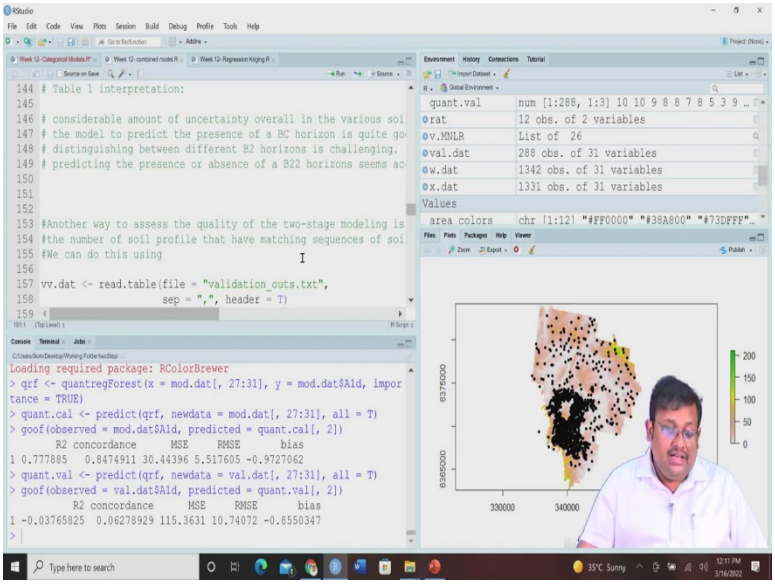
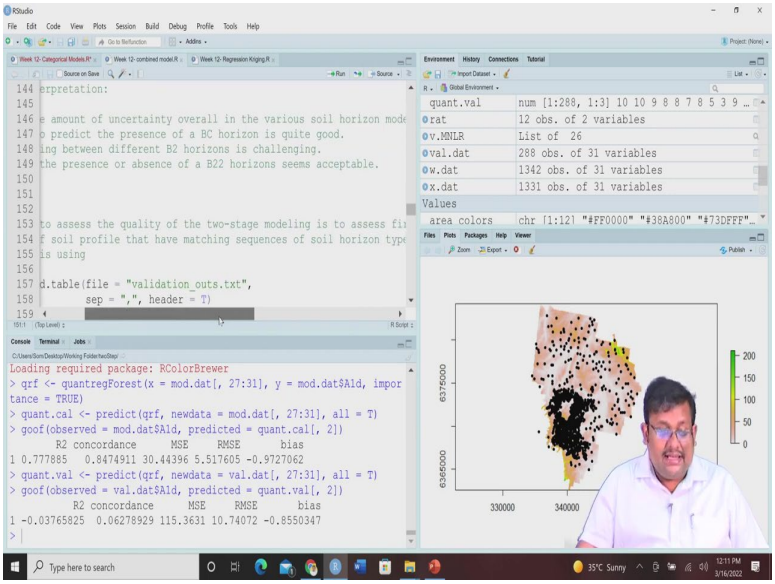
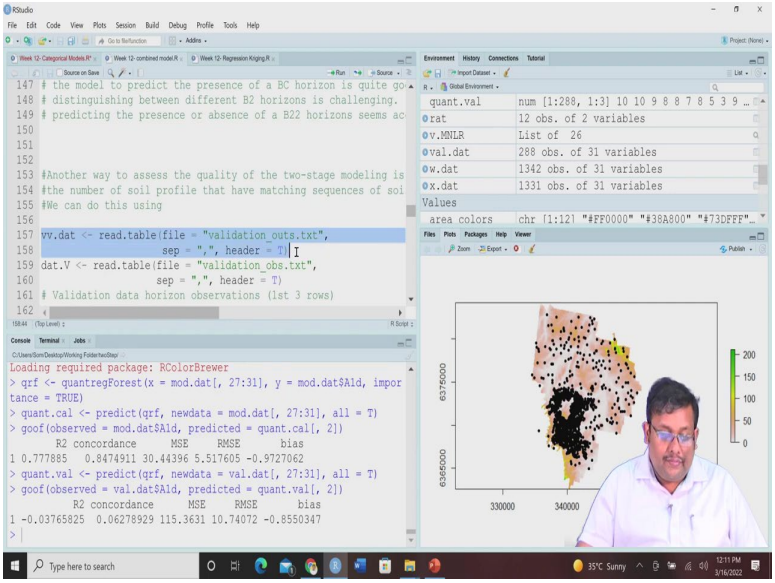
**Machine Learning for Soil and Crop Management**
**Professor Somsubhra Chakraborty**
**Agricultural and Food Engineering Department**
**Indian Institute of Technology Kharagpur**
**Lecture 60**
**Digital Soil Mapping with Categorical Variables (Contd.)**

Welcome friends to this last lecture of this NPTEL online certification course of Machine Learning for Soil and Crop Management in this week 12 we are discussing the Digital Soil Mapping with Categorical Variables. And in my last lecture, we have started discussing about the combined model how to use the R for the, for predicting the presence or absence of any particular horizon and then and if there are presents, then predict their depth.

(Refer Slide Time: 00:55)



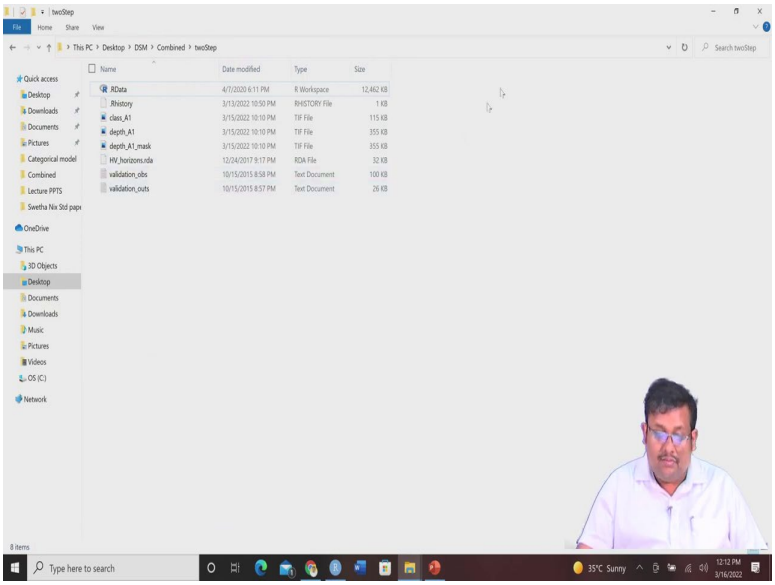



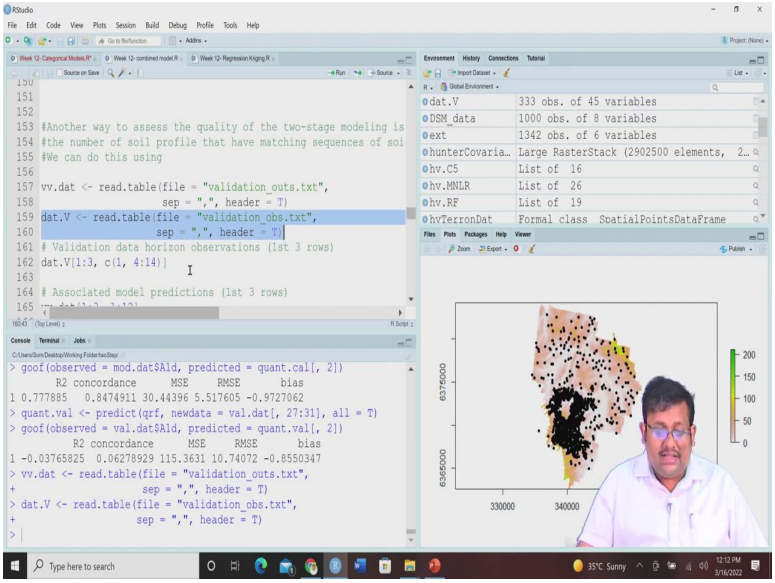


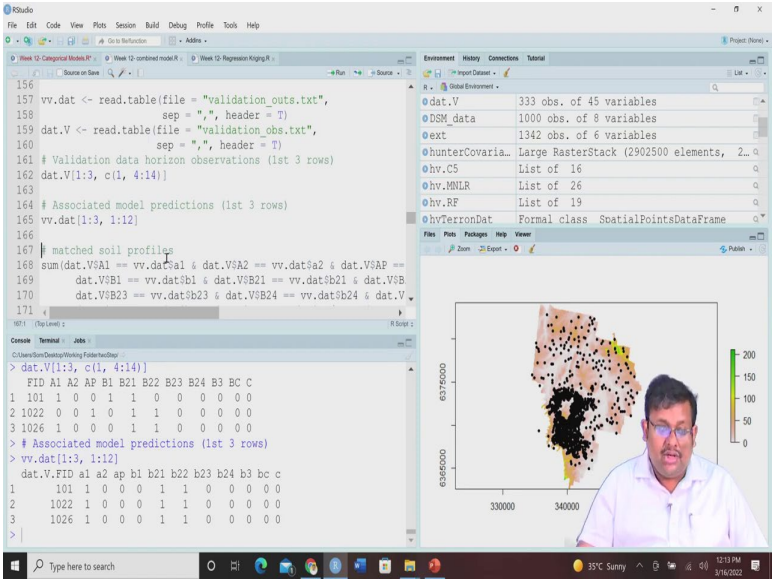

So, let us go back to the script and finish it and so, we have discussed that how to see the presence or absence of a particular horizon using the multinomial logistic regression model and then we are going to we have we have seen that how to using the quantile regression forest, we have seen there the prediction.

