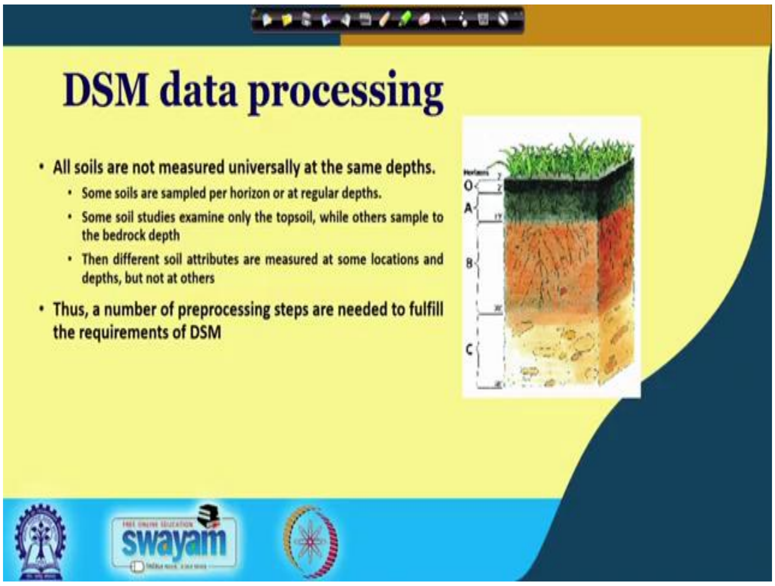
# Soil Science and Technology
Prof. Somsubhra Chakraborthy
Department of Agriculture and Food Engineering
Indian Institute of Technology, Kharagpur

## Lecture – 57
## Modeling Continuous Variables

(Refer Slide Time: 00:28)



DSM data processing

- All soils are not measured universally at the same depths.
  - Some soils are sampled per horizon or at regular depths.
  - Some soil studies examine only the topsoil, while others sample to the bedrock depth
  - Then different soil attributes are measured at some locations and depths, but not at others
- Thus, a number of preprocessing steps are needed to fulfill the requirements of DSM

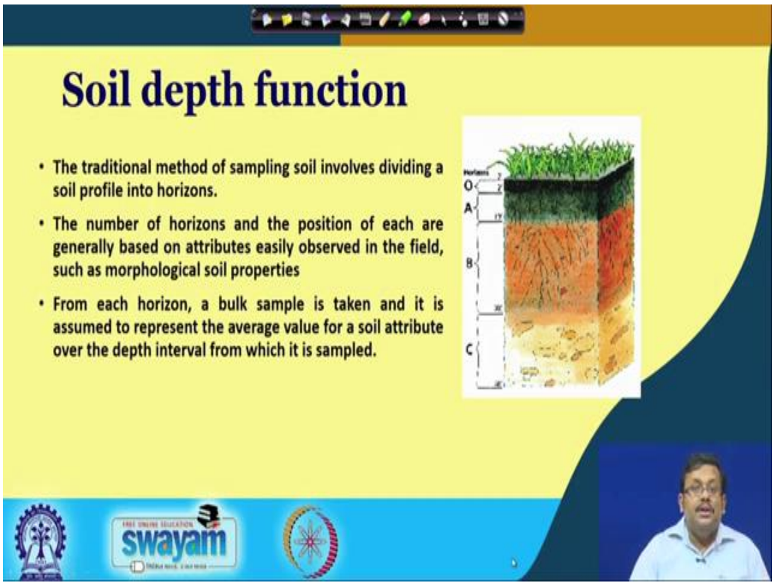
Welcome friends to this second lecture of week 12 of Soil Science and Technology and in this lecture we will be talking about the different data processing for digital soil mapping and then we will be talking about different types of continuous modeling for digital soil mapping.

So, as I told you that digital soil mapping requires some data processing because all soils are not measured universally at the same depth because as you can see here there are different horizon like O A B C these are the master horizons; however, in the legacy soil data which are compiled by different scientists you will see some data is available some data is not available. So, you see some time the data for O and A is available; however, the B and C data is not available. However you will sometime you will see O and C data is available; however, B and A data is not available.

So, there is a continuous there is a heterogeneity in terms of data availability and that is why we require some digital using some data processing. So, some soils are sampled per you know all soils are not measured universally at same depth, some soils are sampled per horizon or at regular dips, some soil studies examine only the topsoil while other sample the bedrock depth and then different soil attributes are measured at some location and depth, but not on others.

So, you can see huge amount of heterogeneity as far as the, you know, this legacy soil data is concerned that is a number of pre processing steps are needed to fulfill the requirements of DSM.

(Refer Slide Time: 01:56)

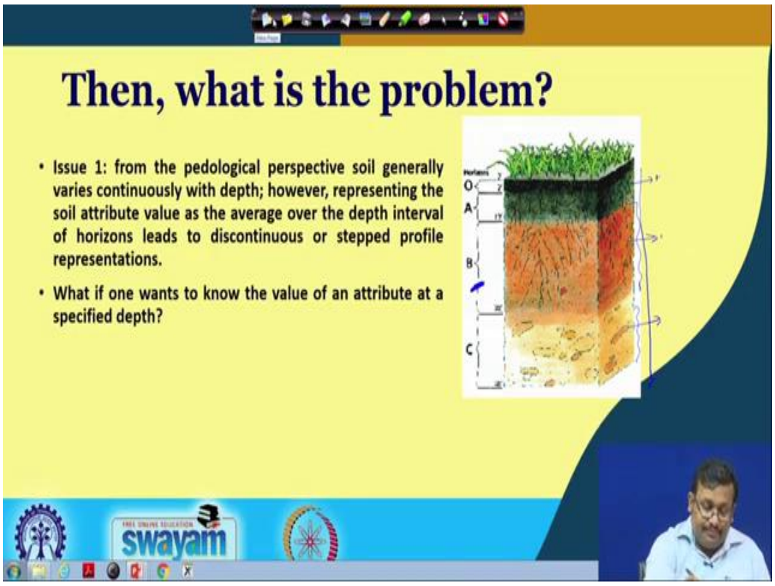


Soil depth function

- The traditional method of sampling soil involves dividing a soil profile into horizons.
- The number of horizons and the position of each are generally based on attributes easily observed in the field, such as morphological soil properties
- From each horizon, a bulk sample is taken and it is assumed to represent the average value for a soil attribute over the depth interval from which it is sampled.

So, what are these steps? With the first step is that soil depth function and the traditional method of sampling soil involves dividing a soil into profile into horizon as you can see O A B and C and the number of horizon and the position of each are generally based on attributes easily observed in the field such as morphological properties you have already discussed that. And from each horizon a bulk sample is taken and it is assumed to represent the average value for a soil attribute over the depth interval for which a sample. For example, if you are taking a bulk sample from the B horizon, then we generalize the property of that sample as the representative property of B horizon.

(Refer Slide Time: 02:41)



# Then, what is the problem?

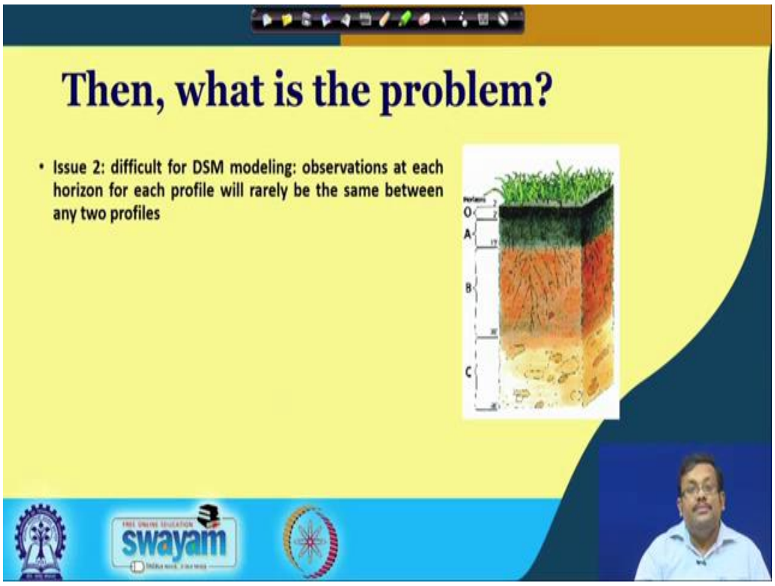
- Issue 1: from the pedological perspective soil generally varies continuously with depth; however, representing the soil attribute value as the average over the depth interval of horizons leads to discontinuous or stepped profile representations.
- What if one wants to know the value of an attribute at a specified depth?

However what is the problem in that case? The first issue is from the pedological perspective soil generally varies continuously with depth; however, representing to the soil attribute values as the average over the depth interval of horizon leads to discontinuous or stepped profile representation.

For example, we remember that, we know, we know that the soil property varies continuously from the top of the profile to bottom of the profile, but according to this bulk soil representation if we take one sample from here, one sample from here and one sample from here and based on the measurement of that particular samples, we are generalizing the whole depth that is the whole horizon.

