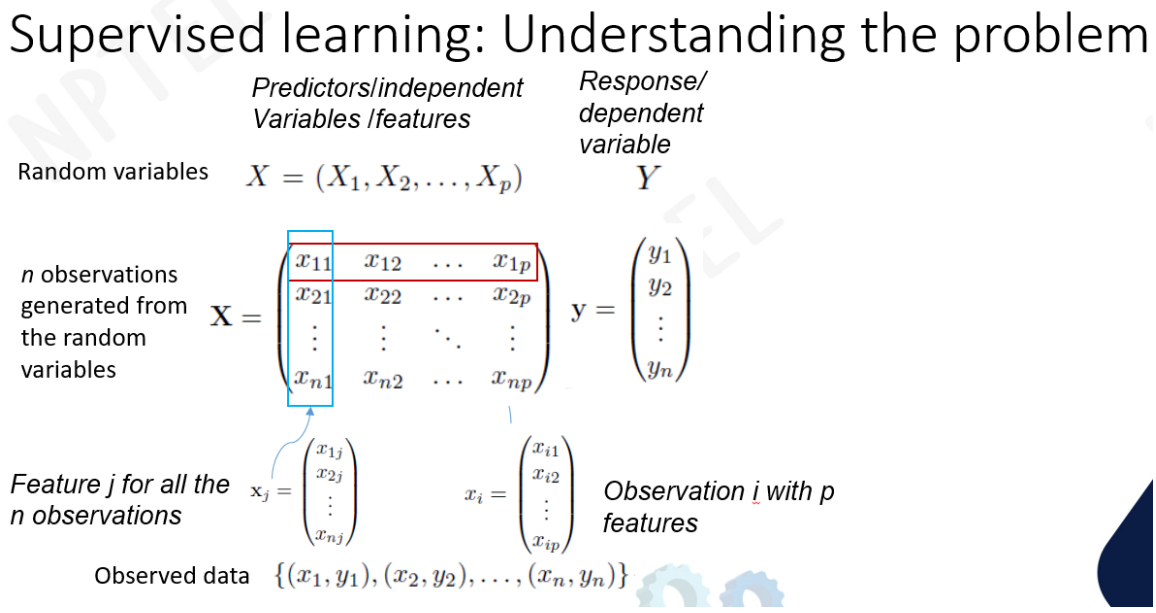
Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 02
Lecture - 08

Lecture 08: Introduction to machine learning-I

Hello everyone. We are going to continue our discussion on Statistical and Machine Learning Foundation. In this context, we have already discussed about PCA and SVD as two approaches for dimensionality reduction. So, in this lecture, we are going to talk about introduction to machine learning and specifically we will be talking about supervised approach on regression. Now, machine learning is the foundation for recommender system. In fact, recommender system is one of the most visible and successful application of machine learning.

So, in the context of recommender system, our idea is to learn how various products are rated by the users. If we remember one of the initial classes, we discussed there are three basic decision making problems in the context of recommender system. The first one is rating the predicting the rating, second is finding the top n top n items which is to be recommended and top m users. Now, here while recommending this and making predictions, you need to find out how to predict the ratings which are not predict the rating for the unrated products by a specific user and based on that you will be suggesting the items.



So, if this machine learning task can be broadly classified into two parts supervised

learning and unsupervised learning. In case of supervised learning which is further classified into regression and classification, we provide examples to the algorithms. So, that algorithm builds the model and based on that model it makes predictions. So, since the this course is on recommender system, it is quite obvious that I will not be talking in detail about machine learning algorithms. However, I will be talking about few fundamental ideas which people can use further when we actually go to discussions on recommender system algorithms.

$$Y = f(X) + \epsilon$$

$$\hat{Y} = \hat{f}(X)$$

$$\begin{aligned} & \mathrm{E}(Y - \hat{Y})^2 \\ = \quad & \mathrm{E}[f(X) + \epsilon - \hat{f}(X)]^2 \\ = \quad & \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}} \end{aligned}$$

So, this supervised method as I told you can be either regression or classification. These are some of the names of regression and classification approaches. Now, start with understanding the problem for supervised learning. In case of supervised learning, the machine has to learn or the algorithm has to learn from the data and what is data. Let us try understanding what the data is.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

