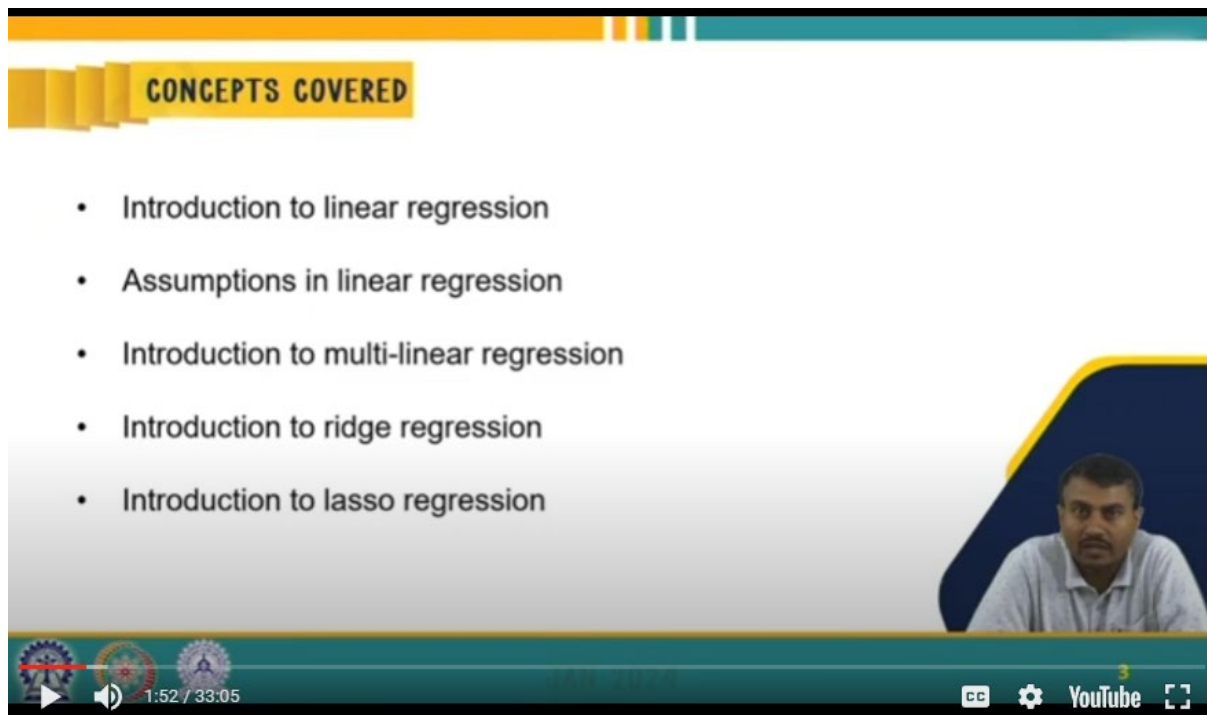


Mine Automation and Data Analytics  
Prof. Radhakanta Koner  
Department of Mining Engineering  
IIT (ISM) Dhanbad  
Week - 10  
Lecture - 46  
Regression

Welcome back to my course on Mine Automation and Data Analytics. Today, we will discuss regression in this lesson. In the last lecture, you saw that we are dealing with regression and classification in machine learning, mainly supervised learning. Regression is one of the most used models we use daily in the machine learning community, as well as in statistical methods. So, in this lesson, we will focus on a few types of regressions. Next, we will proceed with different types of regression.