So, that is producing some discontinuous or stepped profile representation so that is problematic, this so what if one wants to know the value of an attribute at a specified depth. So, if we want to know the value at this particular depth what is how it will be possible? It will not be possible if you are taking a bulk samples and generalizing the whole, you know, horizon. So, that is why we need some depth function and let us see what are the other issue?

(Refer Slide Time: 04:01)

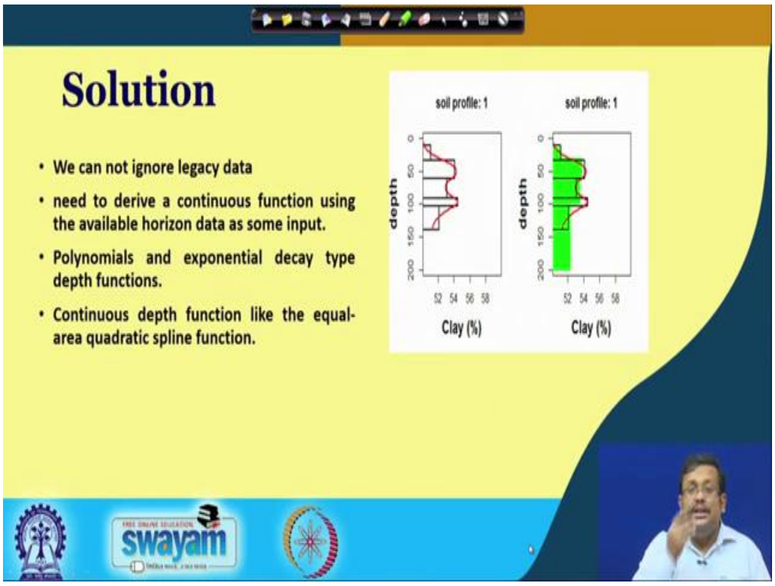


## Then, what is the problem?

- Issue 2: difficult for DSM modeling: observations at each horizon for each profile will rarely be the same between any two profiles

The second issue is difficult for DSM modeling because observation at each horizon for each profile will rarely be the same between any two profiles. So, observation at each horizon for each profile will be rarely be same between any 2 profiles.

(Refer Slide Time: 04:20)



## Solution

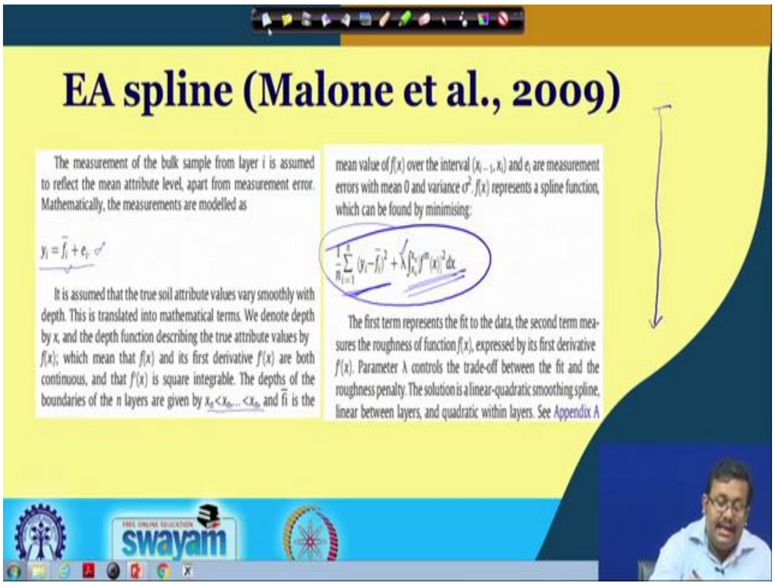
- We can not ignore legacy data
- need to derive a continuous function using the available horizon data as some input.
- Polynomials and exponential decay type depth functions.
- Continuous depth function like the equal-area quadratic spline function.

So, these are two issues for which we require some pre processing and the solution is we cannot, remember that we cannot ignore the legacy data because this is the major base data which we require for digital soil mapping. So, we need to derive a continuous function using the available horizon data as input for example, if in this case we are having the depth data from, you know, somewhere between 10 to 30 and then 30 to 60 and then up to 140 or something like this.

So, these are our available clay data and we need to fit a polynomial and exponential decay type of depth function, as you can see this is a polynomial or exponential type of depth function this red line and this is a continuous depth function also I am sorry this is a continuous depth function like equal area quadratic spline function. So, as you can see here these are the available soil data for different depths like, you know you know, for say for 10 to 20 and then 20 to 60 and so on so forth up to 140.

So, we are fitting a continuous depth function like equal area quadratic spline function here represented by this red line and using this continuous depth function we can predict the depth at a particular point.

(Refer Slide Time: 05:54)



# EA spline (Malone et al., 2009)

The measurement of the bulk sample from layer *i* is assumed to reflect the mean attribute level, apart from measurement error. Mathematically, the measurements are modelled as

$$y_i = \bar{f}_i + e_i$$

It is assumed that the true soil attribute values vary smoothly with depth. This is translated into mathematical terms. We denote depth by *x*, and the depth function describing the true attribute values by $f(x)$; which mean that $f(x)$ and its first derivative $f'(x)$ are both continuous, and that $f'(x)$ is square integrable. The depths of the boundaries of the *n* layers are given by $x_0 < x_1, \ldots < x_n$, and $\bar{f}_i$ is the mean value of $f(x)$ over the interval $(x_{i-1}, x_i)$ and $e_i$ are measurement errors with mean 0 and variance $\sigma^2$. $f(x)$ represents a spline function, which can be found by minimising:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{f}_i)^2 + \lambda\int_{x_0}^{x_n} f'(x)^2 dx$$

The first term represents the fit to the data, the second term measures the roughness of function $f(x)$, expressed by its first derivative $f'(x)$. Parameter $\lambda$ controls the trade-off between the fit and the roughness penalty. The solution is a linear-quadratic smoothing spline, linear between layers, and quadratic within layers. See Appendix A

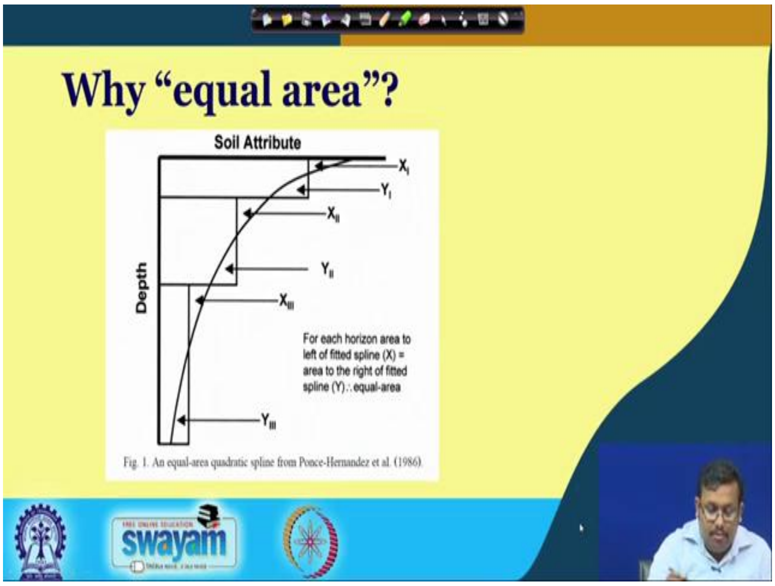
So, this is an example and let us see in details about the mathematical function; obviously, this equal area spline the formula for equal area spline was first we got from this Malone et al in 2009, where we assume that the measurement of bulk sample from layer i is assumed to reflect mean attribute level apart from measurement error. So, mathematically the measurement can be modeled as y i equal to f i bar plus e i. So, it is assumed that the true soil attributes really vary across very smoothly with the depths.

So, the soil attribute varies smoothly from the top of the profile to the bottom of the profile and this is translated into mathematical terms. So, we denote depth function by x and the depth function describe the two attribute values by f x, a function of x. Obviously, so which means that f x and its first derivative are both continuous and the first derivative of x is a square integrable. So, the depth of the boundaries of n layers are given by for example, x n x n and x n.

So, the f i bar here this f i bar is the mean value of this f x over the interval of x i minus 1 to x i. So, it is for one layer it is the mean value over the interval and e i as the measurement error; obviously, this is a measurement error and with the vary with the mean 0 and variance sigma square. So, f x represent a spline function which can be found by minimizing this term and this term basically represent the first term basically represents the fit to the data and the second term measure the roughness function or penalty. So, roughness function or f x, so expressed by its first derivative.

