else as a constant, it is clearly evident that y does not vary linearly with b.*

**Vector Product**

Consider two vectors a = (a1,a2) and b= (b1,b2). Find the dot product of the vectors - (a\*b). Select the correct option.

Top of Form



**a1b1 + a2b2**

**Feedback :***If the concept of dot product is still unclear, it is recommended to revise the concept once more through the links given in the additional reading.*

Bottom of Form

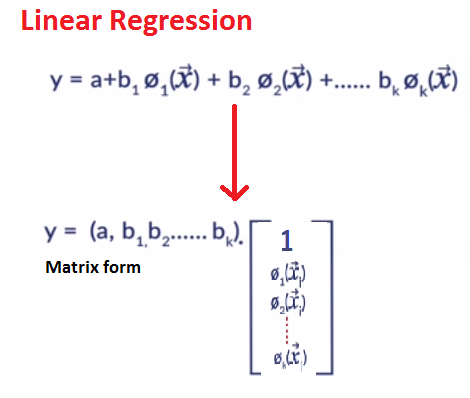
Let us take two important takeaways from the above lecture:

1. Here, the general regression problem remains the same where we compute the values of constants so that the function fits the data set best. We have just replaced x1{}{_{_{}}}, x2....xk with \phi(x1), \phi(x2).........\phi(xk).
2. The term 'linear' in regression depicts the linear expression in the coefficients of the linear combination of the features. It does not mean linear expression in raw attributes or features. Feature functions could be non-linear.

## Expressions

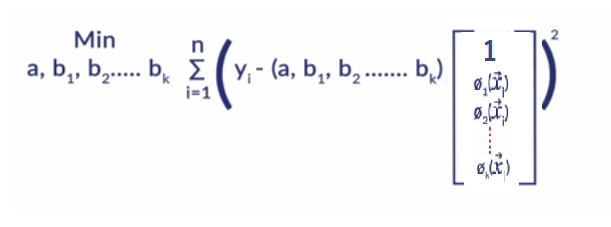
We can express the solution of the regression framework as dot product of 2 vectors - vector with all coefficients vector with all feature values. As shown below, we have done nothing new nor magic.

First, we wrote the same old formulas of regression in the matrix format:



**Figure 1: Regression Matrix**

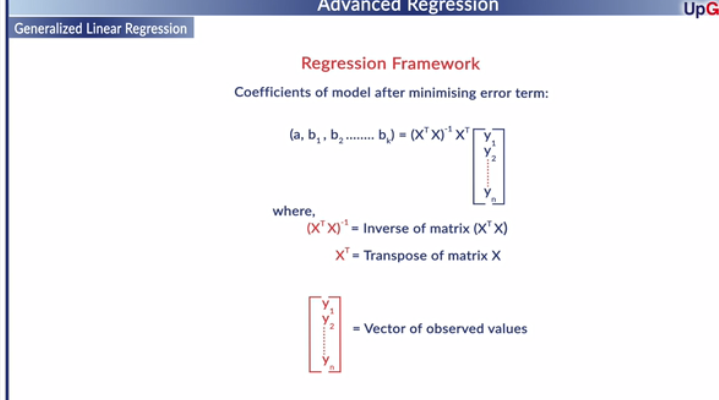
Next, we summed up the errors between predicted and actual response variables and minimized the residual sum of errors:



**Error minimization**

This is because the minimal difference between the predicted and actual values indicates that the model fits well with the data. We solve the problem by identifying the values of constants that makes the model fit the data best.

Till now, we have been exposed to the general framework for solving any regression problem. But the enigma comes when we understand and execute the formula in this context. So let's see how we can implement this:

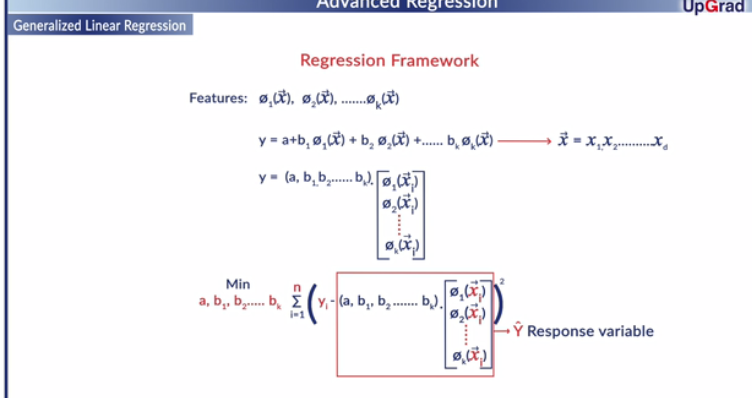


Consider a matrix A with elements aij. What will be the elements of a matrix B =transpose of A with elements bij?  Select the correct option.