**CONCEPTS COVERED**

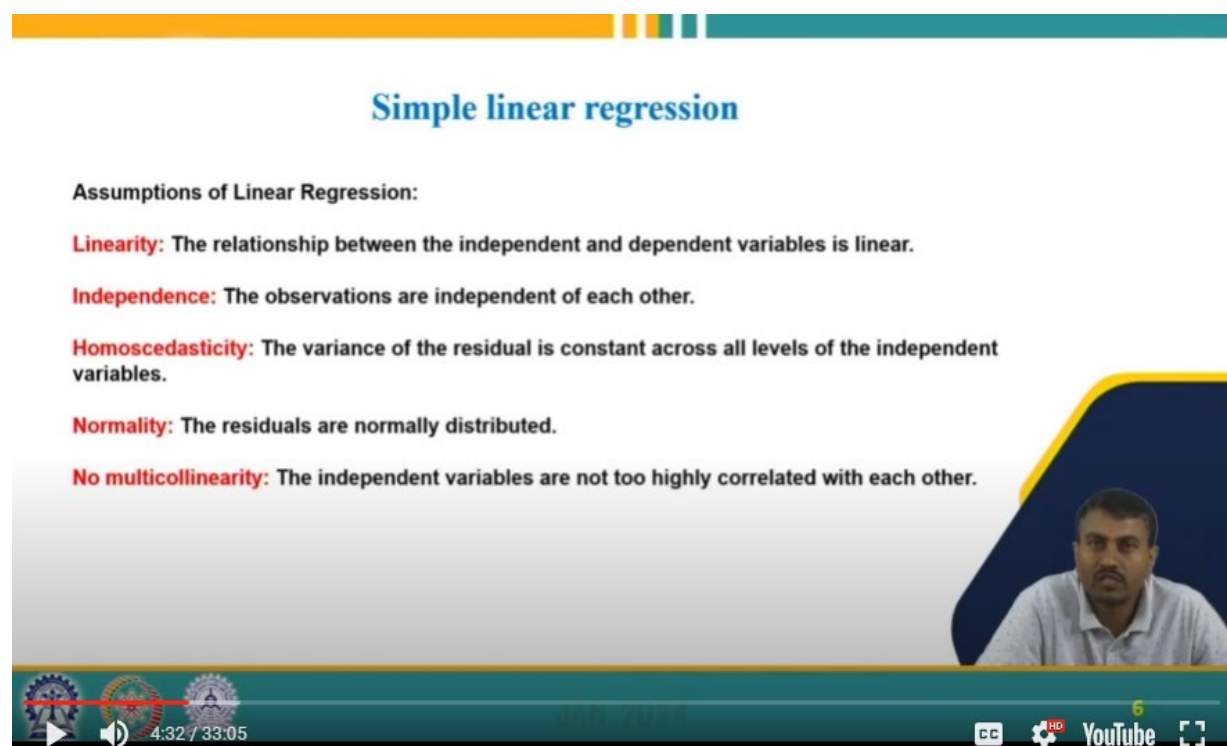
- Introduction to linear regression
- Assumptions in linear regression
- Introduction to multi-linear regression
- Introduction to ridge regression
- Introduction to lasso regression

So, in today's lesson, we will discuss linear regression, then the assumption in linear regression, and we will also introduce multilinear regression and the associated regularization to deal with overfitting, that is, ridge regression and lasso regression. We will discuss both these regularization methods. So, let's start with the simple or linear regression models. As we have already seen, the most famous example is  $y$ , which equals  $mx$  plus  $b$ .

In this,  $y = mx + b$ ,  $y$  is the dependent variable,  $x$  is the independent variable, and  $m$  is the slope of the line representing the relationship between  $x$  and  $y$ . And  $b$  is the intercept on the  $y$ -axis. So here we are predicting something from the input value  $x$ . We predict the dependent variable based on a specific set of  $x$  data. So, this is one of the simplest forms of the linear regression model.

And popularly, if you see it in the market survey, rock mechanics, and finance, these have a perfect application of these regression methods. So today, we will limit our discussion to linear and multiple linear regression. So, in simple linear regression, only one independent variable exists. In multiple linear regression, we have multiple independent variables. So, in our linear, only  $y = mx + c$ , where  $x$  is the single independent variable.

Whereas for the multiple linear regression, there would be  $x_1, x_2, x_3, x_4$ . Up to the number that is prevalent in that particular case. So, let's start with the assumption of linear regression because whenever we deal with some model, that model has some assumption. These assumptions are fundamental to follow, and we have to satisfy these assumptions during the construction of the model. So, the first assumption is linearity.