So, the parameter lambda here basically this is a lambda which is called the tuning parameter, it is basically tradeoff between fit and the roughness penalty and the solution is a linear quadratic smoothing spline. So, let us see, so this is basically a mathematical function of these linear quadratic smoothing spline and let us see how it will work

(Refer Slide Time: 08:11)



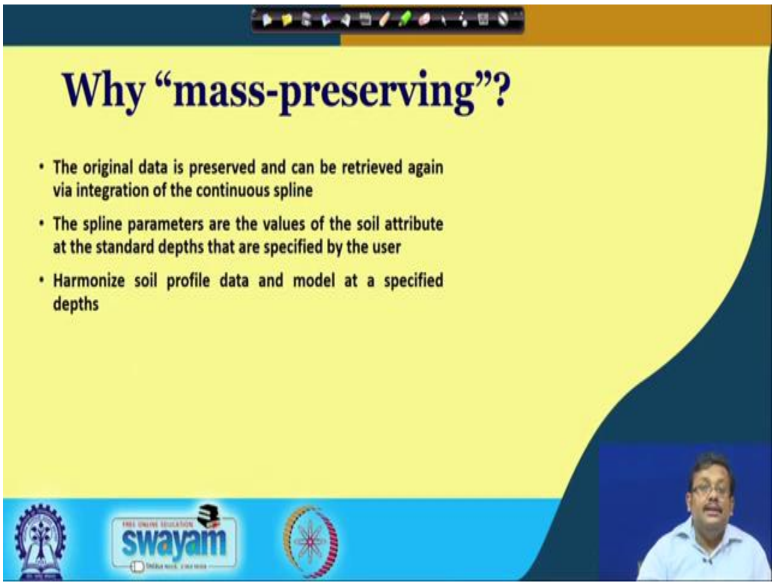
## Why “equal area”?

For each horizon area to left of fitted spline (X) = area to the right of fitted spline (Y) ∴ equal-area

Fig. 1. An equal-area quadratic spline from Ponce-Hernandez et al. (1986).

So, this is basically the equal area spline or linear quadratic spline. So, this is a quadratic spline which you use to harmonize the soil data across our desired depth, remember that this is called equal area spline because for each horizon area to the left fitted spline you know; obviously, equal to the area to the right fitted spline. So, that is why it is called equality, so you can see this is spline, this area will be equal to this area, this area will be equal to this area, this will be equal to this area. So, that is why it is called the equal area spline.

(Refer Slide Time: 08:50)

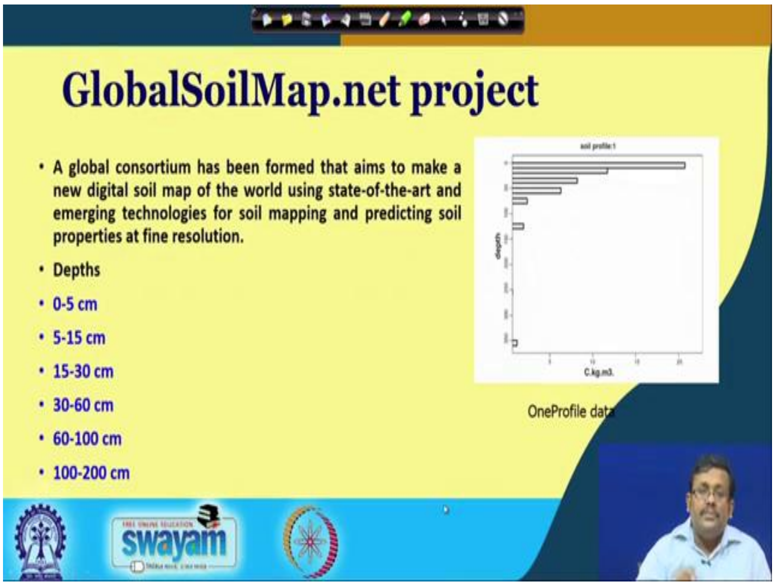


## Why "mass-preserving"?

- The original data is preserved and can be retrieved again via integration of the continuous spline
- The spline parameters are the values of the soil attribute at the standard depths that are specified by the user
- Harmonize soil profile data and model at a specified depths

Another name of this spline is called mass preserving spline, because the original data is preserved and can be retrieved again via integration of the continuous spline. So, the spline parameter are the value of soil attribute at the standard depth that are specified by the users and remember that using the spline profile, using this mass preserving equal area spline you can harmonize soil profile data and model at a specified depth.

(Refer Slide Time: 09:17)



# GlobalSoilMap.net project

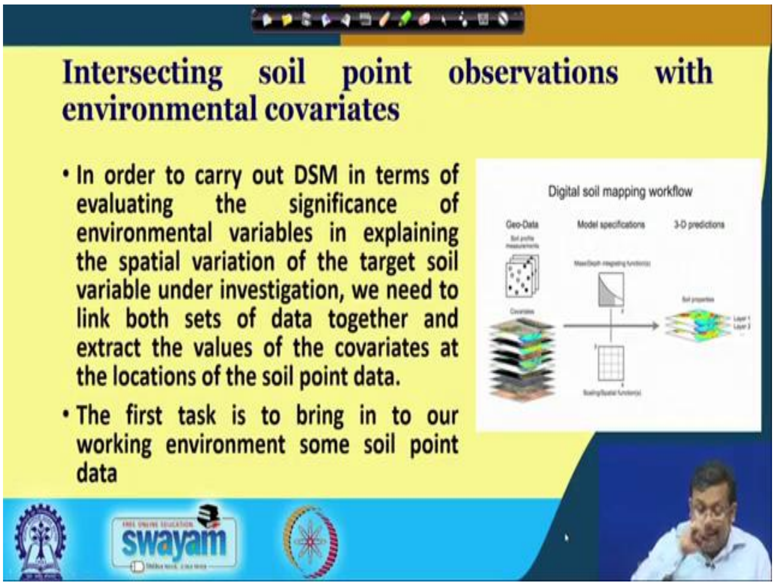
- A global consortium has been formed that aims to make a new digital soil map of the world using state-of-the-art and emerging technologies for soil mapping and predicting soil properties at fine resolution.
- Depths
- 0-5 cm
- 5-15 cm
- 15-30 cm
- 30-60 cm
- 60-100 cm
- 100-200 cm

OneProfile data

So, as I have told you the global soil map dot net project is a global consortium that, you know, aim to make new digital soil map of the world using this state of the art and averaging and this emerging technologies for soil mapping and predicting soil properties in fine resolution, they have prescribed some 6 standardized depth. These 6 standardized depths are these 0 to 5 centimeter, 5 to 15 centimeter, 15 to 30 centimeter, 30 to 60, centimeter 60 to 100, centimeter and 100 to 200 centimeter.

So, if our profile data looks like this the original legacy data looks like this it is possible by using this equal area spline to get the continuous value or, you know, of this particular parameter up to our desired depth by fitting this equal area spline.

(Refer Slide Time: 10:15)



## Intersecting soil point observations with environmental covariates

- In order to carry out DSM in terms of evaluating the significance of environmental variables in explaining the spatial variation of the target soil variable under investigation, we need to link both sets of data together and extract the values of the covariates at the locations of the soil point data.
- The first task is to bring in to our working environment some soil point data

Digital soil mapping workflow
Geo-Data
Soil profile measurements
Covariates
Model specifications
Mass/Depth integrating function(s)
Scaling/Spatial function(s)
3-D predictions
Soil properties
Layer 1
Layer 2
swayam

So, guys another important aspect of data processing is intersecting soil point observation with environmental covariates. So, in order to carry out DSM in terms of evaluating the significance of environmental variables in explaining the special variation target soil, you know, the target soil variable under investigation, we need to link both sets of data together and extract the values of the covariance at the location of the point data.

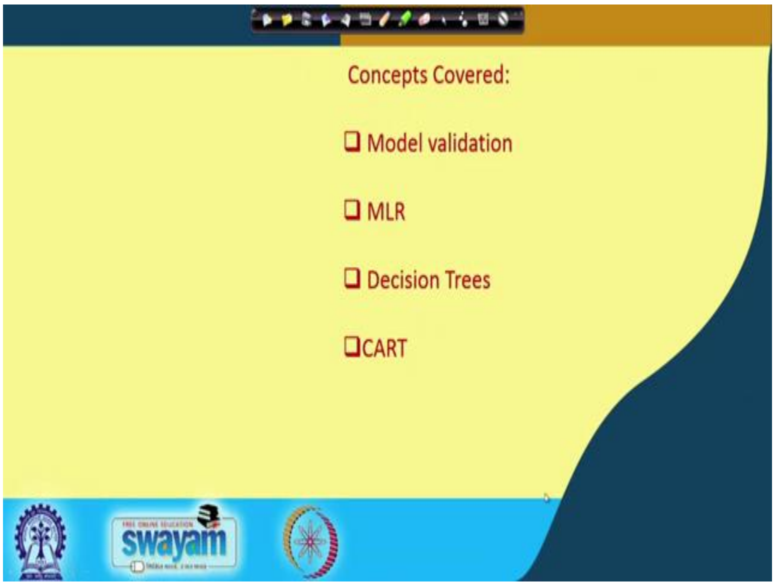
Obviously, as you can see as we have discussed already the workflow of in the work flow of DSM, the first starting point is our legacy soil data and remember all this legacy soil point data is, you know, is associated with some x and y coordinate data because we are using our GPS receiver to measure their coordinates. So, once we measure their coordinates the next step we can see we are stacking all the covariance data and it is required that the covariate values from the grids once you stacked all these covariates you know.

The next step we have to extract all the covariates values which is available for this particular soil point data and all these will be related by this x and y coordinates. So, again once we get the legacy soil data and their coordinates the next step will be to extract the soil covariate information for those particular points. So, we do not need the soil covariate information at for all the area, we only need the soil covariate information for those particular points for which the legacy soil data is available. So, we call it intersecting and this is another major pre processing step before we go for digital soil mapping.