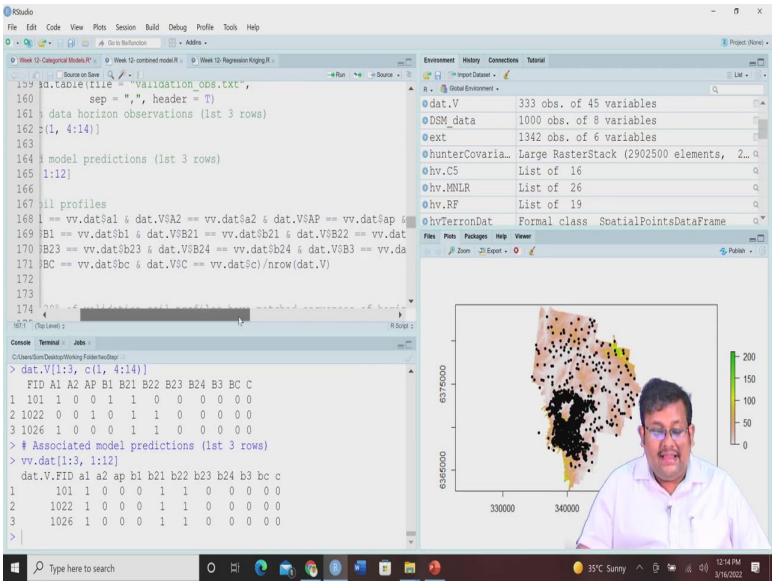
Now, another way to assess the quality of the two-stage modelling is assessing the first the number of the soil profile that have the matching sequence of the soil in the output file. So, let us see here, let us consider this is the let us give a name that is vv dat and let us remember in the twoStep file twoStep folder we have seen a validation output. So, let us rename it as vv dot dat. So, we are reading this file first and then the observation file validation observation file which is already there in this.

(Refer Slide Time: 02:08)




#Another way to assess the quality of the two-stage modeling is
#the number of soil profile that have matching sequences of soi
#We can do this using
vv.dat <- read.table(file = "validation_outs.txt",
sep = ",", header = T)
dat.V <- read.table(file = "validation_obs.txt",
sep = ",", header = T)
# Validation data horizon observations (1st 3 rows)
dat.V[1:3, c(1, 4:14)]
# Associated model predictions (1st 3 rows)
> goof(observed = mod.dat$Ald, predicted = quant.cal[, 2])
R2 concordance MSE RMSE bias
1 0.777885 0.8474911 30.44396 5.517605 -0.9727062
> quant.val <- predict(qrf, newdata = val.dat[, 27:31], all = T)
> goof(observed = val.dat$Ald, predicted = quant.val[, 2])
R2 concordance MSE RMSE bias
1 -0.03765825 0.06278929 115.3631 10.74072 -0.8550347
> vv.dat <- read.table(file = "validation_outs.txt",
+ sep = ",", header = T)
> dat.V <- read.table(file = "validation_obs.txt",
+ sep = ",", header = T)
dat.V 333 obs. of 45 variables
DSM_data 1000 obs. of 8 variables
ext 1342 obs. of 6 variables
hunterCovaria... Large RasterStack (2902500 elements, 2...
hv.C5 List of 16
hv.MNLR List of 26
hv.RF List of 19
hvTerronDat Formal class SpatialPointsDataFrame
6375000
6365000
330000
340000
200
150
100
50
0


vv.dat <- read.table(file = "validation_outs.txt",
sep = ",", header = T)
dat.V <- read.table(file = "validation_obs.txt",
sep = ",", header = T)
# Validation data horizon observations (1st 3 rows)
dat.V[1:3, c(1, 4:14)]
# Associated model predictions (1st 3 rows)
vv.dat[1:3, 1:12]
# matched soil profiles
sum(dat.V$A1 == vv.dat$a1 & dat.V$A2 == vv.dat$a2 & dat.V$AP ==
dat.V$B1 == vv.dat$b1 & dat.V$B21 == vv.dat$b21 & dat.V$B
dat.V$B23 == vv.dat$b23 & dat.V$B24 == vv.dat$b24 & dat.V
> dat.V[1:3, c(1, 4:14)]
FID A1 A2 AP B1 B21 B22 B23 B24 B3 BC C
1 101 1 0 0 1 1 0 0 0 0 0 0
2 1022 0 0 1 0 1 1 0 0 0 0 0
3 1026 1 0 0 0 1 1 0 0 0 0 0
> # Associated model predictions (1st 3 rows)
> vv.dat[1:3, 1:12]
dat.V.FID a1 a2 ap b1 b21 b22 b23 b24 b3 bc c
1 101 1 0 0 0 1 1 0 0 0 0 0
2 1022 1 0 0 0 1 1 0 0 0 0 0
3 1026 1 0 0 0 1 1 0 0 0 0 0
6375000
6365000
330000
340000
200
150
100
50
0


> dat.V[1:3, c(1, 4:14)]
FID A1 A2 AP B1 B21 B22 B23 B24 B3 BC C
1 101 1 0 0 1 1 0 0 0 0 0 0
2 1022 0 0 1 0 1 1 0 0 0 0 0
3 1026 1 0 0 0 1 1 0 0 0 0 0
> # Associated model predictions (1st 3 rows)
> vv.dat[1:3, 1:12]
dat.V.FID a1 a2 ap b1 b21 b22 b23 b24 b3 bc c
1 101 1 0 0 0 1 1 0 0 0 0 0
2 1022 1 0 0 0 1 1 0 0 0 0 0
3 1026 1 0 0 0 1 1 0 0 0 0 0
6375000
6365000
330000
340000
200
150
100
50
0

So, if you go back to this twoStep, you will see that the validation observation and validation outputs are already there. So, we are also reading these validation observations and validation outputs. So, here these validation outputs are created for all the different horizons here we have given previously with the A1 horizon, but here you can see the presence or absence of all the horizons.

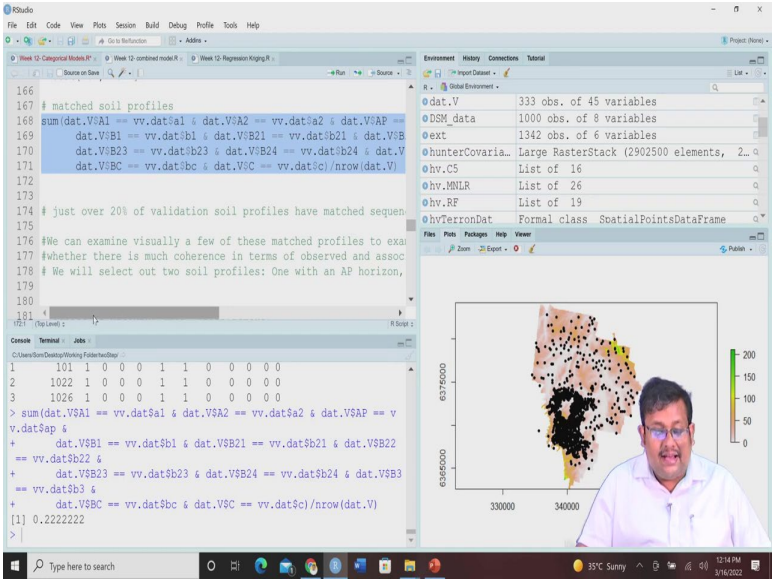
Now, let us just first see the validation data horizon observation for first three dose. So, this is the observation from the observer observed data file, we are going to see the presence or absence of different horizons which are given here in the capital letters and then we can see the associated model prediction for from this output dataset.

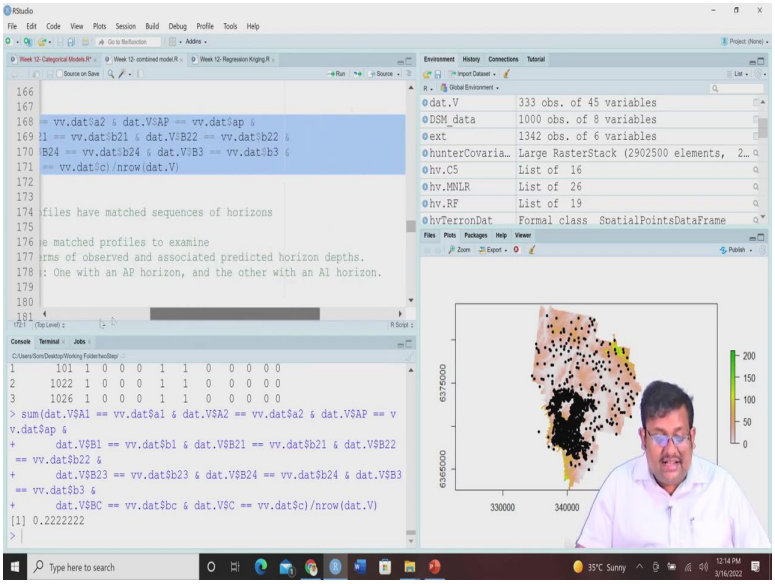
So, we can see this is the associated model output. So, let us see how much among these 1342 observation how much observation we will have, sorry, among these 1342 observation we have kept the 25 person is a validation dataset. Now, in the validation data set, how much what is the percentage of the validation data, which is showing the exact matching between the validation outputs and validation observation.