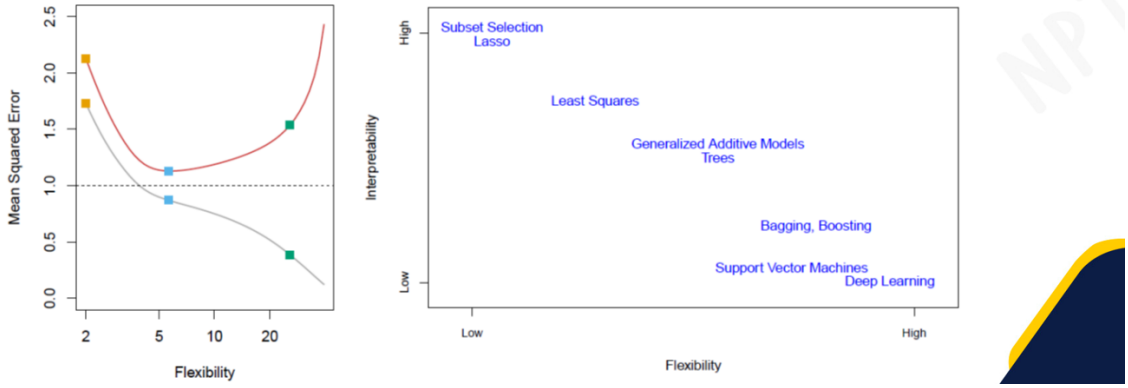
Let us say these X is the observations. Now, let us say X is the data set under consideration. So, what this data set represent? This represent data set represent n number of distinct observations. Let us say these are the n number of people who are the users of the recommender system. Now, all of them have p features.

So, the first user have p features, second user, third user and so on. So, this matrix is represented as a n cross p matrix. Now, let us say the this element is let us say this is age and all these X 1 1, X 2 1, X n 1 represents age of n number of people. Now, this age is

actually observed from this data set, but in future many people will come. How do we know their age? So, this age is actually a random variable.

So, this X 1 is a random variable which is generating these values. X 2 is another random variable which is generating these values. This is another random variable. So, all these are so, basically X is a set of random variables, X is a set of random variables X 1 to X p and all of them are generating this data. As we move ahead and we keep collecting the data Y n more number of data points will come and all of them for distinct features will be getting generated from individual random variables.

Tradeoffs: Quality of Fit, flexibility and interpretability



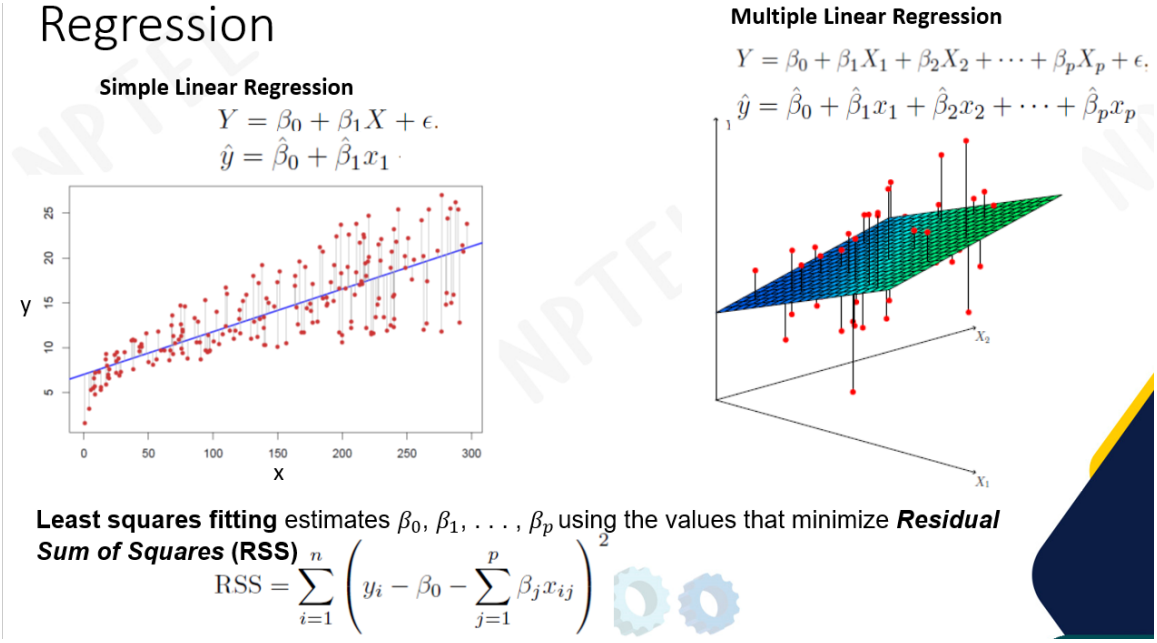
Now, this is the data about the let us say about the user. Now, users will make certain purchase decisions let us say they buy or they do not buy certain items. So, if or they let us say they give some rating. So, for this particular data this will be the choice, for this particular data this will be the choice, this will be the choice and so on. So, in essence in essence we will be representing the data as a matrix where we have n number of data points and p number of features and each of these feature can be represented as a feature vector and each of the data elements can be represented in terms of certain observation X X let us say this is the for the i th for the i th person i th observation let us say i th person as per our example i th person these are the p number of features ok.