### Simple linear regression

**Assumptions of Linear Regression:**

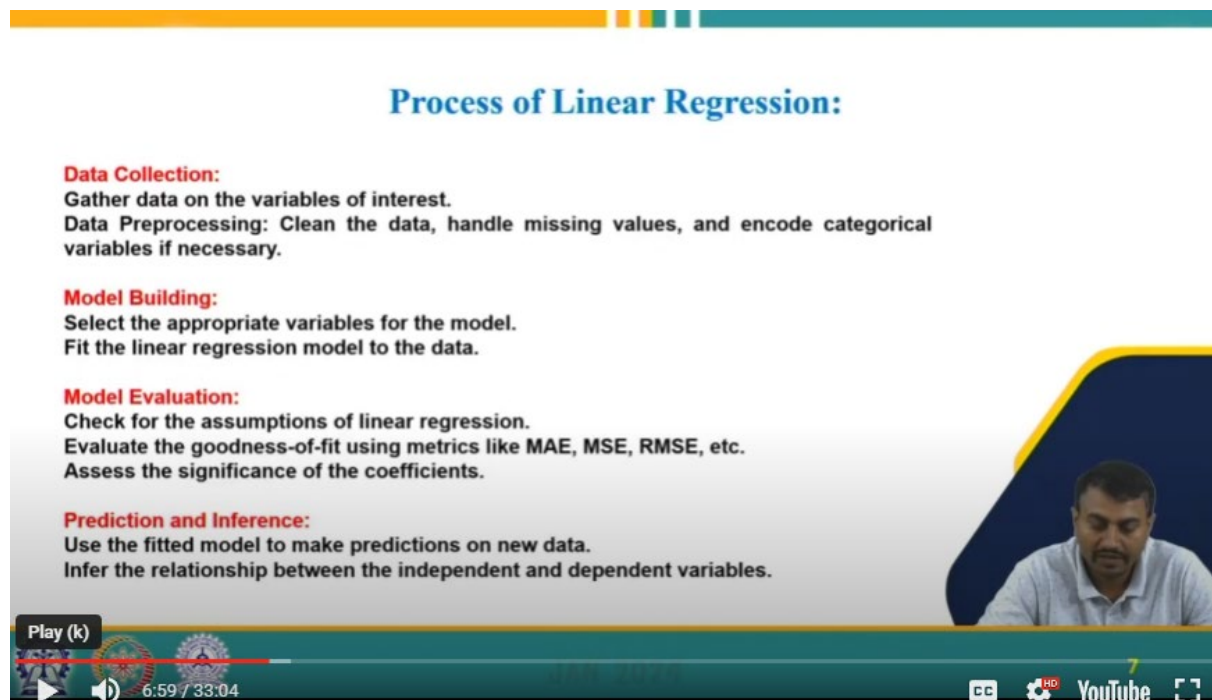
- Linearity:** The relationship between the independent and dependent variables is linear.
- Independence:** The observations are independent of each other.
- Homoscedasticity:** The variance of the residual is constant across all levels of the independent variables.
- Normality:** The residuals are normally distributed.
- No multicollinearity:** The independent variables are not too highly correlated with each other.

This says that the dependent variable and independent variable are linear, we are assuming. If it is not linear, we cannot apply this simple linear regression method to that data set. So this is the assumption number one. The second assumption is independence. The observations are independent of each other.

This means one observation does not affect the other observation. And subsequently, so forth. So, we have to ensure that. Third is the homoscedasticity (i.e., variance is uniform) ; the residual variance is constant across all levels of the independent variable. Another way of saying this is that the variance of the data points is roughly the same for all data points. Another consideration that we have to follow is normality.

The residuals are normally distributed. In this regard, residuals are normally distributed. And we have to ensure no multi-collinearity. That is, independent variables are not too highly correlated with each other. We have to ensure that x data or different data are not within themselves or are not highly correlated.

We have to ensure that these five assumptions exist, and we have to satisfy them. Based on that, we can apply them. We may need some help, particularly in predicting y in the data set of x like that. So, the process of linear regression. First is the data collection. This involves the data concerning the variable of interest we are measuring or are interested in dealing with.