So, for that, we are going to use this sum function. So, here we want to match the here it is it is the observed file and this is the predicted these are the predicted as you can see this is the validation outputs is small vv and when validation observation is capital V. So, in the observed file, we want to see whether in which cases the original A1 will be matched with the small a1 of the outputs.

So, similarly for a2 then AP and then B1, B21 and all these horizons. So, we want to see how many cases there will be match between the observed the observed data as well as the output data divided by the number of rows in the validation. So, that means, divided by the number of samples in the validation file.

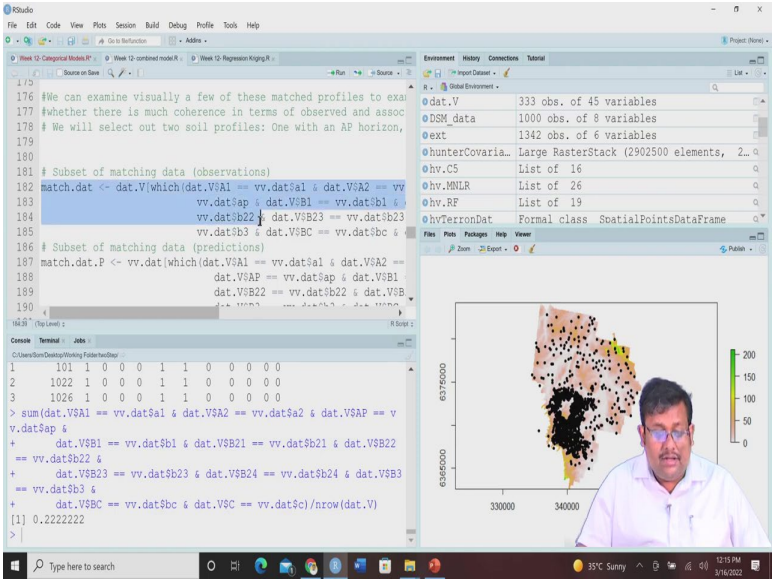
(Refer Slide Time: 04:30)

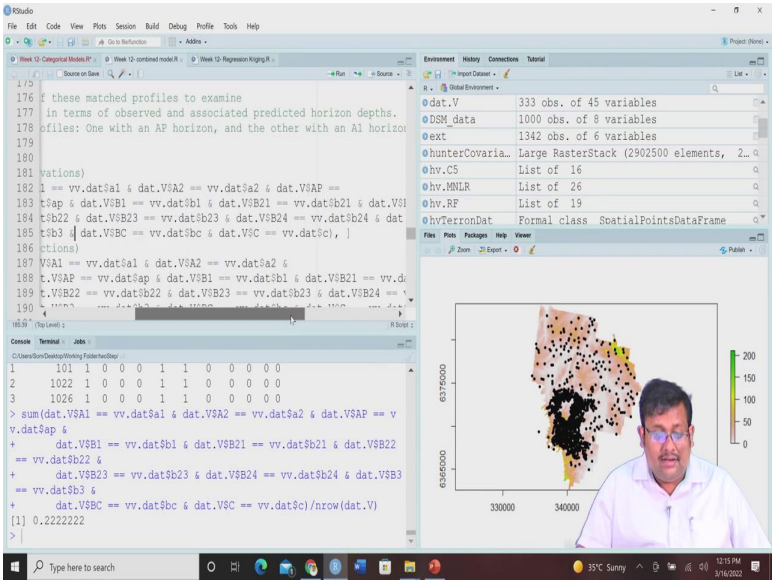


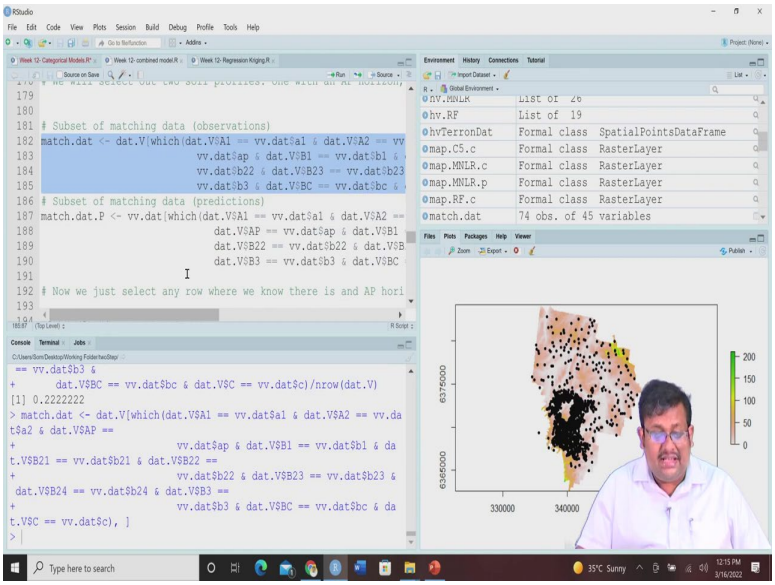




So, we are going to use the sum function and let us see some by the number of rows. So, we are going to see that only 20 percent to 22 percent of the validation soil profiles have matching sequence, we want to see where there is a matching sequence. So, we can examine visually a few of these match profiles to examine, whether there is a much coherence in terms of these observed and associated predicted horizon depth or not, so will do these for using this AP horizon, but you can do it for any other horizon also.

(Refer Slide Time: 05:10)

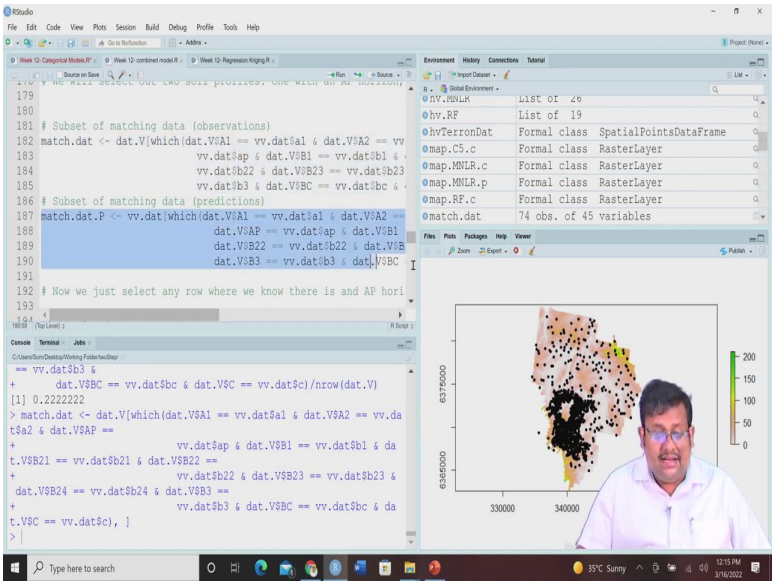


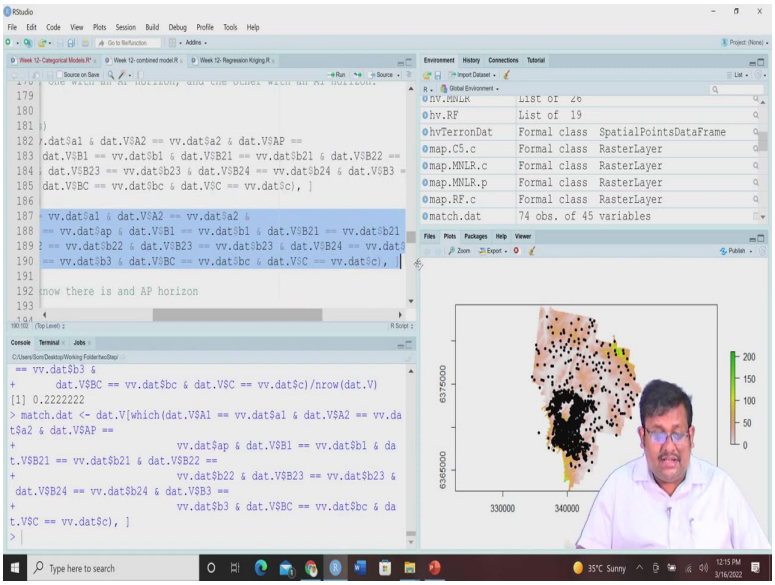



So, let us do this from this observation file, we are going to subset the matching data. So, we are going to these let us do the subset of this matching data. So, we are going to see that so, this is the script for selecting a subset.