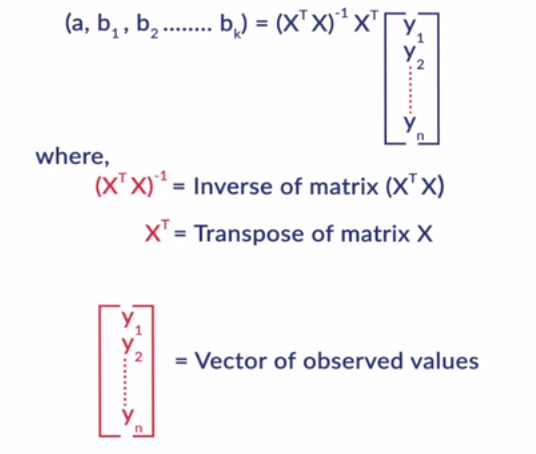
**bij = aji**

**Feedback :***A transpose matrix is formed by turning all the rows of a given matrix into columns and vice-versa keeping the order of rows/columns the same.*

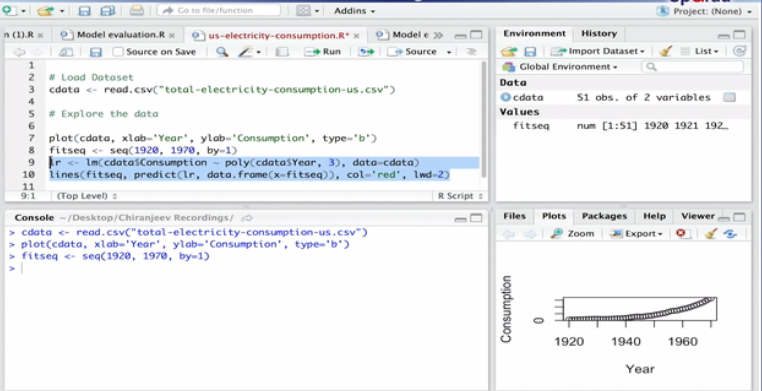


Let us revise this to reinforce our learning.

1. We first created a Feature Matrix, consisting of n number of datapoints in the training dataset as rows and k number of features extracted from raw attributes as columns.
2. We then implemented the formula to identify coefficients that would correspond to the best-fit regression model and minimize the residual sum of errors:



1. **Coefficients of Model**



To summarize, we learnt that one needs to follow a 3-step process to build a regression model:

* First, explore and visualize raw attributes to understand the shape of scatter plots.
* Second, we assessed which function of the explanatory variable would explain the shape of the data.
* Lastly, we wrote the generalized regression formula using the matrix format. We then summed up the errors between predicted and actual response variables and minimised the residual sum of error to arrive at the best fit regression curve

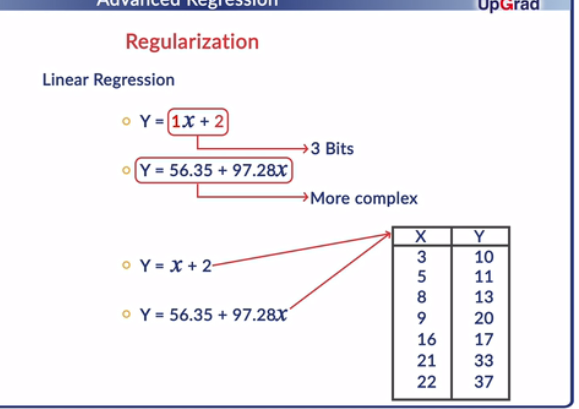
Another thing to note is that the term 'linear' in regression depicts the linear expression in the coefficients of the linear combination. It does not mean linear expression in raw attributes or features.

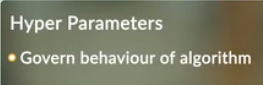
# Concepts of Ridge and Lasso Regression

A predictive model has to be as simple as possible, but no simpler. There is a deep relationship between the complexity of a model and its usefulness in a learning context because of the following reasons:

* Simpler models are usually more generic and are more widely applicable (are generalizable)
* Simpler models require fewer training samples for effective training than the more complex  
  ones

Regularization is the process used in machine learning to deliberately simplify models. Through regularization the algorithm designer tries to strike the delicate balance between keeping the model simple, yet not making it too naive to be of any use. Let us see how regularisation can be applied to linear regression models as Prof. Raghvan walks us through the different regularised regression models.