### Process of Linear Regression:

- Data Collection:**  
Gather data on the variables of interest.
- Data Preprocessing:** Clean the data, handle missing values, and encode categorical variables if necessary.
- Model Building:**  
Select the appropriate variables for the model.  
Fit the linear regression model to the data.
- Model Evaluation:**  
Check for the assumptions of linear regression.  
Evaluate the goodness-of-fit using metrics like MAE, MSE, RMSE, etc.  
Assess the significance of the coefficients.
- Prediction and Inference:**  
Use the fitted model to make predictions on new data.  
Infer the relationship between the independent and dependent variables.

Then, based on that data collection, if it is mining, we have to understand that these data have a good amount of noise and some missing data. And then, we have to level it; we have to encode it to certain variables so data pre-processing is required on the data set. So, we have to deal with data cleaning, handling the missing values, and all these things at the first stage. The second is the model building. Based on this data, we have to build a linear model.

Okay. We also have to fit the linear regression model. Then the third is the model evaluation. We have to check because, in the first model, we have seen data, then ML, then prediction, and here we have evaluation. So, we have to evaluate the model. So, based on the prediction from the input data, we have to check with the corresponding output data.

Then, we have to calculate the MAE, MSE or RMSE, which are all the things that will be used to evaluate the model performance. Based on this model performance, the values of M and C are to be scaled to fit the data. Okay. We must also fine-tune this value M and C for linear model Y, which equals MX plus C. So once M and C are evaluated correctly and based on this

model evaluation, we can fit that data, we can fit that model in new data, and we can build the model  $Y$ , equal to  $MX$  plus  $C$ .

$$y = mx + b$$

$M$  dash and  $C$  dash are the new values we get from the data. So this indicates finally the relationship. So here,  $Y$  is the dependent variable, and  $X$  is the independent variable, and we should have a good amount of data, so  $X_i$  starts from 1 and goes to  $N$ , which is the  $N$  number of the data set we have. Here, we are using the same statistical method we have already dealt with in this model to get a better  $Y$  equal to the  $MX$  plus  $C$  model, or here,  $M$  and  $B$  are the constants that we have to predict. So, instead of  $C$ , it is  $B$ , and  $B$  is the interceptor.

Now, the MSE, mean squared error, because this is one of the criteria for evaluating the model. So, in MSE, the mean squared error is the squared difference of the level output versus the predicted output. So,  $Y_i$  minus  $\hat{Y}$  is the expected value. So the squared difference is the  $Y$  equals one by  $N$  divided by  $N$ , which is the MSE, mean squared error. So, we calculate this mean squared error to see this model's performance, which is one of the criteria for evaluating the model.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - (mx_i + b))^2$$

So, from this MSE, we can also go to the following model criteria for calculating the root mean squared error. This is the root over of this particular data. So here we are, concentrating on getting a lesser or minimum MSE, okay? We are achieving, we are targeting. So to achieve that, we need to fine-tune the  $M$  and  $B$  okay.

$$m = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$b = \bar{y} - m\bar{x}$$

We must follow the least square estimate of  $M$  and  $B$  to fine-tune the  $M$  and  $B$ . So,  $M$  is to be calculated like this: the summation of  $i$  equals 1 to  $N$  over  $X_i$  minus  $\bar{X}$  into  $Y_i$  minus  $\bar{Y}$  bar divided by  $X_i$  minus  $\bar{X}$  bar. The whole square from  $i$  is equal to 1 to  $N$ . And  $B$  is equal to the  $\bar{Y}$  bar minus the  $M \bar{X}$  bar. Okay. So, the  $\bar{X}$  bar is the mean of the independent variable, the  $\bar{Y}$  bar is the mean of the dependent variable.  $B$  is the interceptor we are determining.