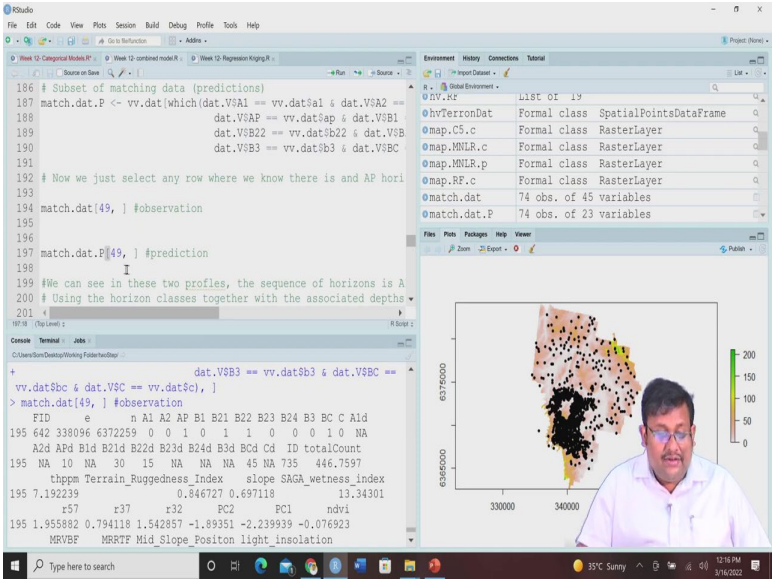
(Refer Slide Time: 05:33)

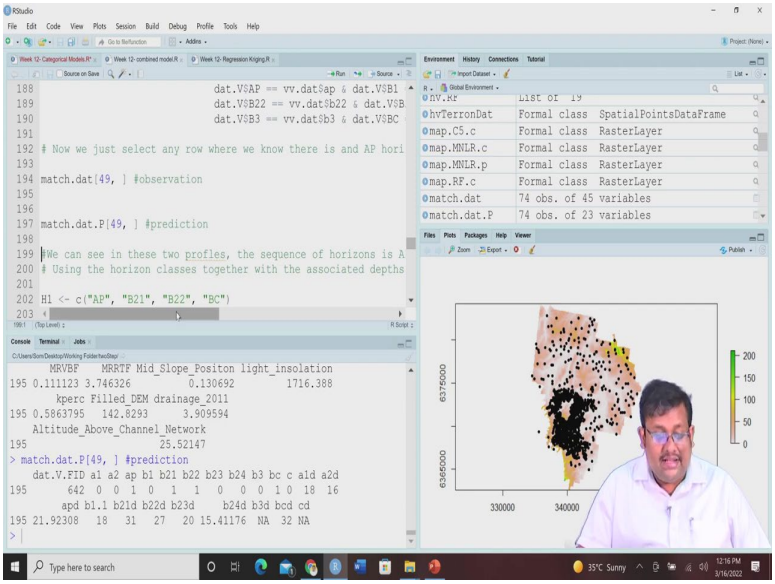


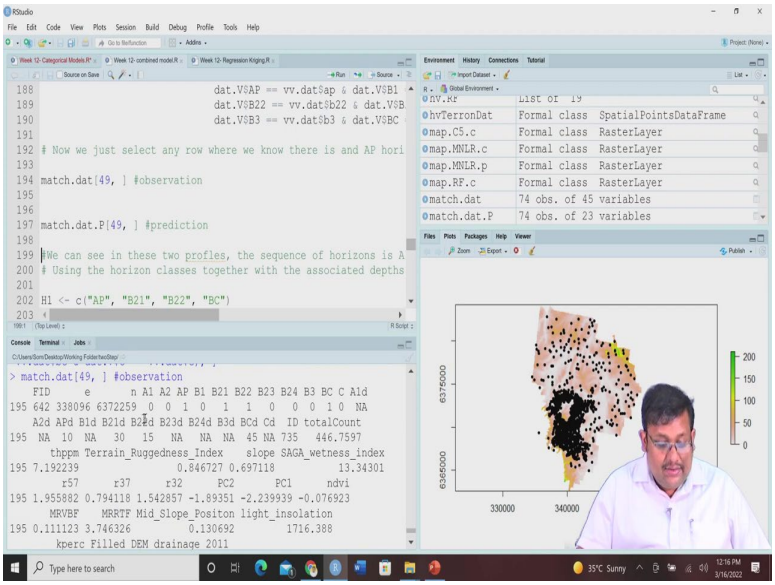



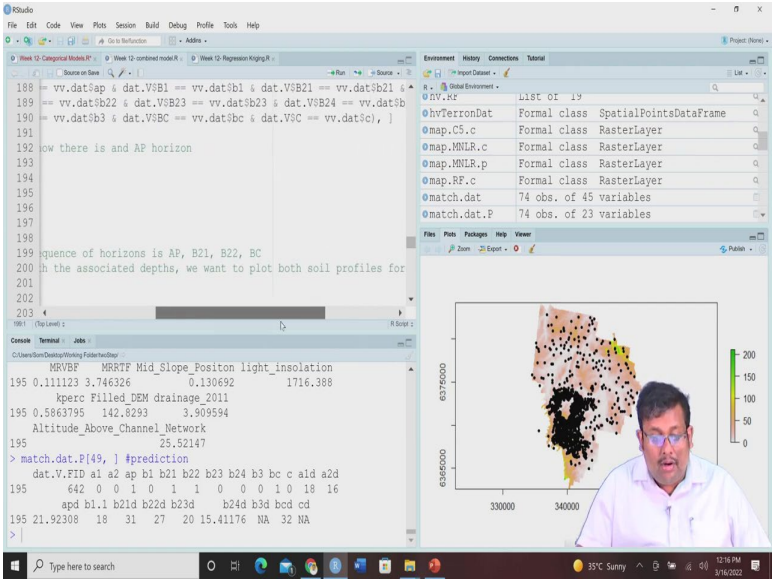

And similarly, for the matching data for the prefer for matching the from the prediction data also, we are going to do this similar subset where there will be perfect match. Now, we just want to select any row where we know there is an AP horizon.

(Refer Slide Time: 05:51)



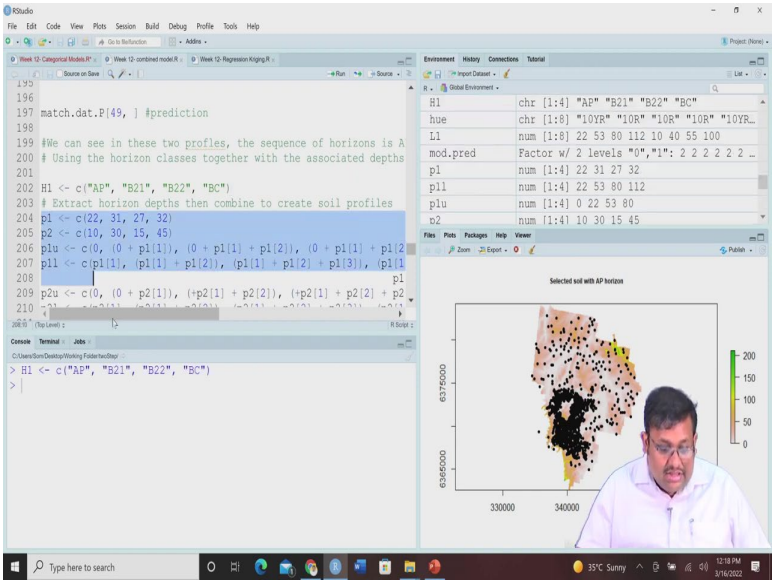



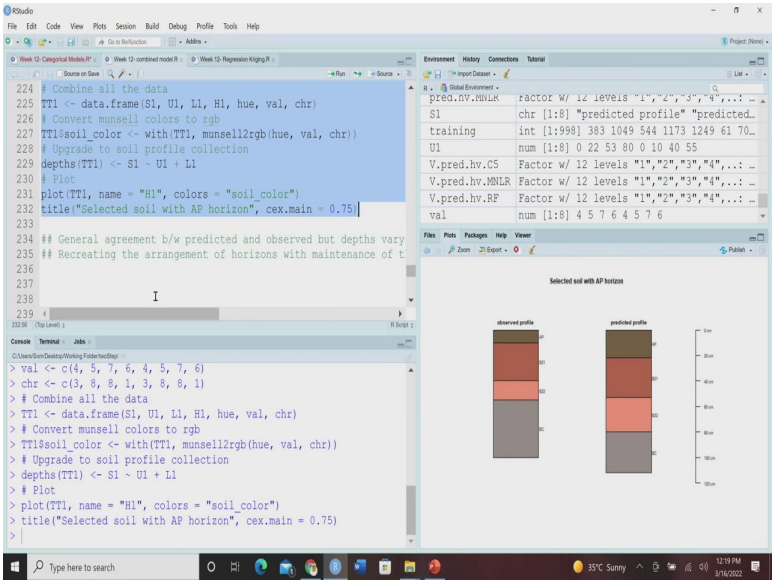



So, we know that on 1549 observation there will be an AP horizon and then let us see, so this is the observation. So, in the observation you can see here, these are observed data, so, there is an AP horizon as we denoted by 1 and if we want to see the prediction, so, here you can see, they are two, one is the from the observation and here when we are adding these match dot dat dot P that is the prediction.

So, we want to see what will be the results for the predicted sequence whether there will be presence or not. So, we can see that these two profiles, the sequence is basically the same. So, here you can see the sequences AP1, B21, B22 and then BC1. Similarly, here also you can see the same sequence AP1, B21, B22 and BC1. Now, using the horizon classes together with the associated depths, we want to plot both the soil profile for comparison. Now, let us so, we know that there is AP horizon B21 horizon, B22 horizon and BC horizon.