So, these are the people features these are the features for all the people and this is this is a specific feature for all the people and this is all the features for one person one observation ok. So, and each of this observation per person is associated with another response or dependent variable. So, this is the first persons data. So, which consists of these elements and this y this and this y x 1 and y 1 x 2 y 2 and so on. Now our purpose is to build a model.

The model that we make will take this input as p features and produce the output as the response variable y. So, here all this feature vector will be the input and response variable will be corresponding y i ok. So, now, the idea is use this pattern which is available to us to build this model in terms of model we will be making a mathematical function. So, now, this function will have many parameters. So, those parameters are to be learned from this example patterns.

Now, because all these are random variables once we make this model and a new observation gets generated from this random variable. If we put this observation I mean the these characteristics of this observation the features of this observation to the model we will be getting certain estimate estimated value of y certain predicted value of the y. So, our aim is to build a model that will be trained from this and for any future observation to predict the value of y. So, this is a typical supervised learning problem that I told you. The supervised models assume that there is some relationship between y and x what is y? y is the response variable and x is the input vector.

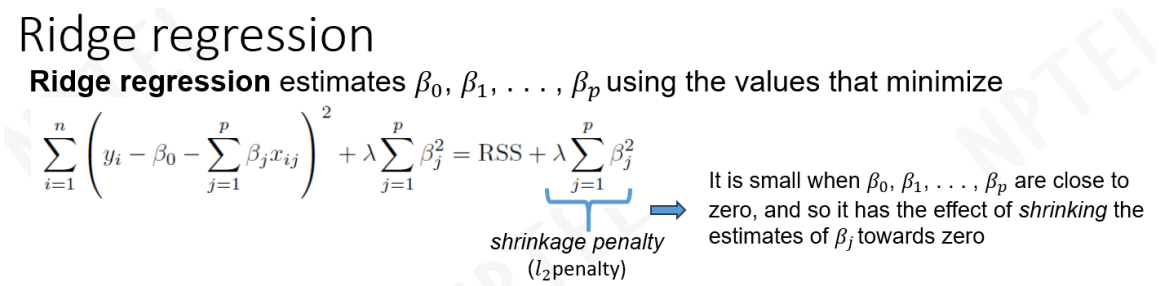
So, this is let us say this is the true relationship ok. So, in this setting the error term averages to 0. So, therefore, one can predict using the function y cap equal to f cap which is a function of x. So, the here f cap represents the estimate for f and y cap represents the resulting prediction for y. Now, in case of supervised technique the estimate f with the aim of minimizing the I mean the we estimate f with the aim of minimizing the reducible error.



Now, what is reducible error? Now, look we represented y as f x plus some error term epsilon. So, this y minus y cap when you try to reduce this f x minus f cap x square is something which we will be able to reduce, but this variability of on this epsilon term is irreducible. So, the estimation of f is has to be done using a set of available pattern which are also called the training data. So, which means whatever we may do there will always be certain error in the model, but we try our best to reduce at least whatever is reducible. So, this reducible part to we try making this function f cap as close as possible to the actual function f.

Now, in case of supervised learning how do we measure the quality of fit? So, what do we mean by quality of fit? Look we said y can be represented by f x plus some epsilon which is an error term and we actually represented a with it with this. So, in terms of actual data

which is available to us we can only find out what is the quality of fit that was in terms of random variable, but when it comes to the data in terms of the actual data how was the data? The data consists of many features called x and one response variable called y. So, this was x 1, x 2 up to some xp number of features. So, this f x is a f x i is actually a function of for the i th person for the i th observation in terms of all these variables which of course, for this i th person it will be x i 1, x i 2, x i p and so on. So, this x i which will be generating some value for y i cap will try taking the square the difference between the square.