So, in a nutshell you know; obviously, we will start with this x y data, we will start with this geo data, legacy soil data and then we will fit a mass depth integrating function or equal area spline to get the depth wise values of that variable up to our desired depths. There are 6 desire depths as proposed by global soil map dot net project and once we harmonize all the profile data up to that standard depths the next step is to stack the covariates and extract the covariate information for those particular points for which we have the soil point data.

Now once we get this thing then we will use this in the spatial prediction model using some mathematical function to produce the 3D maps of soil for that particular property. So, this is in a nutshell about the work flow in the digital soil mapping and so we have covered the basic overview of digital soil mapping, the next and very very important topic of in digital soil mapping is modeling and mapping of continuous variables.

(Refer Slide Time: 13:29)



Concepts Covered:

❑ Model validation

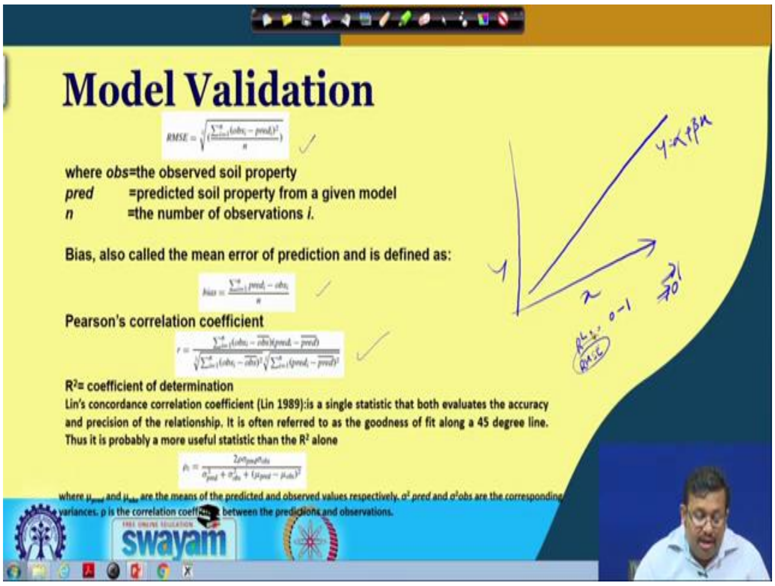
❑ MLR

❑ Decision Trees

❑CART

And in this we will be discussing about the model validation and then what is multiple linear regression, what is decision trees and what is cart and also we will be discussing about random forests in brief also.

(Refer Slide Time: 13:45)



# Model Validation

$$RMSE = \sqrt{\left(\frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{n}\right)}$$

where *obs*=the observed soil property
*pred* =predicted soil property from a given model
*n* =the number of observations *i*.

Bias, also called the mean error of prediction and is defined as:

$$bias = \frac{\sum_{i=1}^{n} pred_i - obs_i}{n}$$

Pearson's correlation coefficient

$$r = \frac{\sum_{i=1}^{n}(obs_i - \overline{obs})(pred_i - \overline{pred})}{\sqrt{\sum_{i=1}^{n}(obs_i - \overline{obs})^2}\sqrt{\sum_{i=1}^{n}(pred_i - \overline{pred})^2}}$$

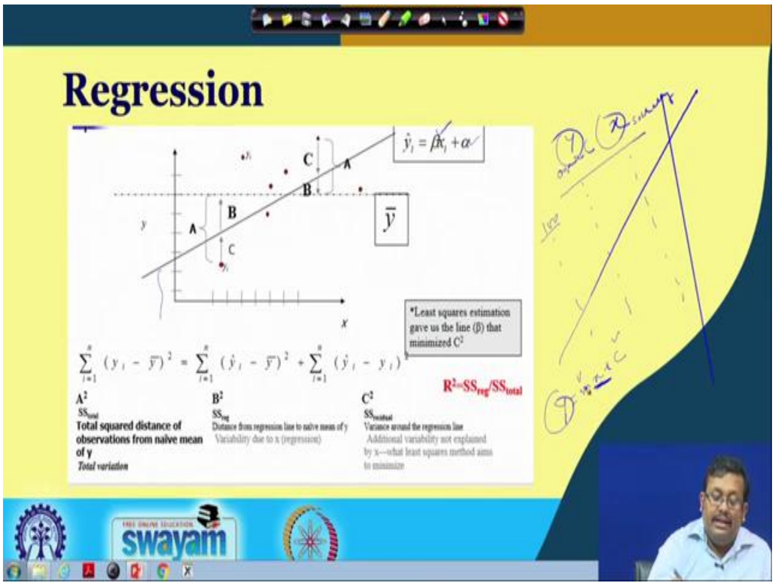
$R^2$= coefficient of determination

Lin's concordance correlation coefficient (Lin 1989):is a single statistic that both evaluates the accuracy and precision of the relationship. It is often referred to as the goodness of fit along a 45 degree line. Thus it is probably a more useful statistic than the $R^2$ alone

$$\rho_c = \frac{2\rho\sigma_{pred}\sigma_{obs}}{\sigma_{pred}^2 + \sigma_{obs}^2 + (\mu_{pred} - \mu_{obs})^2}$$

where $\mu_{pred}$ and $\mu_{obs}$ are the means of the predicted and observed values respectively. $\sigma^2$ *pred* and $\sigma^2$*obs* are the corresponding variances. ρ is the correlation coefficient between the predictions and observations.

(Refer Slide Time: 13:51)



# Regression

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

*Least squares estimation gave us the line (β) that minimized $C^2$

$R^2$=$SS_{reg}/SS_{total}$

$A^2$
$SS_{total}$
Total squared distance of observations from naive mean of y
*Total variation*

$B^2$
$SS_{reg}$
Distance from regression line to naive mean of y
Variability due to x (regression)

$C^2$
$SS_{residual}$
Variance around the regression line
Additional variability not explained by x—what least squares method aims to minimize

So, guys if we talk about the model validation let us see what is regression? Now in the regression basically we are interested to predict certain variable using another random variable. So, for example, if we have two variable say x and y or in other words say y is our as far as the soil data is concerned for example, soil organic carbon and x is our for example, let us say it is soil clay.

So, if we want to have for example, we have 100 soil samples and for 100 soil samples we have 100 records of organic carbon and 100 records of corresponding clay content.

So, if we want to predict this y or target variable using our independent variable let us see this y is our target variable and x is the independent variable, if we want to predict or if we want to make a relationship where we will predict the y in terms of x, that is called regression.

So, simple linear regression basically takes the form mx plus c or here y is our target, x is the predictor, m is the slope and c is the offset. Now you can see here, this is the regression in a nutshell; obviously, in the x axis these are the predict these are the predictors, in the y axis this is the target variable and ultimately we try to fit a linear regression; we try to fit a linear, you know, we try to fit a linear relationship as you can see here this is the straight line and it has a representation of b x i plus alpha; obviously, alpha is the offset and beta is beta is that here the slope.

So, this is called regression this is the simplest form of regression and in case of multiple linear regression when there are more than one predictor variable then we call it multiple linear regression. So, guys this is the overview of regression and the why we are discussing this let us see. Now, if we go back so once we create a model for predicting a particular soil property, why we create a model?

We create a model so that it can generalize or it can predict any particular property using some unknown samples. So, that is the major use of creating a model. So, once we create a particular linear model or any other non linear model using some soil data or soil covariate data, the next step is to, you know, ascertain their accuracy, model accuracy. Now the model accuracy can be calculated by model validation statistics.

The model validation is basically the appraisal of the usefulness of the model or in other words whether the created model or calibration model we call it. The calibration model can be used to generalize or to predict any unknown sample that can be, you know, that can be predicted via validation. So, there are several important points; obviously, like root mean squared error; root mean squared error can be, you know; obviously, you can say this is a formula of root mean squared error.

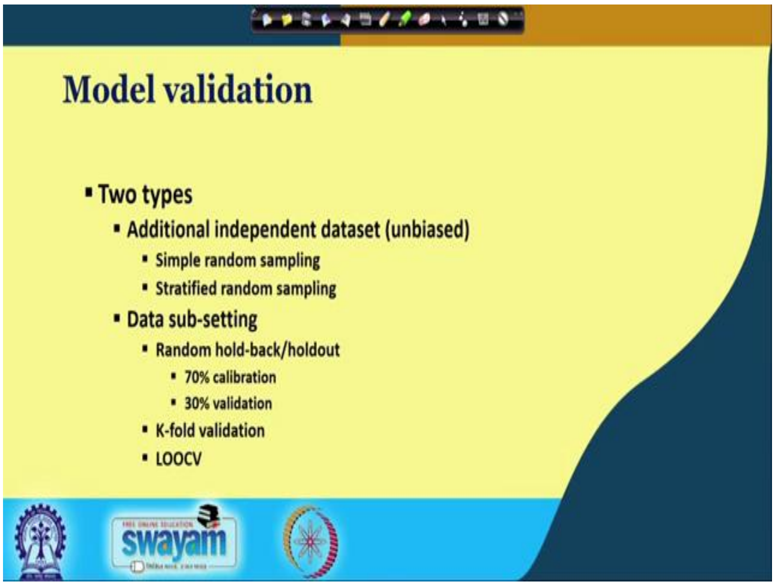
And this is one of the major important index of soil index of model validation we always want the root mean squared error to be minimized because the less the root mean squared error the more accurate the model and also bias which is called the mean error of prediction and it is defined by this formula and then Pearson correlation coefficient;

Pearson correlation coefficient basically shows the strength of linear relationship between two variables. So, this is another index of model validation.