(Refer Slide Time: 06:56)



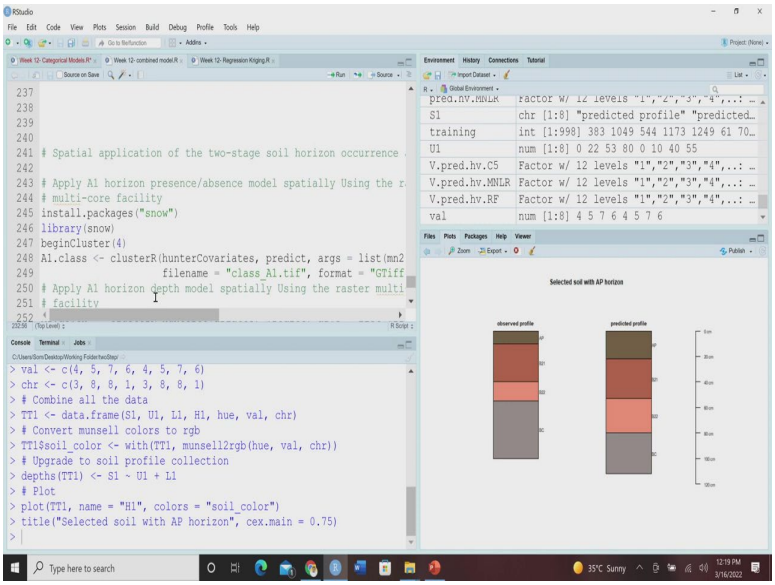



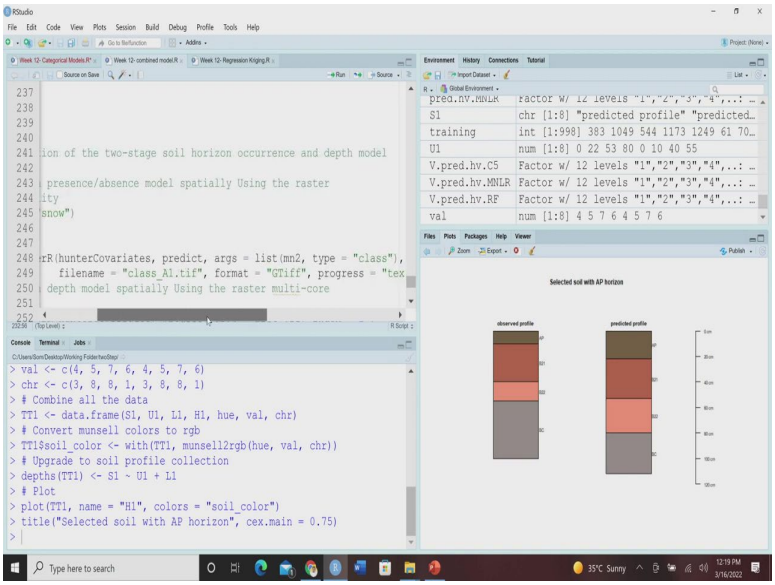

Next is we want to extract these horizon depths and then we combine to create the soil profile will produce the plots for a side by side comparison. So, let us run these codes and see how this has looked like. So, if we do this, they will produce these this is an observed profile and you can see this the predicted profile in the observed profile and predicted profile both of them we are having the same sequence like AP, B21, B22 and BC some AP, B21, B22 and BC however, dip that depth is varying.

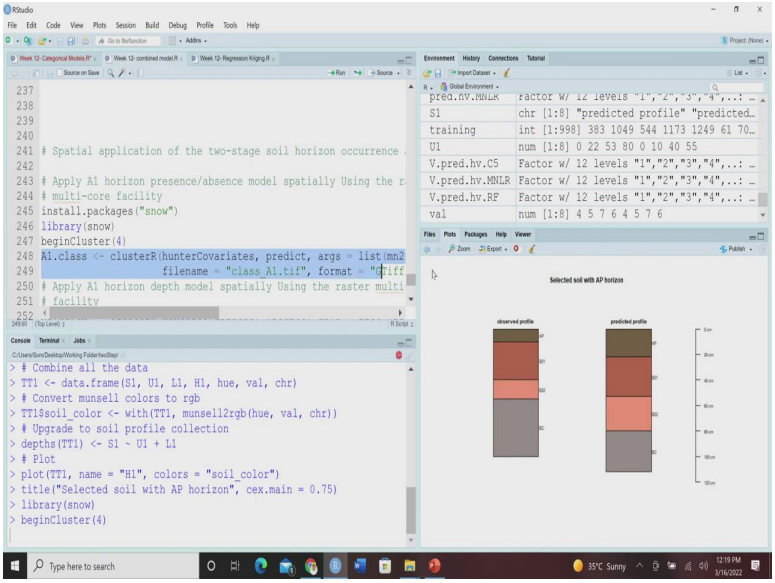
So, what do we understand from this interpretation from these results, we can see that although there is a general agreement between predicted and observed results, but depths vary and recreating the arrangement of the horizons with maintaining of their depth is a challenging task. So, new methods are evolving in the DSM domain and they are trying to see

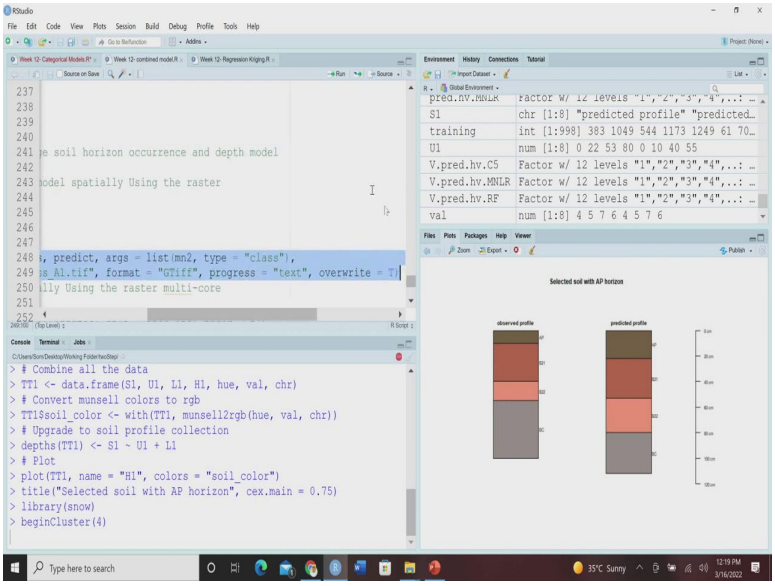
how accurate we can predict and produce these depth for so, that there is a complete match not only for these, sequence of the horizon, but also their predicte the depth, their predicted depth also.

(Refer Slide Time: 08:15)