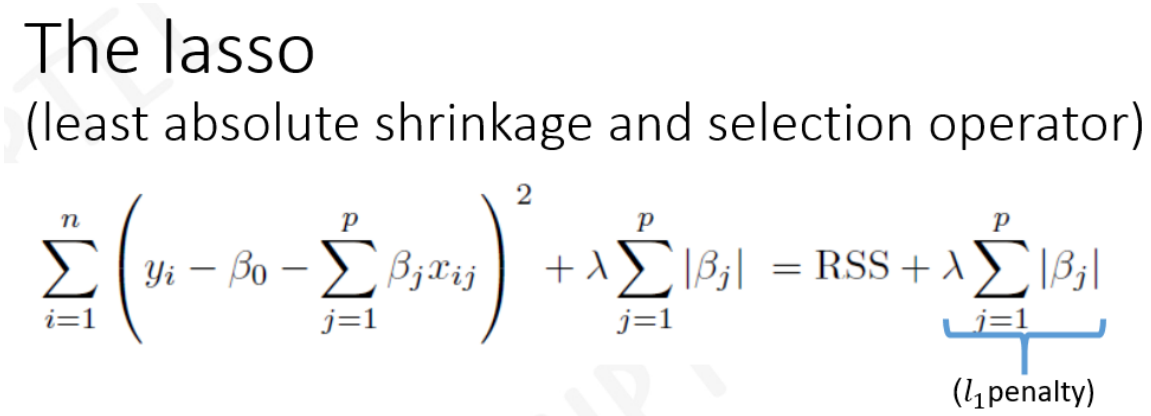


Now, y square because if we do not take square some of the terms will be positive, some of the terms will be negative and they will cancel this. If in fact, my advice would be for machine learning you have to take help of some other book or take certain course, but here I will just telling you few fundamentals which we are going to use later on. So, most commonly used measure for finding this quality of fit is mean squared error. Now this when we try computing this we compute it using the training data. So, when we compute it with the training data and our if our purpose is to make it as low as possible with respect to the training data then it may so happen with respect to the test data this MSE may not be minimum.

So, therefore, we have to keep track of two things one is it has not only it has to be minimized with respect to the available data it should also be minimized with respect to the on scene data. Here when we select the method we have to keep track of two things one is flexibility of a method which is about how well the shape of this fitted function which is the fitted function this f f cap is the fit as function can be adjusted with the data and when we fit it whether we are compromising with the interpretability. What is interpretability? Interpretability is the ability of a method to explain the relationship to draw the inference on the result. Look at this this is what I was talking about the difference between the mean the mean square error the minimizing the mean square error with respect to both training and testing data. Look at this this is the function this is the function with respect to the training data.

So, if we it is possible that we can decrease it further with respect to the training data, but what happens in turn for the test data it slowly increases. So, there is certain point in between let us say this dotted line where we have the optimal set of parameters for which both the training as well as the testing is getting minimized. So, the MSE on the training data may improve with increase in flexibility, but it may increase the MSE with respect to the test data. So, which means as we move on to adopting a more predictable approach it

may so happen that we are able to have this minima mean square error will be able to decrease, but our interpretability is going to be hampered. So, looking at this one shows of course, the methods etcetera we are going to see shortly, but looking at this we can know as the methods like your subset selection using LASSO etcetera the interpretability is very high the flexibility is low whereas, in case of deep learning interpretability because they are complex networks the interpretability is very low whereas, the flexibility is very high and as we see with a more flexible result MSE it is possible to decrease mean square error compared to the others.



So, while deciding a method we have to find the tradeoff between the quality of fit flexibility and interpretability. It depends on the problem whether we want the result to be interpretable or not and are we achieving this interpretability at the cost of losing flexibility and in turn is it hampering our quality of fit. So, we have to choose those methods depending on our application area. Now, come to regression the basic regression is about relating one variable independent variable with that of the response variable independent variable with that of the response variable. So, when we relate it as the name indicates we try fitting one best fit line.