So from this data, you can see that this particular model is  $Y$  is equal to 1, 2, 3, 4, 5,  $Y \times X$  is equal to 1, 2, 3, 4, 5, and  $Y$  is equal to 2, 3, 4, 5, 6. So, this indicates that  $Y$  is equal to  $X$  plus

1. So, M is equal to 1, and C is equal to 1. M is equal to 1, which means the gradient is 1. So, the gradient is 45 degrees.

It intercepts at one on the Y-axis. So this is a data. Blue dots are the data; we predict the line fitting and match the data with the predicted or dependent variable. Now, there are three kinds of evaluation matrix to evaluate the model. One that we have already shown you is mean squared error, which measures the average of the square of the errors that are residuals, and the square root of the MSE is the root mean squared error.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Another matrix is the mean absolute error. The average of the absolute error is the mean absolute error. So, these values are crucial for assessing the quality of the model—the quality of the model, its interpretability, and its predictability on the input data. So, the evaluation metric MAE is nothing but summing over  $Y_i$  minus  $\hat{Y}_i$  of mod from  $i$  equals 1 to  $n$ , summing over divided by  $n$ .

So, the  $\hat{Y}_i$  represents the predicted value of the target variable for the  $i$ th data set.  $Y_i$  is the actual value of the target value variable of the  $i$ th data set. So, this MAE gives an equal weightage to all errors regardless of the magnitude. It provides a more straightforward interpretation since it is the same unit as the target variable because  $Y$  is the target variable, which is also the difference; the absolute difference between these two is that we are calculating that it is the same unit as the target variable. MSE, here we are heavily penalizing the errors.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Because of the error difference, this is the actual value, this is the predicted value, the difference, and then it is squared. We are penalizing the errors heavily because of the squaring of operation. So, the lower the MSE value, the better the model performance. That is what we expect, and we rely on this particular matrix.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

RMSE, root mean squared error. So here again, when it is squared, it is root over, so the RMSE and the predicted variable of the same unit are okay. It is one advantage in terms of interpretability and is the most commonly used metric. The goal is to understand the magnitude of the error on the same scale as the target variable. So, in summary, in the evaluation matrix, MSE is the average of the squared error, penalizing more significantly for the mistakes or penalizing heavily for the errors. The mean absolute error provides a more interpretable measure, and RMSE is the square root of the MSE, providing interpretable measures in the same scale as the target variable. So, we have to choose which matrix we will use for assessing the performance of the model based on the performance; based on the data, we calculate and see the pattern of the errors and the residuals.

Now, relying on this foundation, we want to jump to the next stage. That is an extension of the linear regression, which is the polynomial regression, extending linear regression to accommodate the polynomial relationship. Then another is ridge regression, and LASO regression is a technique for regularization to prevent overfitting. Overfitting is a problem in linear regression or regression. Logistic regression is used for the binary classification problem and generalized linear model GLM extension of linear regression to accommodate non-normally distributed dependent variables.

So, this linear regression is a widely used method in economics, finance, biology, social science, and engineering for modeling and predicting tasks. So, we will also examine some of the applications of this linear regression model. Multiple linear regression. So, multiple linear regression extends the simple linear regression model. So here, instead of only a single independent variable  $X$ , we have several independent variables  $k$  and  $X_1, 2, 3, 4, 5$  up to  $k$   $X_k$ .

$$\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k$$

So, the expression remains the same.  $\hat{y}$  is equal to, or  $\hat{y}$  is equal to,  $b_0$  plus  $b_1 X_1$   $b_2 X_2$  up to  $b_k X_k$ . So,  $\hat{y}$  is the dependent variable's predicted value,  $b_0$  is the Y-intercept, and  $b_1$  to  $b_k$  is the coefficient associated with the independent variable.  $X_1$   $X_2$  up to  $X_k$  is the independent variable. So here, also in the multiple linear regression, we have to follow the same pattern.

First is the data collection. So, we have to gather the data on the dependent and independent variables. We have to collect these data sets. Then, for data processing, we have to clean the data, handle the missing value, and encode the categorical value if necessary. The model is fitting. We have to select the appropriate independent variable for the model.