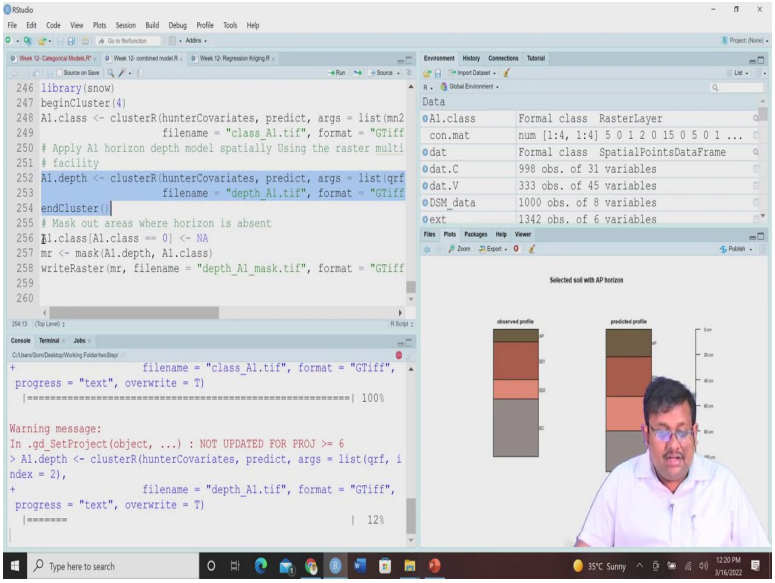



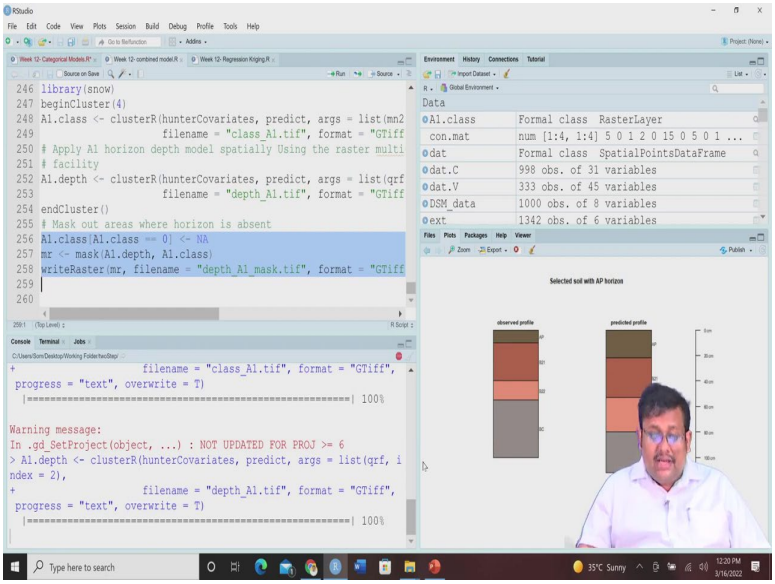


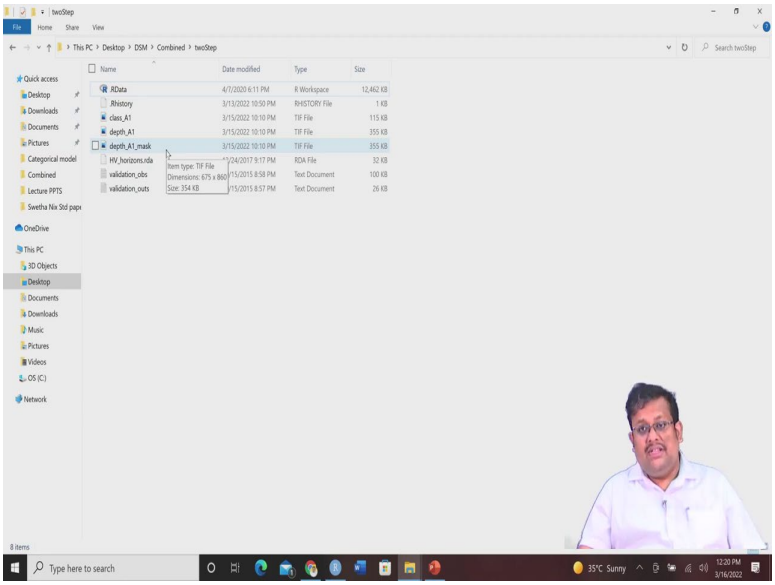




Now, we can also apply now, the once we have developed this model, the next operation is spatial application of this two stage model for the horizon occurrence and their depth. So, for these we are going to use these snow package, please install the snow package, and then let us call this library snow package, and then we are going to use this following cluster and once we here, we are going to use based on the presence of A1 horizon. So, you will see that the maps will be produced in the your working folder.

(Refer Slide Time: 9:00)



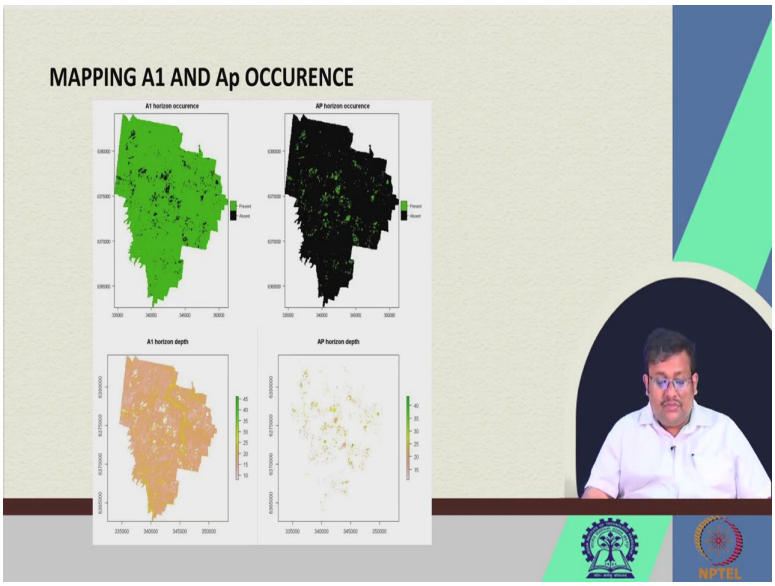



So, and then the depth of there will be also will be produced and also you can mask the out the areas where the horizon is absent. So, of course, you can produce not only the present the map showing the presence or absence of horizons, but also you can predict the depth and mask out those areas where that horizon is not present.

So, the output will be produced in your working folder. So, if you go back to your twoStep, you will see that class A1 that is the prediction the map, which is going to show the presence of A1 and their depth of A1 will be produced at the same time and also the masked file where the absence of the areas where this A1 horizon is absent will be also produced.

(Refer Slide Time: 10:00)



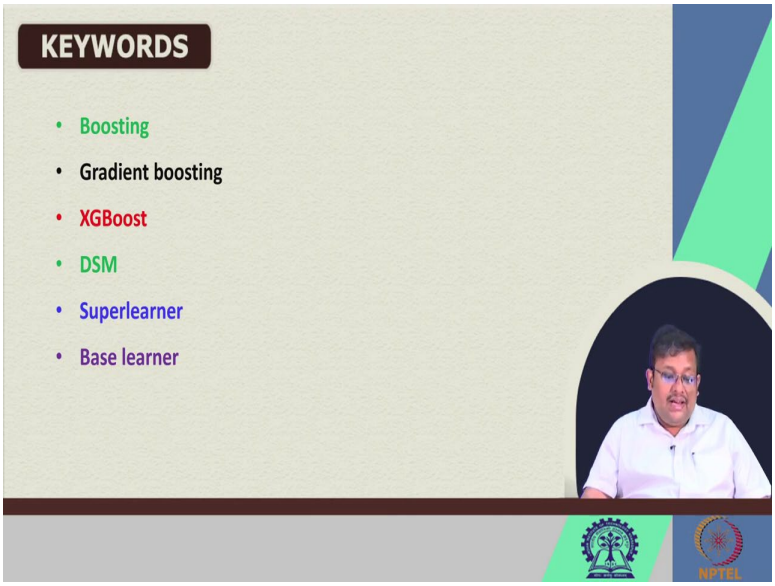

So, just to show you an example, you can see here, if you want to see the mapping of A1 and AP occurrence, you can see this is the A1 horizon occurrence, where black is showing the absent and green is showing the present and AP horizon you can see black is showing the absent and green showing the present.

So, if you want to predict the A1 horizon depth you can predict from here and also the AP horizon depth you can predict from here. So, the Ap so this is how you can use this twoStep model to produce the, you can produce the two step model for not only the mapping of the of the presence for the presence of any horizon, but also you can predict their depth also.

(Refer Slide Time: 10:45)

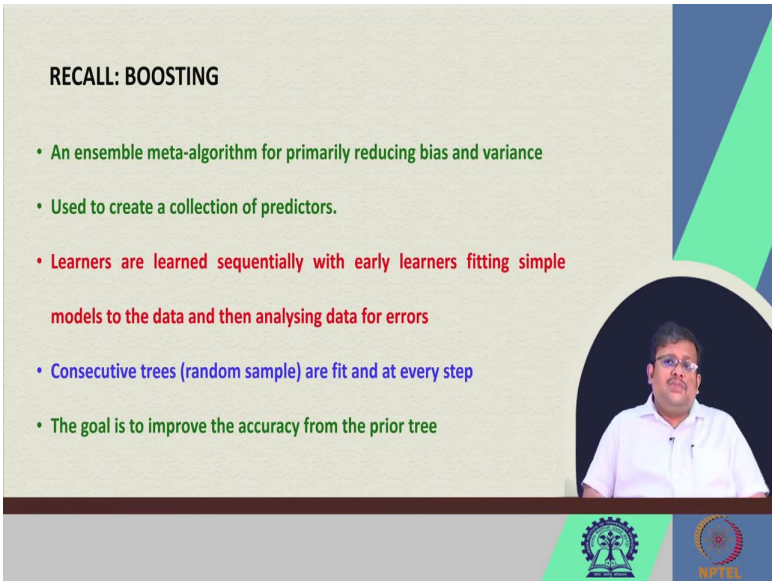

So, guys, we are at the final, lecture. So, here in this lecture also apart from this, we are going to discuss some important machine learning models and I will also discuss the digital soil mapping with AIML with two case studies.

(Refer Slide Time: 11:00)




So, these are the keywords for this lecture, like Boosting, Gradient boosting, XGBoost, then Digital Soil Mapping, then Superlearner and base learner these are some of the keywords for this lecture.

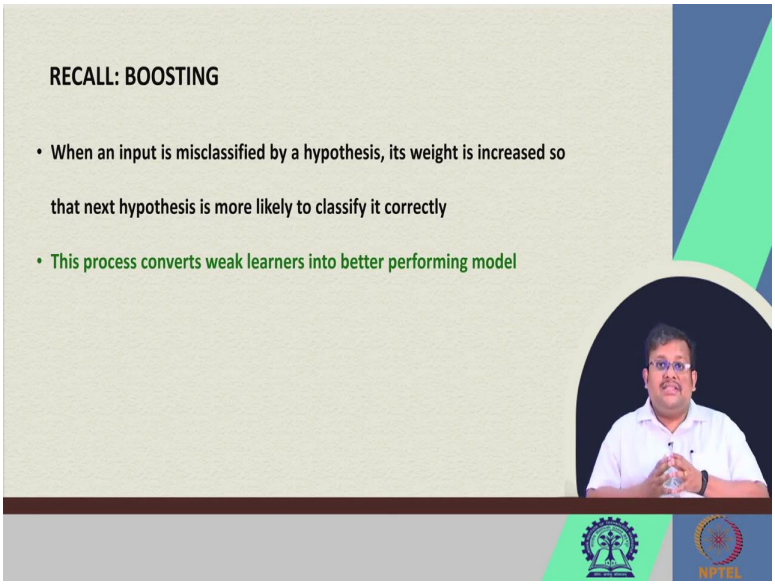
(Refer Slide Time: 11:14)




And let us start with the Boosting. So, we have already discussed what is boosting. So, if you can recall that boosting is an ensemble meta algorithm for primarily reducing the bias and variance and it is generally used to create collection of predictors. So, basically what it does? Learners are learned sequentially with early learners fitting simple model to the data and then analysing the data for errors.