And then Lin's concordance correlation coefficient which is a single statistic that both evaluates the accuracy and precision of the relationship, remember that it is often referred to as the goodness of the fit along the 45 degree line thus it is probably a more useful statistics than the R square alone. So, if we consider this x and y and we create a model like y equal to alpha plus beta x so, this will be a linear regression and in this linear regression; obviously, remember that these are some important indices like R square RMSE.

Remember that R square measures the strength of the regression relationship and R square varies from 0 to 1 as R square tends towards 1, the model is more accurate and as it goes towards 0 the model is there is no relationship and RMSE is also Root Mean Squared Error, as you can see this is root mean squared error. So; obviously, this is an RMSE and RMSE measures that what is the strength of this linear relationship. So, these are some important indices of model validation and they have some spacing values which generally tells us whether that model is good or bad.

(Refer Slide Time: 20:11)



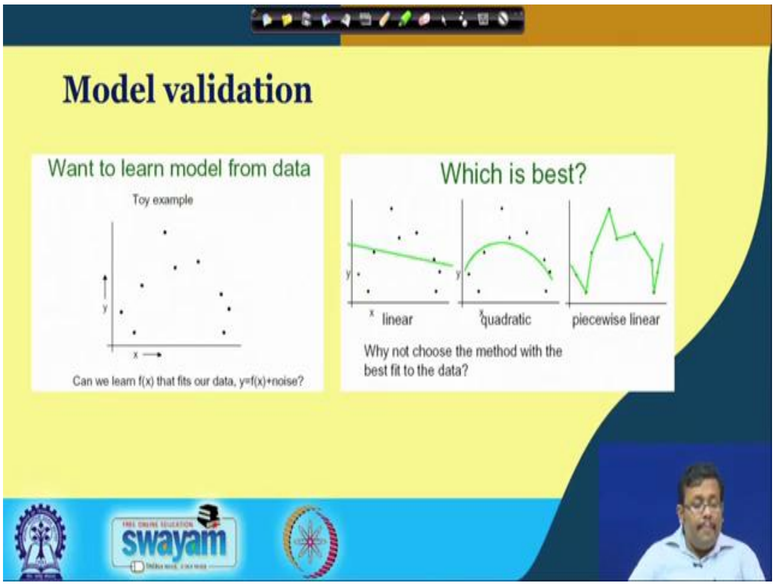
## Model validation

- Two types
  - Additional independent dataset (unbiased)
    - Simple random sampling
    - Stratified random sampling
  - Data sub-setting
    - Random hold-back/holdout
      - 70% calibration
      - 30% validation
    - K-fold validation
    - LOOCV

So, let us see what are the different types of model validation, model validation is of two types; one is, you know, basically the first way of doing the model validation is using some additional independent data set which is basically unbiased and we can draw that by simple random sampling and or stratified random sampling this is the best way. I mean you create one model and then you draw a random sample and independent data set and which is unbiased and you use this, we know, independent data set to judge the accuracy of your created calibration model.

So, this is the first, you know, way and this is the most important way. Another is the data sub setting and we will be talking about in details. So, the data sub setting basically are of three types one is random hold back or hold out and then K fold validation and leave one out cross validation or LOOCV. Now in the random hold back or hold out basically we divide the data into 70 percent and we create the model using this 70 percent data we call it calibration data set, the rest 30 percent of the data is used for validating the model ok.

(Refer Slide Time: 21:38)



Model validation

Want to learn model from data

Can we learn f(x) that fits our data, y=f(x)+noise?

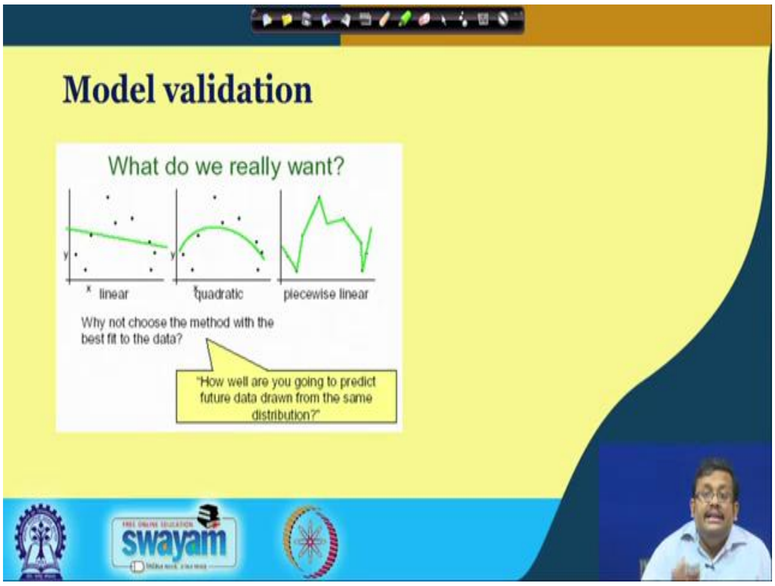
Which is best?

Why not choose the method with the best fit to the data?

So, let us see them one by one. So, model validation basically, ask whether, you know, can we fit our data into this model. So, for example, here there is an particular example it is a independent variable and this is a dependent variable, the question is can we learn or create a relationship that fits our data y equal to f x plus noise or f x plus error.

So, you can see there are three different relationship one is the linear relationship another is quadratic relationship another here you can see piecewise linear relationship. So, let us see them one by one and our question is which one is the best.

(Refer Slide Time: 22:24)



## Model validation

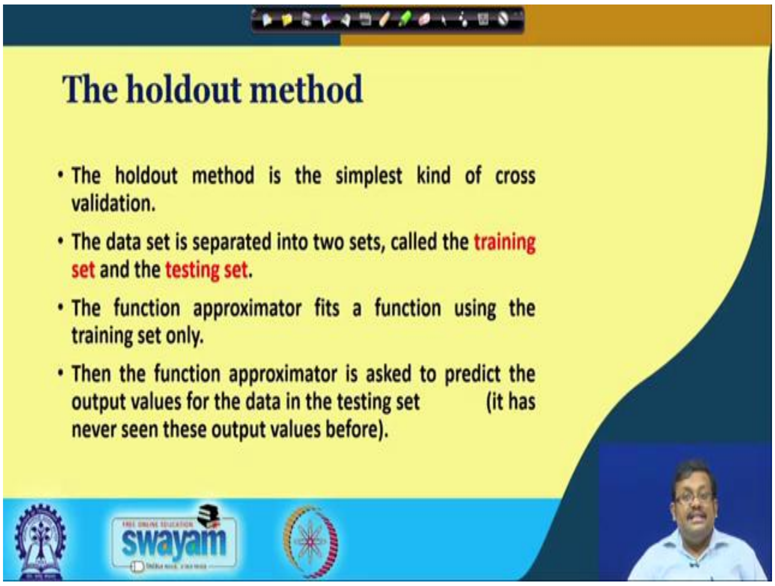
What do we really want?

Why not choose the method with the best fit to the data?

"How well are you going to predict future data drawn from the same distribution?"

So, you can see what do we really want? So we why not choose the method with the best fit of the data. So, our intention is to select the best model which can generalize this behavior either linear or quadratic or piecewise linear.

(Refer Slide Time: 22:42)

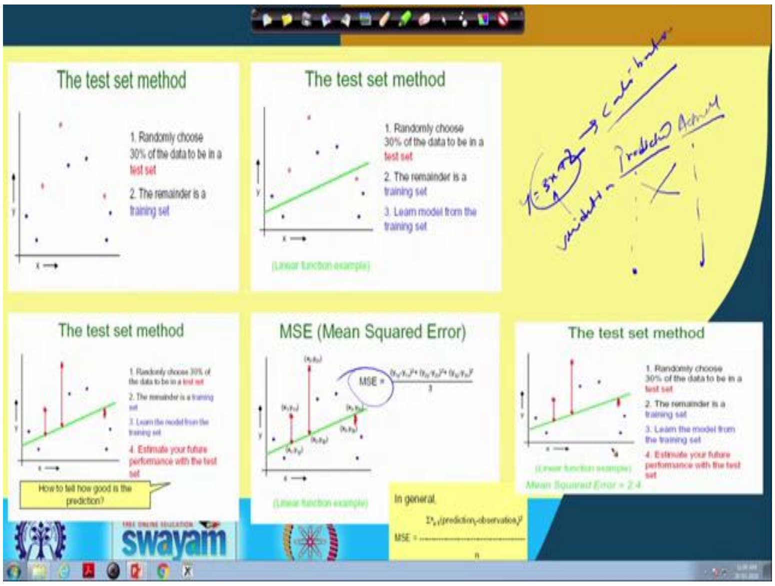


## The holdout method

- The holdout method is the simplest kind of cross validation.
- The data set is separated into two sets, called the training set and the testing set.
- The function approximator fits a function using the training set only.
- Then the function approximator is asked to predict the output values for the data in the testing set (it has never seen these output values before).