So, now, it is possible that if these red points are our actual data points it is possible that we can have many lines going through them and so on there can be many lines. So, the question is which lines to choose we are supposed to choose that line which will be actually minimizing the error ok. In terms of our regression we call it the residual sum squares which is in turn is the difference between this is the model look this is the model and this is the actual value. So, if you have m number of n number of training patterns with respect to each of the n number of training patterns you take this difference square sum it up and this is called your residual sum of squares. And this is when we have more number of dimensions we have something called multiple linear regression in which this the model is this there is certain random response variable and there are p number of input features.

So, this input features can generate a number of data points and for each of the data points we are supposed to find out build a model which is basically this beta 0 cap plus beta 1 cap x 1 and so on. Where this x 1, x 2, x p etcetera are known are known for a data set and during training whatever y we have based on that we try determining these values of beta 0, beta 1.

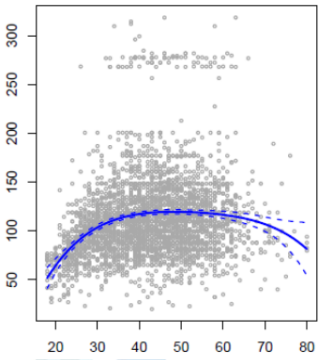
So, these are my model parameters. So, these model parameters I will be determining and how do I determine again following this function I have to minimize this and while minimizing this because this is a linear function I will be now fitting a plane in 3 dimension and in hyper plane in case of multiple dimension. So, you will be now find the best fit in terms of a linear function.

So, that is how the name is linear regression. So, this is simple linear regression this is multiple linear regression with 3 variables x 1, x 2 and x 3. Now, what is the problem with simple linear regression simple and multiple linear regression? We are trying to minimize this part which we call a residual sum of squares. Now, if we do so what is the problem we can minimize and try to fit a model, but we have already discussed the concept of over fitting. So, therefore, while determining these parameters we should be able to penalize them little bit.

Adding non-linearity to regression: The basis function approach

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$$

- The basis functions $b_1(\cdot)$, $b_2(\cdot)$, . . . , $b_K(\cdot)$ are fixed and known.
- Polynomial functions, piecewise constant functions , *splines functions*, wavelets or Fourier series are some of the choices for basis functions

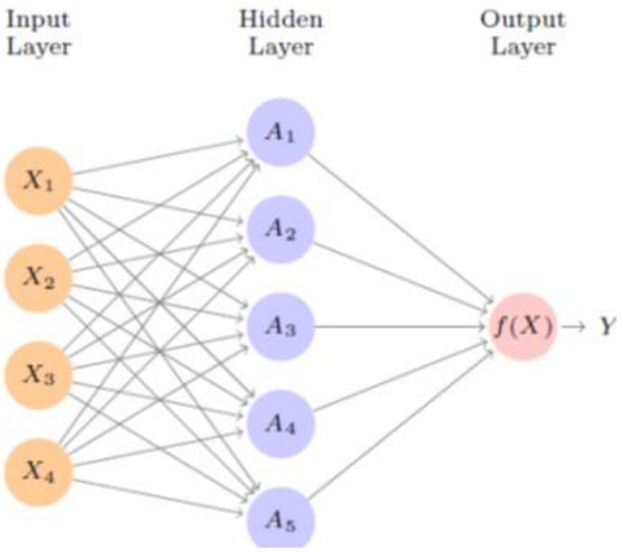


A degree-4 polynomial fit

So, that our while determining them we are not over fitting. So, this penalty is called shrinkage penalty ok. So, shrinkage penalty consists of 2 terms one is square of the parameters and there is a term lambda. So, this lambda is also called the regularization parameter or tuning parameter. Now, while we try determining try minimizing this error we use this tuning parameter and try adjusting the values of this model parameters beta.

So, that we do not over fit. So, this tuning parameter lambda which is also called the regularization parameter serves to control the relative impact of the terms on the regression coefficient estimates accordingly. This ridge regression will produce different sets of coefficient estimates therefore, selecting the right value of lambda is also critical. Now, this ridge regression will include all p predictors in the final model. All the p predictor means means this beta values. So, all these beta values will be there in the final model.

So, this may not be a problem in terms of making having a good fit and giving a improved prediction accuracy, but sometimes it creates problem in terms of interpretability specifically when value of p is quite large. So, what do we do? We can use another type of penalty which is called the least absolute shrinkage and selection operator the LASSO, where the penalty function is not the square of the model parameters rather it is the mod of the model parameters. So, as the as with the ridge regression the LASSO also shrinks the coefficient estimates towards 0. However, in case of LASSO this is your L 1 penalty. Now, L 1 penalty has the effect of forcing some of the coefficient estimates to exactly 0.