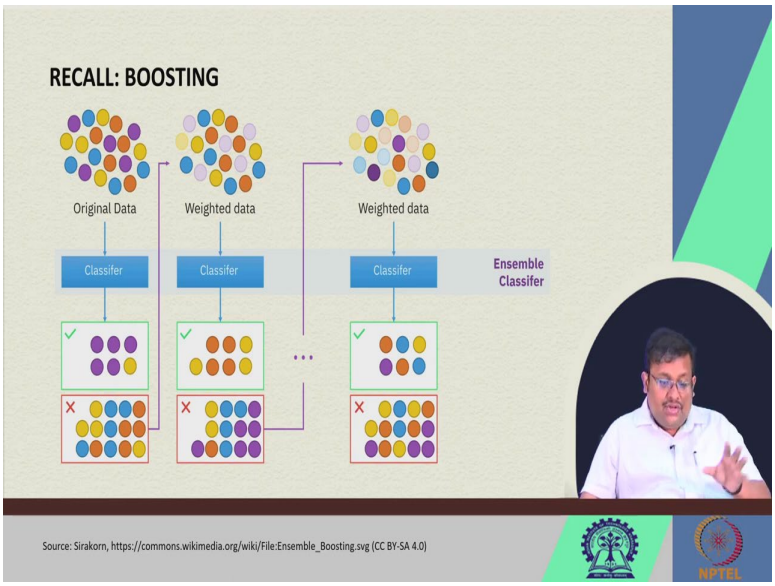
So, of course if we classify if we use a classifier or learner to classify or predict any model, then obviously, some amount will be correctly classified or correctly predicted and some amount will be, misclassified all produce the errors. So, based on these misclassified samples, so consecutive, trees are being fit at every steps, and the goal is of this boosting is to improve the accuracy from the prior tree.

(Refer Slide Time: 12:13)




Now, we know that when an input is misclassified by a hypothesis, its weight is increased. So, that the next hypothesis is more likely to classify it correctly. And this process converts weak learners into better performing model. So, we already know this boosting thing which we have already discussed in our previous lectures.

(Refer Slide Time: 12:35)

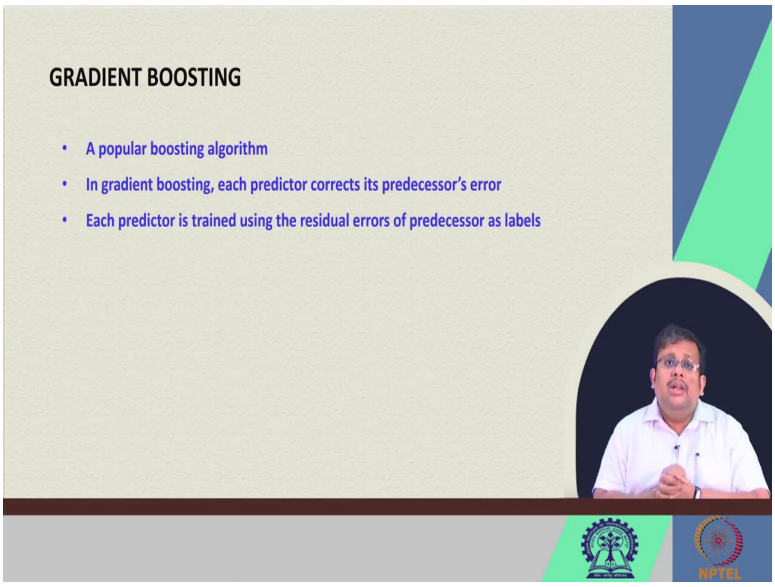



Now, this is a boosting thing in a pictorial view of these boosting algorithm. So, we have original data, we have classifier, so we can correctly classify some samples and some samples will be wrongly classified or misclassified.

So, we will give them the proper weightage and then go and then pass it to the second classifier, and among the second classifier, some of them will be correctly classified, some of

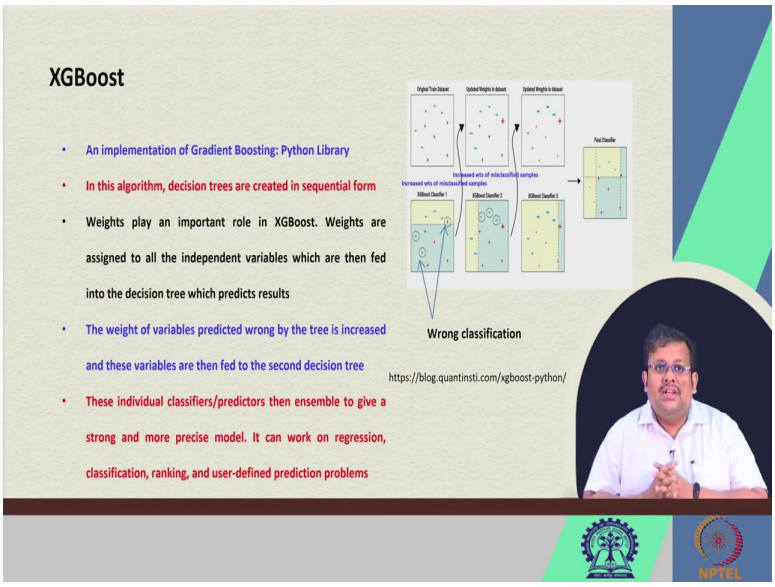
them will be misclassified, and then again, they will be passed to the that consicute the next classifier in this way, at a final step and ensemble classifier will be there, which will classify all these things all these observations correctly. So, in the step by step it improves the classification accuracy. So, this is known as the boosting algorithm.

(Refer Slide Time: 13:24)




Now, there is a algorithm called gradient boosting. So, it is a very popular boosting algorithm in gradient boosting each predictor corrects its predecessor's error. Now, each predictor is trained using the residual errors of the predecessor's as labels. So, this is a gradient boosting this is a very popular.

(Refer Slide Time: 13:49)



And one of the variant of this gradient boosting is called the XGBoost. So, it is an implementation of this gradient boosting and of course, it is an algorithm and also a Python library, Python Software which we have already discussed in our live session. So, these this XGBoost is nowadays a very, very widely used and very popular machine learning algorithms. I want to tell you about this that is why we are discussing this.

So, this XGBoost is an implementation of the gradient boosting. So, in this algorithm decision trees are created in sequential form and weights play an important role in XGBoost how? Because weights are assigned to all the independent variables which are then fit into the decision tree which predict the results.

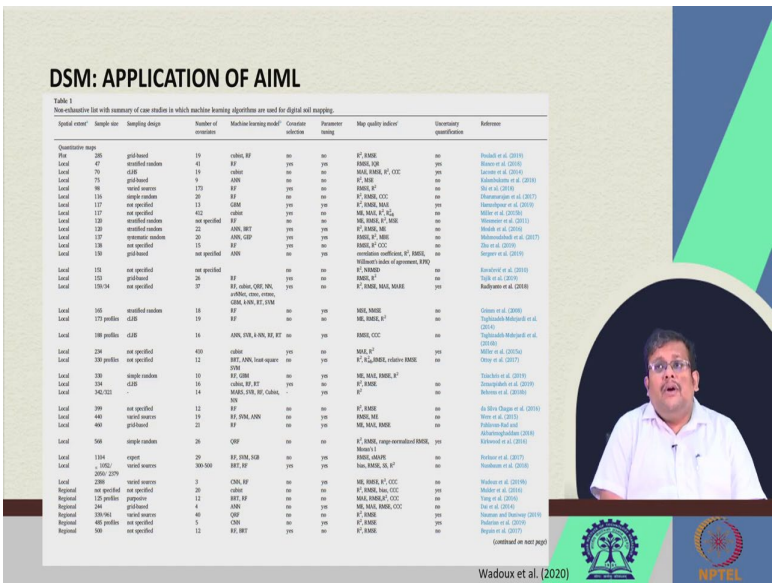
Now, the weight of the variables predicted wrongly by the tree is increased in the subsequent steps and then they are fet into the decision tree just like in the boosting algorithm. So, here the weights of the variables which are wrongly predicted. We are going to increase their weight and then pass it through the next tree to in the second decision tree and step by step we are doing the same thing and these individual classifiers or predictors, then ensemble to give a strong and more precise model, it can and this is called the XGBoost model.

And this XGBoost model works very good on regression classification, ranking and user defined prediction problems. So, here you can see, this is an original training data set contains both plus and minus. So, we are putting these in the first classifier XGBoost classifier 1 which classified rightly this negative sign and this positive sign. But, there are some misclassification as we have, identified, this plus and this minus are misclassified.

Now, these wrong classification will give will be having the increased weights and then we are going to update their weights and we will pass it pass them through the XGBoost classifier 2 in the second step they will again, rightly classified some of them and wrongly classified some of them and then again, we will update their weights, increase their weights of this misclassified sample and again pass through the XGBoost classifier 3 and all once all these classifiers are trained and final classifier will give you very good classification accuracy.

So, this is called the XGBoost, the reason I am telling you the XGBoost is the XGBoost is one of the important machine learning algorithm which is being used nowadays for Digital Soil Mapping.