So, first let us see what is the holdout method? So the holdout method is a simplest kind of cross validation. The data set is separated into two set randomly sometimes called the training set and testing set. Training set, another name is calibration set and testing set the other name of testing set is validation set and the function approximator fits a function using the training set only and then the function approximator is asked to predict the output values for the data in the testing set.

(Refer Slide Time: 23:13)



The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set

The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Learn model from the training set

(Linear function example)

The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Learn the model from the training set
4. Estimate your future performance with the test set

How to tell how good is the prediction?

MSE (Mean Squared Error)

$$MSE = \frac{(y_1-y_1')^2+(y_2-y_2')^2+(y_3-y_3')^2}{3}$$

(Linear function example)

In general

$$MSE = \frac{\sum_{i=1}^{n}(prediction_i - observation_i)^2}{n}$$

The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Learn the model from the training set
4. Estimate your future performance with the test set

(Linear function example)

Mean Squared Error = 2.4

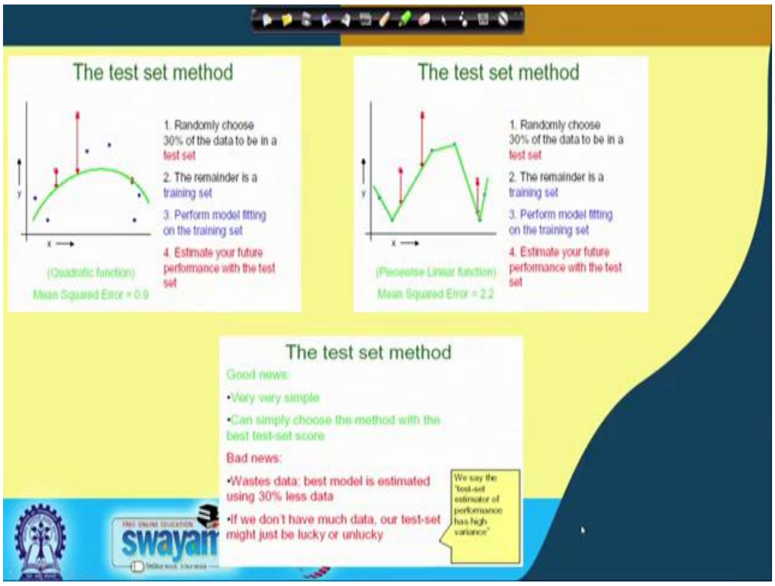
So, let us see this in the step by step. So, let us see this is test set method, so randomly first in the first we randomly choose 30 percent of the data to be in the data test set and the remainder is in the training set. So, as you can see these blue dots are basically the training set and these red dots are the test set and in the next step is learning the model from the training set. So, once we created this division between the learning or training set and the test set, let us use this training set to create a first model y equal to alpha plus beta x.

So, this is our next step, the next step says estimate your future performance with the test set. So, once we create a model now let us use our test set to use in this particular relationship to predict and then let us estimate our future performance with the test set. For example, if we have created a particular model for example, y equal to say for 3 x plus 2.

So, here we will then input. So, this is our calibration model, now once you create this calibration model let us input our validation data points or test set into this relationship and get some values these will be predicted values. So, once we get the predicted values we also know their original values, so let us make a relationship let us see how they are accurate or not or let us make a relationship between the predicted and actual so this relationship is called the validation relationship.

So, again once you create the trading model or calibration model then we will estimate our future performance with the test set and then we will calculate the mean square error by using this particular formula and then mean square error; obviously, and then we will estimate our future performance. So, in this case our mean square a particular example mean square error is 2.4. So, this is an example of the random holdout method.

(Refer Slide Time: 25:48)


The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform model fitting on the training set
4. Estimate your future performance with the test set

(Quadratic function)
Mean Squared Error = 0.9

The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform model fitting on the training set
4. Estimate your future performance with the test set

(Piecewise Linear function)
Mean Squared Error = 2.2

The test set method

Good news:

•Very very simple

•Can simply choose the method with the best test-set score

Bad news:

•Wastes data: best model is estimated using 30% less data

•If we don't have much data, our test-set might just be lucky or unlucky

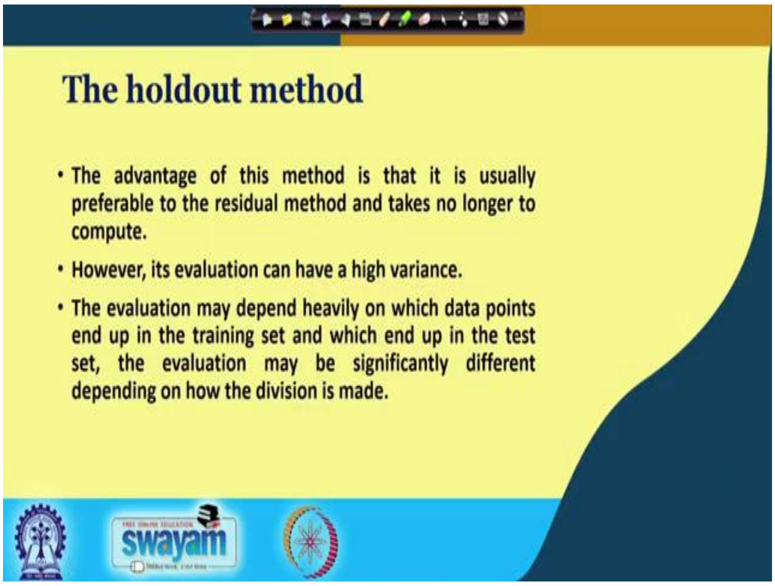
We say the "test-set estimator of performance has high variance"

Second is test set method so randomly choose again and this for quadratic function the similarly just like we did for linear regression for this for quadratic regression, this is for piece wise regression. And ultimately we will get an estimate of mean squared error you can see in the first linear regression we are getting a mean squared error of 2.4 and in the quadratic method we are getting mean squared error of 0.9 and in the test set in the piecewise method we are getting a mean squared error of 2.2.

So, according to, you know, by come by seeing this result we can say that this quadratic function shows the best fit of in the available data. Now; obviously, there are some positive things and negative things in this test set method, the positive things are they are very very simple and they can simply choose the method with the best test score and the problem is there is wastage of data; that means, best model is estimated using only using 30 percent less data because we are using only 70 percent of the data.

And if we do not have much data, our test set might be just lucky or unlucky because it may be possible that that test set samples are completely different than that of our calibration set and as a result of that there will be complete mismatch and the model validation will produce very very poor results.

(Refer Slide Time: 27:31)

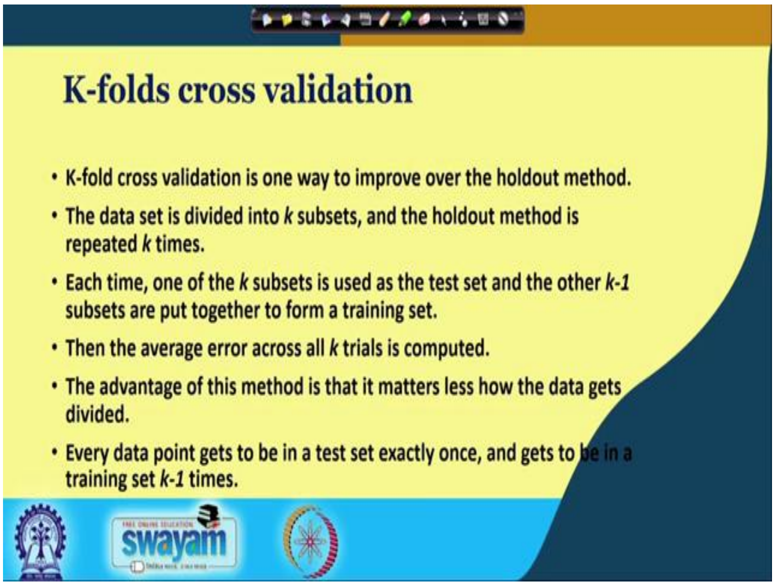


The holdout method

- The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute.
- However, its evaluation can have a high variance.
- The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, the evaluation may be significantly different depending on how the division is made.

So, advantage of this method is that it is usually I am sorry it is the previous method holdout method we have not gone to our next method. So, the advantage of this method is that it is usually preferable to the residual method and takes no longer to compute; however, it's evaluation can have a high variance and the evaluation may depend heavily on which data points end up in the training data set and which end up in the test set and the evaluation may be significantly different depending on how division is made so it is very very important.