Or fit the model to the training data using ordinary least square methods or gradient descent. Then, we have to evaluate the model. We have already discussed some of the metrics MSE and RMSE. Based on that, we have to assess the significance of this coefficient and its impact on the interpretability of the variable to the independent variable. Based on that, we have to finally predict and build a model that will predict in a new data set.

The multilinear regression assumes that the relationship between the independent and the dependent variable is linear. Independence and the observations are independent of each other. One observation does not affect the different observations. The homoscedasticity, the variance of the residuals, is constant across all levels of the independent variable.

Normality: the residuals are normally distributed. There is no multicollinearity; the independent variables are not too highly correlated. So, these are the five assumptions we have also seen in the simple linear regression method. So, multiple linear regression aims to find the value of  $b_0$ ,  $b_1$ , and  $b_2$  up to  $b_k$ , which minimizes the difference between the predicted value  $\hat{y}$  and the actual value  $y$  from the data set. So, this  $b$  is calculated in a matrix format.  $x^T x$  of inverse multiple by  $x^T$  into  $y$ .

$$\mathbf{b} = (X^T X)^{-1} X^T \mathbf{y}$$

So  $y$  is the vector of the dependent variable,  $x$  is the matrix of the independent variable, and  $b$  is the vector of coefficients. Okay. So, one of the popular packages is Scikit-Learn in Python, and the R and R statistical tools can also be used to calculate these  $b_0$ ,  $b_1$ ,  $b_2$ , and  $b_k$  for a multilinear regression model. This can be efficiently done using the Python package and Scikit-Learn. Now, one of the problems we have, assuming you have seen no multicollinearity, is that independent variables are not too highly correlated.

**Ridge regression**

Ridge regression, also known as Tikhonov regularization, is a linear regression technique that extends ordinary least squares (OLS) regression by adding a regularization term. The regularization term helps prevent overfitting, especially in situations where there is multicollinearity (high correlation) among the independent variables.

In ridge regression, the standard OLS objective function is modified to include a penalty term that discourages large coefficients. The objective function for ridge regression is given by:

$$\text{minimize } J(\mathbf{b}) = \sum_{i=1}^n (y_i - \mathbf{b}^T \mathbf{x}_i)^2 + \lambda \sum_{j=1}^p b_j^2$$

where:

- $J(\mathbf{b})$  is the objective function to be minimized,
- $\mathbf{b}$  is the vector of regression coefficients,
- $\mathbf{x}_i$  is the vector of predictor variables for the  $i^{\text{th}}$  observation,
- $y_i$  is the observed value for the  $i^{\text{th}}$  observation,
- $p$  is the number of predictors,
- $\lambda$  is the regularization parameter (also known as the tuning parameter or shrinkage parameter).

23

26:28 / 33:04

CC BY YouTube

But if it is so, if the multicollinearity assumption is violated, there is a high correlation between the independent variables, which can lead to an unstable coefficient that will not predict in a new data set efficiently. So, the regularization technique is one of the most popular methods for dealing with this situation. So, we will discuss two methods today. One is the ridge regression, and the other is the Lasso regression, which mitigates the effect of the multicollinearity between the independent variables and the regularization of the dependent variables. So, it is a very efficient method with a high-efficiency level in the regression model and machine learning algorithms.

So, let us discuss the ridge regression. So, this ridge regression, also known as Tikhonov regularization, is a linear regression technique that extends to an ordinary least square

regression by adding a regularization term. So this is the regularization term  $\lambda$ , summation of  $j$  equals 1 to  $p$ ,  $b_j$  squared. So  $b_j$  squared, summing over  $j$  equals 1 to  $p$  into  $\lambda$ .

$$\text{minimize } J(\mathbf{b}) = \sum_{i=1}^n (y_i - \mathbf{b}^T \mathbf{x}_i)^2 + \lambda \sum_{j=1}^p b_j^2$$