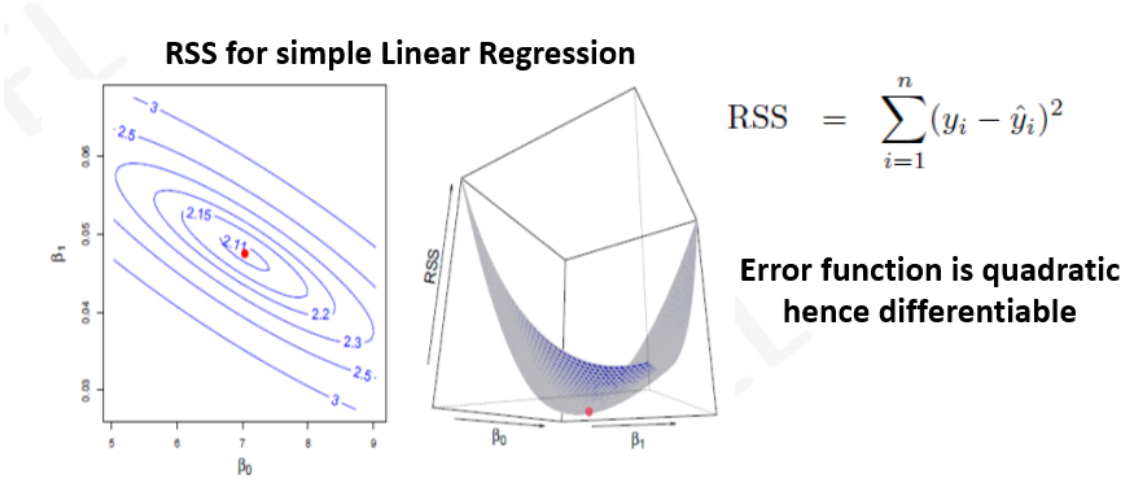
$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X)$$
$$= \beta_0 + \sum_{k=1}^{K} \beta_k g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j)$$

Typically, when lambda is 0 all of them of course, I mean the will be 0, but with different values of lambda also it is possible I mean the with when the lambda becomes very large when the lambda becomes sufficiently large then many of these beta coefficients can also become 0. So, thus LASSO performs variable selection and improves interpretability. Now, what is this variable selection? Now, out of look when we have total p number of variables it may not be necessary to keep all the variables some of these variables are will not be very important we have to maybe remove them. So, therefore, with this when we take this L 1 penalty then many of the beta value variable automatically are forced to 0. So, when they are forced to 0 those variables are automatically eliminated.

So, whatever remains those are the variables which are important and thus using LASSO can also improve model interpretability. Now, when it we talk about the regression so far which is a supervised learning technique which was trying to fit a linear function a line precisely a line in case of 1 dimensional problem and a plane in case of a 2 dimensional problem and a hyper plane in case of a higher dimensional problem. Now, it may so happen

that actually the true relationship among the variable may not be linear. So, therefore, we are supposed to take care of this nonlinearity by extending regression problem. So, one of the way to extend at nonlinearity to the regression problem is to use the basis functions.

So, if you look at this now y i is modeled in terms of many functions that time in case of regression your multiple linear regression beta 1 was getting multiplied with x 1 beta 2 was getting multiplied with x 2 the second component of this. So, now, here instead of using directly the variable we are using a function of that particular variable. Now, see x i was basically x i 1 up to x i p. So, when we had simple linear regression and multiple linear regression beta 1 x 1 beta 2 x 2 up to beta p x p what was x p? x p was the x i was basically having p number of attributes. So, here using all the attributes using all the attributes as the input to this function we are building this function.



So, how many such functions depends? So, you here in this particular case there are k such functions. So, in each of this k such functions x i becomes input. So, using this basis functions b 1 to b k we can actually make a non-linear all these functions are non-linear. So, you can make a non-linear fit to the data. So, the popular choices for this are polynomial functions, piecewise constant functions, splice functions, wavelets and Fourier series and so on.

Now, we can also imagine neural networks as an extension to the regression problem in which we are adding further non-linearity by making a network ok by making a network. In case of simple linear regression what was happening? We were also adding them up. So, we had input variables x 1 to x 1 to x p and all of them were getting multiplied with certain beta 1 to beta p and we were summing them up and getting y. Now, this network which where we were simply trying to discover this betas now we will be adding more layers to this. So, in a typical neural network there will be multiple layers and in this particular example we show only one hidden layer.