(Refer Slide Time: 28:02)

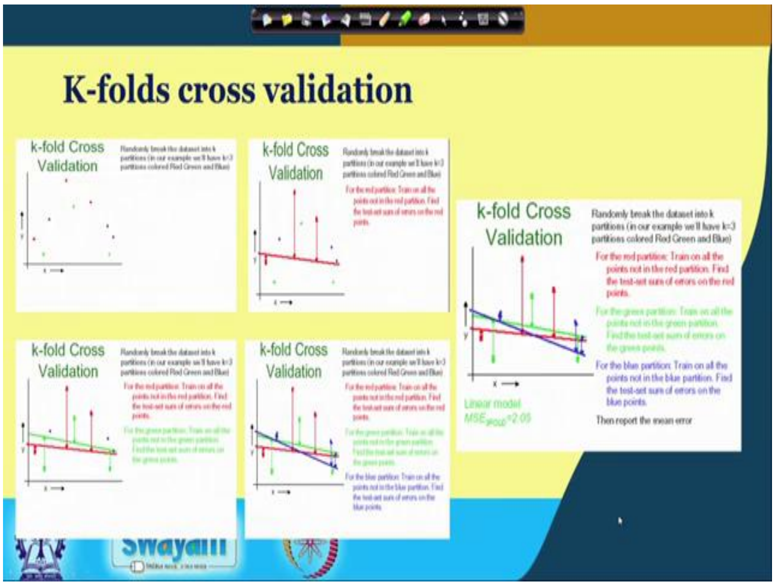


## K-folds cross validation

- K-fold cross validation is one way to improve over the holdout method.
- The data set is divided into *k* subsets, and the holdout method is repeated *k* times.
- Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set.
- Then the average error across all *k* trials is computed.
- The advantage of this method is that it matters less how the data gets divided.
- Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times.

The second method is called the K fold cross validation; now K fold cross validation is one way to improve over the hold out method. And the data set here is divided into k subsets and the holdout method is repeated k times, in each time one of the k subsets is used as a test set and the other k minus 1 subsets are put together to form a training set, then the average error across all the k trials is computed. The advantage of this method is that it matters less how the data gets divided and every data gets to be in the tests set exactly once and gets to be in a training set k minus 1 times.

(Refer Slide Time: 28:44)



## K-folds cross validation

k-fold Cross Validation

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

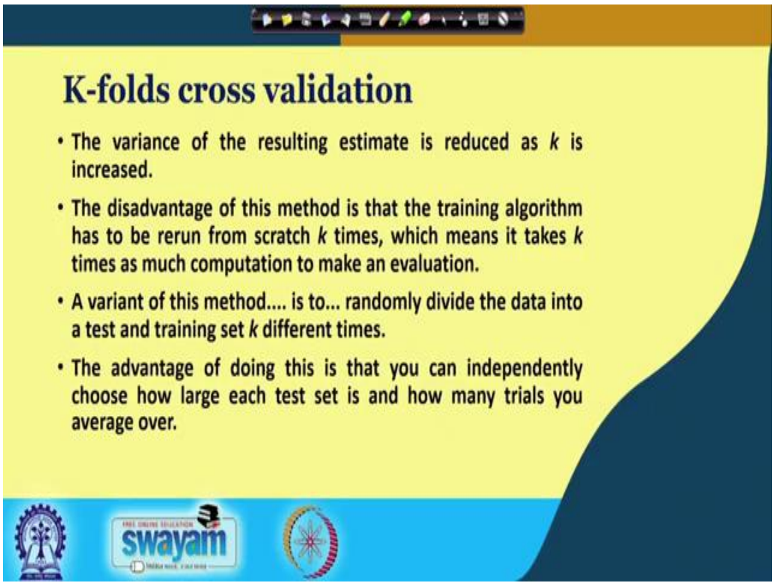
Then report the mean error

So, as you can see here this is an example of K fold cross validation and the K fold cross validation let us divide for example, this is our total data set let us randomly break the data set into k partition in our case let us do three partition and we denote them into red, green and blue. And in the second step for the red partition let us use the; for the red partition let us train all the points not in the red partition find the test sets comes square error on the red points.

So, basically what we will do? Will use all the red, all the green points and the blue points as our training points we will fit a training model and then we will produce then we will see the error from the by using this red test cases. And in the next iteration we will do the same using the green samples and the last step we will do the same using the blue samples.

So, you can see in each of the step we are holding one particular set. In the first time we are holding the red set, the second time we are holding the green set and the third time we are holding the blue set as test set and we are using the rest of the two for creating a calibration model and then we are producing, we are estimating the error. So, once you complete all these three iteration we will have an average and then this will be our mean and you know this will give us the mean validation performance.

(Refer Slide Time: 30:31)

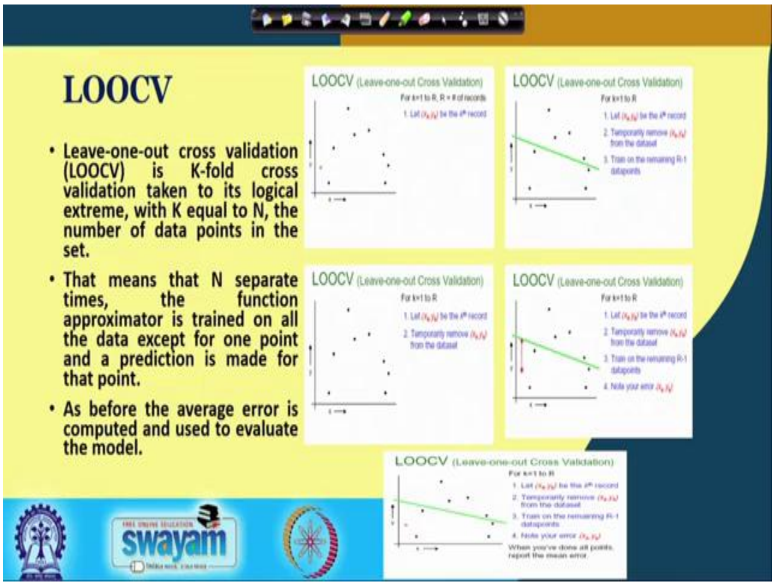


## K-folds cross validation

- The variance of the resulting estimate is reduced as *k* is increased.
- The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation.
- A variant of this method.... is to... randomly divide the data into a test and training set *k* different times.
- The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

So, this is about the K fold cross validation and the K fold cross validation says the variance of the resulting estimator reduces k is increased and the disadvantage of this method is that training algorithm has to be rerun from scratch to k times which means it takes k times as much as computation to make an evaluation. A variant of this method is to randomly divide the data into a test set and a training set k different times and the advantage of doing this is that you can independently choose over large, you know, large each test set and it is how many trials we average over.

(Refer Slide Time: 31:02)



## LOOCV

- Leave-one-out cross validation (LOOCV) is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set.
- That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point.
- As before the average error is computed and used to evaluate the model.

LOOCV (Leave-one-out Cross Validation)

For k=1 to R, R = # of records

1. Let $(x_k,y_k)$ be the $k^{th}$ record

LOOCV (Leave-one-out Cross Validation)

For k=1 to R

1. Let $(x_k,y_k)$ be the $k^{th}$ record
2. Temporarily remove $(x_k,y_k)$ from the dataset
3. Train on the remaining R-1 datapoints

LOOCV (Leave-one-out Cross Validation)

For k=1 to R

1. Let $(x_k,y_k)$ be the $k^{th}$ record
2. Temporarily remove $(x_k,y_k)$ from the dataset

LOOCV (Leave-one-out Cross Validation)

For k=1 to R

1. Let $(x_k,y_k)$ be the $k^{th}$ record
2. Temporarily remove $(x_k,y_k)$ from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error $(x_k,y_k)$

LOOCV (Leave-one-out Cross Validation)

For k=1 to R

1. Let $(x_k,y_k)$ be the $k^{th}$ record
2. Temporarily remove $(x_k,y_k)$ from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error $(x_k,y_k)$

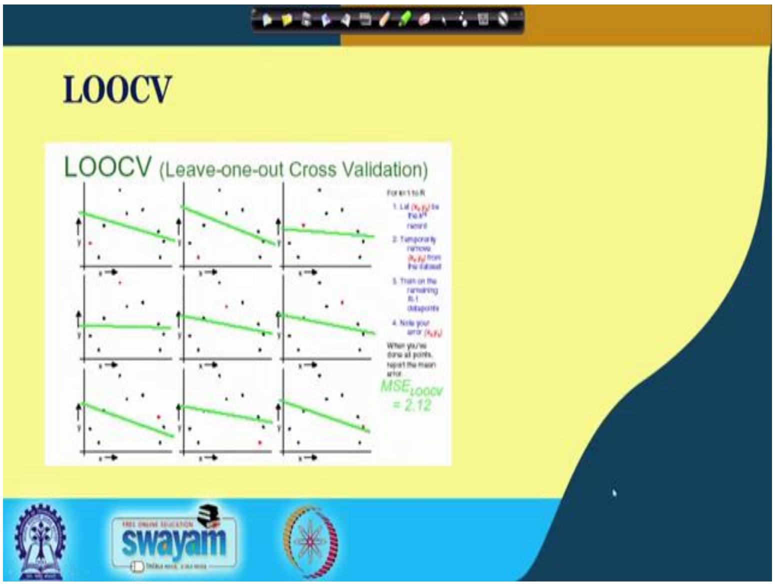
When you've done all points, report the mean error.

Then finally, one is leave one out cross validation; the leave one out cross validation is another form of K fold cross validation, you know, where K equal to ‘n’ and basically here what do we do for from the whole data set we select only one sample as a test sample in each iteration and then for example, there are K if there is an N sample we will select at each iteration we will select one sample as a test sample and then we will use the rest N minus 1 sample as training sample.