So,  $p$  is the number of predictors. Okay. And it is similar to  $y_i$  minus  $\mathbf{b}^T \mathbf{x}_i$  whole squared. So these Jacobian, we want to minimize  $J(\mathbf{b})$ . So,  $J(\mathbf{b})$  is the objective function to be minimized, and  $\mathbf{b}$  is the regression coefficient vector.  $\mathbf{x}_i$  is the vector of the predictor variable of the  $i$ th observation, and  $y_i$  is the observed value.  $p$  is the number of predictors, and  $\lambda$  is the regularization parameter, also known as tuning or shrinkage parameters.

$$\lambda \sum_{j=1}^p b_j^2$$

Okay. So, we are including a penalty term here in the expression that we are minimizing this so that the effect of the multiple linearity between the independent variable impact on the prediction and predicted variable is reduced. So this term,  $\lambda \sum_{j=1}^p b_j^2$ , summing over  $j$  is equal to 1 to  $p$ , is the regulation term, and these regulation parameters  $\lambda$  controls the strength of the regulation. If the  $\lambda$  is 0, it is the same as the ordinary least-squared method. So, as the  $\lambda$  increases, the impact of the regulation term becomes more significant. The reach regression solution can be obtained by minimizing the objective function, Jacobian  $J(\mathbf{b})$ , using techniques like gradient descent or linear algebra.

So, this regulation term tends to shrink the coefficient to 0, which can help prevent overfitting by penalizing the significant coefficients. Okay. So this can be done in the Python package and in `glm net` in the R package. So, the advantage of ridge regression is that it mitigates the overfitting phenomena. It improves the stability of the model, and it works well with high-dimensional data.

So, this ridge regression is a valuable tool in machine learning for improving linear regression models' stability and generalization performance, especially when multicollinearity is present. So, by adding a penalty term to the cost function, reach regression effectively balances the trade-off between the bias and variance, leading to more robust and interpretable models. So, let us examine another regulation method, LASSO regression. So, this LASSO regression, which stands for least absolute shrinkage and selection operator, is another regulation technique used in linear regression. So, similar to the ridge regression we already discussed, LASSO regression also helps mitigate the problem of multicollinearity and prevent overfitting.

However, LASSO regression introduces a different type of penalty that can lead to sparsity in the resulting model. So, multicollinearity occurs when independent variables in a regression model are highly correlated. This can lead to instability in the estimate, which decreases the



model's interpretability. This problem can be solved using LASSO and ridge regression. So, the LASSO regression addresses the multicollinearity issue by adding a penalty term to the cost function of the linear regression model.

$$\text{Cost}(w) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^p |w_j|$$

So, the penalty term in LASSO regression is the L1 norm of the coefficient vector. So the cost function of  $w$  equals  $y_i$  minus  $\hat{y}_i$  whole square summing over  $i$  equals 1 to  $n$  plus  $w_j$  mod summing over  $j$  equals 1 to  $p$  multiplied by  $\alpha$ . So,  $\alpha$  is the regulation parameter here. It is a LASSO parameter or penalty parameter which controls the strength of the regulation. So, the more extensive the value  $\alpha$ , the more significant the shrinkage of the coefficient towards 0.

And  $y_i$  is the dependent variable. This is the predicted dependent variable based on the set of data we are using. So, the sparsity in LASSO regression, unlike ridge regression with a shrink coefficient towards 0, LASSO regression has the property of producing a sparse model. LASSO can select a subset of the most crucial feature by driving some coefficient to precisely 0. So, in other words, the LASSO can perform feature selection as a part of the model fitting process. The advantages of LASSO regression: One of the significant advantages is feature selection, which is the most critical feature, and the rest of the features' coefficients are equal to 0.

It handles high-dimensional data and improves the model's interpretability. So, this is one of the most essential advantages of the LASSO regression. So, by introducing a penalty based on the L1 norm, the coefficient of the LASSO regression encourages sparsity in the resulting model, leading to more straightforward and more efficient models that are easier to interpret. Ultimately, we are building a model for better interpretability.

These are the references. So, we have covered in this lecture linear regression, then multiple regression, their assumptions, and how to deal with the multicollinearity within the dependent variables by introducing the regularization that is RIDGE regression and the LASSO regression. Thank you.