We have discussed about the two different regularised regression methods - Ridge regression and Lasso regression. Both these methods are used to make the regression model simpler while balancing the 'bias-variance' tradeoff. In ridge regression, an additional term of "sum of the squares of the coefficients" is added to the cost function along with the error term, whereas in case of lasso regression,  a regularisation term of "sum of the absolute value of the coefficients" is added.

**Variable Shrinkage**

Which of the following regularisation methods contributes to variable shrinkage in regression model building?



**Lasso regression**

**Feedback :***As discussed in the lecture, variable reduction is one of the advantages of lasso regression over ridge regression.*

**glmnet command**

What value of alpha should be put in the glmnet() command to perform ridge regression and lasso regression in R respectively?



**0,1**

**Feedback :***To get unregularized behaviour in glmnet(), we need to set lambda=0. For lambda>0, with alpha=1 we get lasso and with alpha=0 we get ridge.*

# Summary

In this session, we were introduced to the concept of regularization in regression models. We discussed at length about the two regularized regression models, namely Ridge and Lasso. The concept of hyper parameter (λ) was also described in the context of regularization, along with its impact on the built model. Further, we learnt about the approach of solving the regularized predictive models in R using the glmnet() command and the concept of cross validation. We discussed the car price prediction problem for carrying out the analysis in R. We evaluated the built model on the basis of mean squared error (MSE) of the test data set, which denotes the average error in the predictions of the model.

**Regularization**

As lambda increases from 0 to infinity, select the correct option that describes the pattern of the residual sum of squares (RSS) of the training dataset.



**Steadily increase**

**Feedback :***Differentiating the cost function with lambda=0 gives the value of the coefficients which minimizes the RSS. Again, putting lamda = infinity gives us a constant model with maximum RSS. Thus, the RSS steadily increases with the variation of lambda.*

**Regularization**

As lambda increases from 0 to infinity, select the correct option that describes the pattern of the variance of the model.



**Steadily decrease**

**Feedback :***When λ=0, the alphas have their least square estimate values. The actual estimates heavily depend on the training data and hence variance is high. As we increase λ, alphas start decreasing and model becomes simpler. In the limiting case of λ approaching infinity, all betas reduce to zero and model predicts a constant and has no variance.*

**Regularization**

As lambda increases from 0 to infinity, select the correct option that describes the pattern of the (squared) bias of the model.



**Steadily increase**

**Feedback :***When λ=0, alphas have their least-square estimate values and hence have the least bias. As λ increases, alphas start reducing towards zero, the model fits less accurately to training data and hence bias increases. In the limiting case of λ approaching infinity, the model predicts a constant and hence bias is maximum.*

Why is multicollinearity a problem in linear regression? Select the correct option.



**Singularity of the matrices that need to inverted for the least squares solution causes instability/break-down of the algorithm**

**Feedback :***Multicollinearity means some of the attributes (features) are linearly dependent. This means, that in feature matrix one or more columns can be written as a linear combination of the other features. As a result, the determinant of the matrix will become 0 causing the matrix to be singular.*

What is the order of the matrix that needs to be inverted for the least squares solution on a dataset having n datapoints each with d explanatory attributes:



**d X d**

**Feedback :***The order of [Transpose(X)] and X are d X n and n X d respectively. Thus, the order of the matrix [Transpose(X)]X is d X d.*

**Generalized Linear Regression**

The term  \small \mathrm{\mathrm{X}^{T}Y}  in the linear regression solution gives:



**Weighted sum of the datapoints**

**Feedback :***[Transpose(X)]Y is just the vector obtained by forming the dot-product of every row of Transpose(X). Also remember that a dot product between say vectors A and B can be thought of as a linear weighted sum of the components of A using the corresponding components of B as the weights.*



**Lasso regression is less flexible than the ordinary least squares regression. Thus, lasso will give improved prediction accuracy when its increase in bias is less than its decrease in variance.**

**Feedback :***Regularisation increases the bias and decreases the variance of a regression model as compared to the unregularised model. Generally during model building, the intention is to make both the bias and variance low. Hence, for a model to have improved accuracy, the increment in bias of the model should be less than the decrement in its variance.*