And we will use this N minus 1 sample to create a calibration model and then we will estimate the error using this one sample which we used for testing and then we will repeat this for all the samples and ultimately we will predict the mean error from, you know, by averaging all the errors from this each iteration.

So, you can see this leave one out cross validation we are leaving one out each time and then we are creating a model and then we are estimating the error and this is an extension of K fold cross validation. So, that is why it is called leave one out cross validation, sometimes we will call it full cross validation also.

(Refer Slide Time: 32:39)



## LOOCV

LOOCV (Leave-one-out Cross Validation)

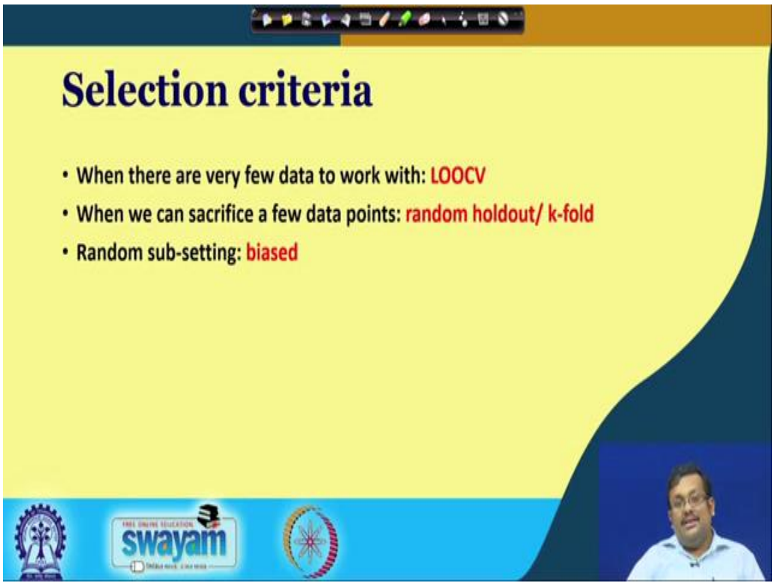
For k=1 to R

1. Let $(x_k,y_k)$ be the $k^{th}$ record
2. Temporarily remove $(x_k,y_k)$ from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error $(x_k,y_k)$

When you've done all points, report the mean error.

$MSE_{LOOCV} = 2.12$

(Refer Slide Time: 32:33)

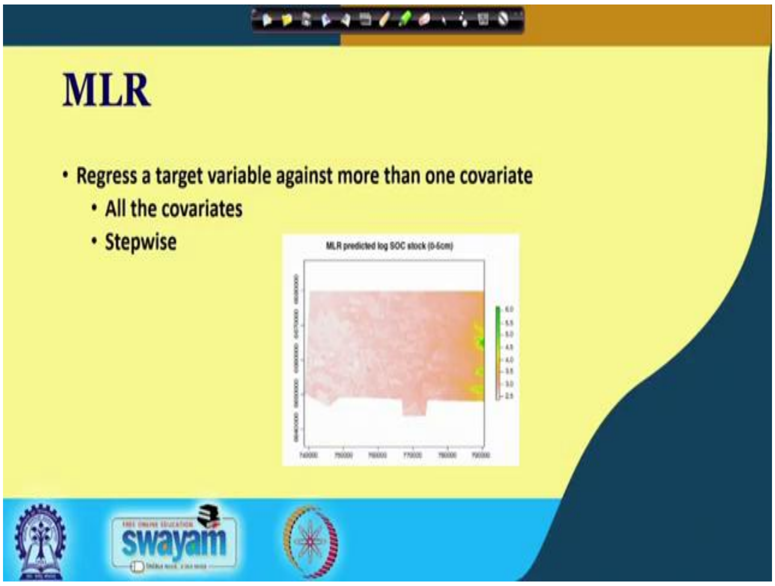


## Selection criteria

- When there are very few data to work with: LOOCV
- When we can sacrifice a few data points: random holdout/ k-fold
- Random sub-setting: biased

So, this is an overview of leave one out cross validation you can see and the selection criteria in which case we will use which one. So, when there are a very few data to work with; obviously, you will go with the leave one out cross validation. When we can sacrifice a few data points; obviously, we will go with the random holdout or K fold. However, in case of random sub setting, there is a chance of biasness because we do not know in case of random subset which one sample will go to the test set and which sample will go to the training set there could be complete mismatch. So, this is the advantages and disadvantages of this method.

(Refer Slide Time: 33:10)

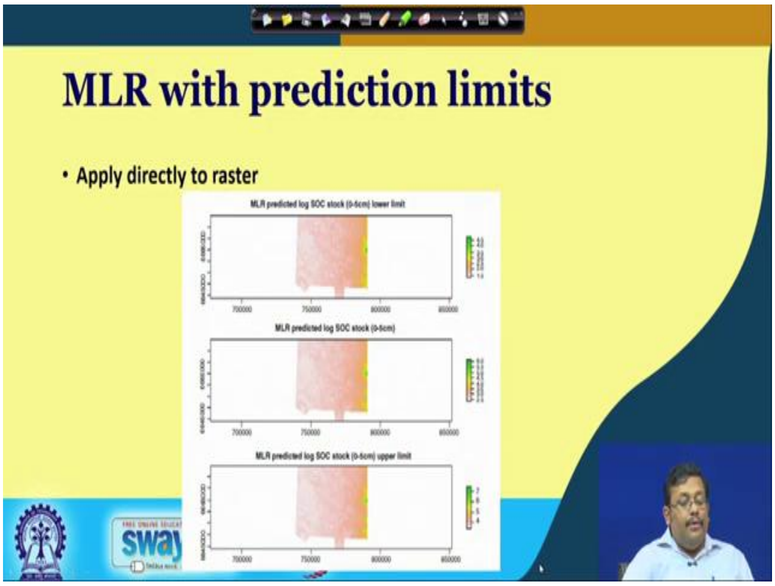


# MLR

- Regress a target variable against more than one covariate
  - All the covariates
  - Stepwise

And you can see here this is a MLR Multiple Linear Regression and this is a MLR example of in digital soil mapping, here we regress a target variable against more than one covariate and we use all the covariates and there are two ways we can do this multiple linear regression we can use either all the covariates in as the predictor or we can use stepwise linear regression as you can see this is a multiple linear regression predicted soil organic carbon stock in 0 to 5 centimeter. So, this map is produced by multiple linear regression.

(Refer Slide Time: 33:47)



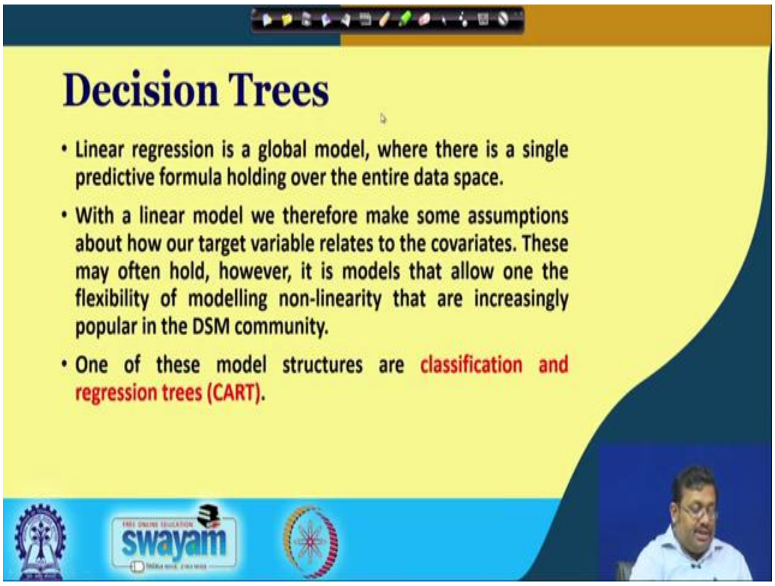
# MLR with prediction limits

- Apply directly to raster

MLR predicted log SOC stock (0-5cm) lower limit
MLR predicted log SOC stock (0-5cm)
MLR predicted log SOC stock (0-5cm) upper limit

And also this multiple linear regression can produce, not only it will give the prediction for this 0 to 5 centimeter. You can see it also produces the lower limit of prediction.

(Refer Slide Time: 34:00)



## Decision Trees

- Linear regression is a global model, where there is a single predictive formula holding over the entire data space.
- With a linear model we therefore make some assumptions about how our target variable relates to the covariates. These may often hold, however, it is models that allow one the flexibility of modelling non-linearity that are increasingly popular in the DSM community.
- One of these model structures are classification and regression trees (CART).

And not only it gives the multiple, you know, the prediction for the soil organic carbon stock for 0 to 5 centimeter, it also gives you the prediction interval that is lower limit of prediction as well as upper limit of prediction. So, you get a sense of the uncertainty in this in this DSM product.

So, guys let us stop here and we will start from here in our next lecture and we will try to finish this continuous model and categorical model in our next lecture and I hope that you have learned something new and in the next lecture we will talk about different types of non linear models also.

Thank you very much.