So, originally the data was this and getting mapped to this hidden layer was this hidden layer was not there in case of a regression problem, but now we have to add a hidden layer. So, consequently if we have k elements in this hidden layer then our model essentially

becomes beta 0 plus summed over k equal to 1 to capital K beta k h k x where this h k x this x is actually the input. So, this input is again getting multiplied with we had p dimensional vectors. So, each of them was getting multiplied with a weight and that weight after getting multiplied with the weight there was certain bias and this sum was fed to a function and this value of this function was getting multiplied with beta. So, essentially now through this function g you are introducing non-linearity in the network ok.

And moreover for when you have multiple layers this this function that we saw just now can be extended in the similar manner. Now, here the hidden layer computes the activation a k which is h k x that are the non-linear transformation to the linear combination of the inputs x 1 to x p. Hence this a k are not directly observed. The function h k are not fixed in advance, but are learned during the training of the network. The output in a linear model that uses this activation a k as input results in a function f x.

So, typically this ReLU that is rectified linear unit and sigmoid functions are some of the choices for activation functions. So, all these parameters beta 0 to beta k and the weights are need to be estimated from the data ok. And as we have to have a good model we are supposed to minimize the error here also we try minimizing the error, but because the network is very complex directly minimizing the error may not be possible. So, therefore, we have good algorithms like that of back propagation algorithm. In fact, there are many more algorithms as we move ahead add more number of layers and make the network dense we have much better algorithms which can reduce this error from the when we try connecting this input to the output whatever is the error that gets back propagated and accordingly the weights are update weights and biases are updated.

Now, when you we talk about minimizing the error function the basis of minimizing the error lies in the gradient descent algorithm. The idea of gradient descent algorithm is to let us say this is our error function this is our error function where we are supposed to determine this right value of beta 1 and beta 2 which minimizes this. We have to in case it is a continuous and this differentiable function we have to move towards this point ok. So, here the method aims to find the local optima of differentiable function of a differentiable function f the gradient gives the direction of the greatest increase and negative gradient gives the direction of the greatest decrease our aim is to decrease the error. So, we will be moving in the opposite direction that of the gradient.

The algorithm starts at a random point and takes steps in the direction that reduce the function value. Now, definition of derivative guarantees that if a small step is taken in the direction of the negative gradient the function will decrease in value. In case of multivariable functions the negative of the partial derivatives if the direction of the maximum decrease the model parameter update equations are developed by setting the partial derivative to 0 and solving the equations. So, once this model parameter model parameter update equation are ready you can use them in term using an iterative algorithms and minimize the error and get a good model by updating the parameter values in each iteration.

Now, when the number of variables are very large. So, number of parameter update equations also increase. So, as a result what happens when the data set is large computing this gradient descent in one go becomes very computationally expensive. So, therefore, most recently most of the machine learning algorithms use a version of gradient descent which is much faster is called stochastic gradient descent. So, in this case of stochastic gradient descent instead of instead of considering all the observations at a time you take only few observations at a time and with respect to that you run your gradient descent algorithms. And because you stochastically select this m observation at a time the name is called stochastic gradient descent.

So, these are few references and all the examples that I have covered is from the first book that is introduction to a statistical learning with applications in python it is a very good book and it can build a very good foundation. So, if you wish you can read this book there are a lot of solved examples in this and this is now we conclude machine learning algorithms can be classified into supervised and unsupervised and reinforcement learning. And regression and classification are two broad approaches under supervised learning simple and multiple regression assumes linear relationship of the output variable with respect to input variable. And we can improve we can deal with the problem of over fitting by penalizing the penalizing the parameters in the in the error equation and reach and lasso regression intend to do that. In case of reach regression regression, we use L 2 norm and whereas, in case of lasso regression we use L 1 norm.

So, the basis function approach and neural network tries capturing the non-linearity in relationship and in an iterative where in case of basis function you have to predefine the function in case of neural network the functions are to be learnt. The error function in regression can iteratively improve the solution using gradient descent algorithm and to make this gradient descent algorithms fast you can use a more flexible version called stochastic gradient descent which takes not all observations at a time, but takes few observations at a time and try working on that. So, with this our first lecture lecture towards the foundation for machine learning is over. Thank you very much.