(Refer Slide Time: 16:42)



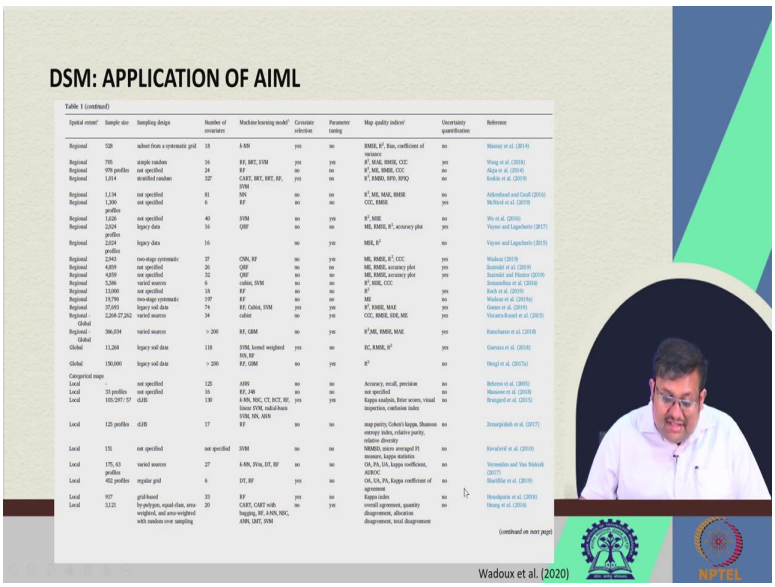

Now, let us consider the, if you consider all the machine learning models, which you have discussed we may be curious that which are useful for Digital Soil Mapping. Now, (())(17:04) all in 2020 they have made a extensive literature review and they have found the application they have basically tabulated the studies where they have used different types of machine learning and deep learning models to predict different types of to create the Digital Soil Mapping.

So, here you can see along with the references also. So, here you can see they have used cubist, random forest, artificial neural network gradient boosting machine just we have talked about then boost regression tree then quantile regression forest and all of these.

So, you can see that most of it we have already covered we have among these we have already covered cubist we have already covered random forest, we have covered nearest neighbour, we have already covered the quantile regression forest, we have covered random forests, we have covered support vector regression, we have covered XGBoost, we have covered neural network, we have covered a lot of these different types of machine learning and deep learning models, which they have already performed.

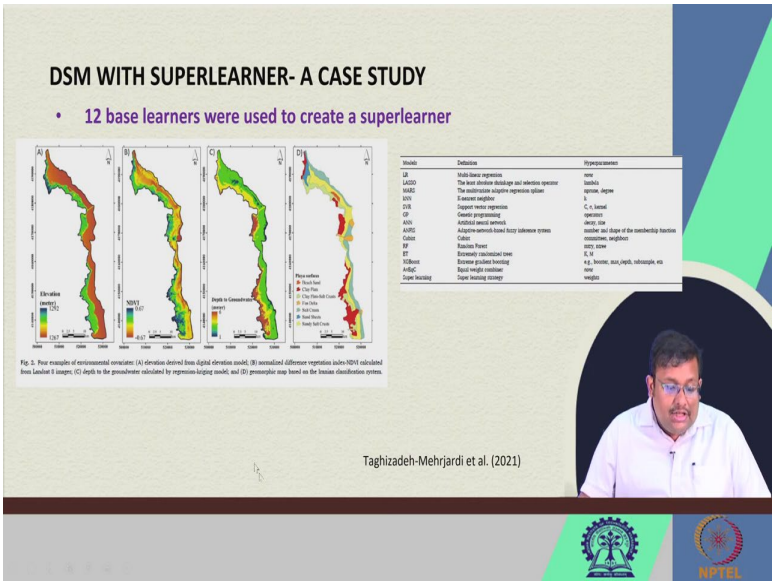
So, you can see that, different types of convolutional neural networks. So, different types of machine learning and deep learning methods they have used for producing the Digital Soil Map both at local scale as well as in regional scale and they are sample size varied from very small number of samples to large number of samples, they have utilised different types of sampling design number of covariates they have also used for digital soil mapping and they have done covariate selection and also they have done the uncertainty quantification also.

(Refer Slide Time: 18:53)




So, these are different types, these are the list of the studies which they have, which used different types of machine learning algorithm. So, it is quite clear that how these machine learning algorithms is improving and along with the deep learning algorithms is augmenting the use of a different digital soil mapping exercise and people are using these different types of machine learning as well as deep learning algorithms to increase the accuracy of their map. And as these new, new algorithms will evolve the accuracy of the of the predicted map or will be also augmented and of course, they will give you more and more high resolution and accurate mapping of soil properties.

(Refer Slide Time: 19:56)



So, just to show you one example, here one example where Mehrjardi et al in 2021 they have used a super learner. So, I will talk about the Super learner, super learner is an advanced ensemble machine learning model they have used for creating high resolution map of multiple soil properties. So, they have used different 12 base learners, base learners means individual models, machine learning models to create these super learners.

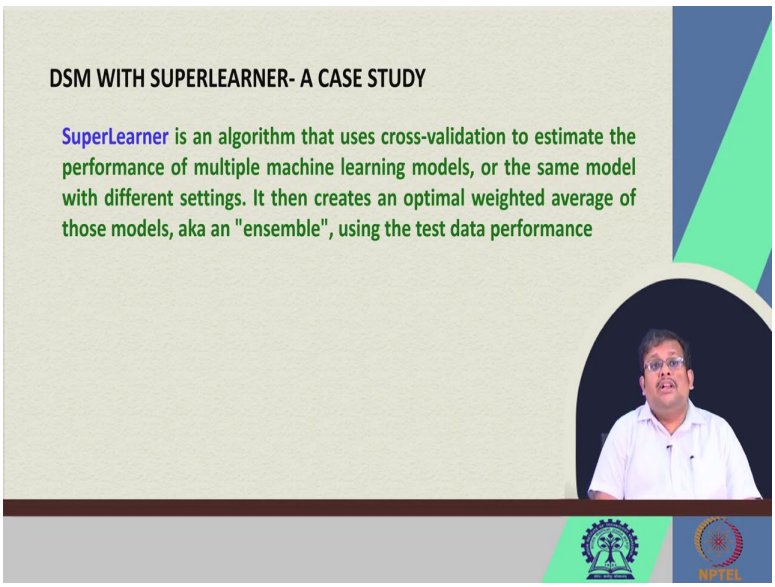
So, these are these 12 models they have used linear regression, then lasso, then multivariate adaptive regression, regression plant spline, K nearest neighbourhood, then support vector regression, then genetic programming artificial neural network, cubist, random forests then extremely randomised trees XGBoost, AV QC then super learning. So, all of them they have used and then they produce the superlearner.

So, ultimately they tried to increase the accuracy of the predicted map. So, these are some examples of the environmental covariates they have used as you can see they have used the elevation derived from the digital elevation model. And then this is the Normalised Difference Vegetation Index or NDVI is calculated from the landsat 8 image and then depth to groundwater, which they have produced using the regression training model.

Now, you see how they are connected now, all the knowledge you have gathered so far and how we can utilise all of them to produce these maps. Now, you can I hope that now, you are getting more and more confidence of how to execute these things. So, here you can see depth to groundwater, you can produce this map using regression taking.

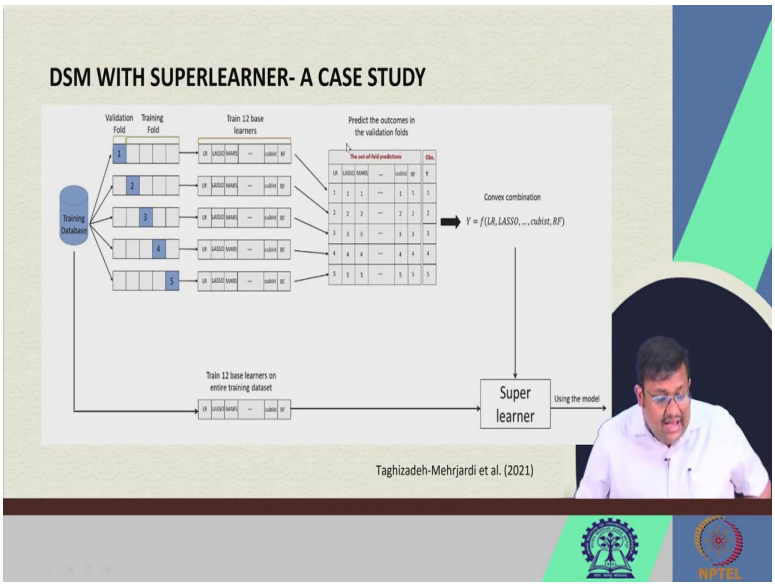
And then finally, you can see here geomorphic map. So, they have used several, covariates I am just giving here four examples. So, using these covariates, and the different types and super learner, so how to develop this super learner we will discuss in the next slide, but for developing the super learner, you need some base learners. So, these base learner model were utilised for developing the super learner.

(Refer Slide Time: 22:18)




Now, what is a SuperLearner? SuperLearner is an algorithm that uses the cross validation to estimate the performance of multiple machine learning models or the same model with different settings. And it then creates an optimum weighted average for those models also known as ensemble models using the test data performance.

(Refer Slide Time: 22:40)




So, this is the architecture of the super learner, which they have used. So, here you can see the training data set and then there will be K fold cross validation with all the best learners and finally, they will be combined them into the validation fold. So, here you can see how the step by step so, in the training data set, they have suppose they have created 5 fold cross

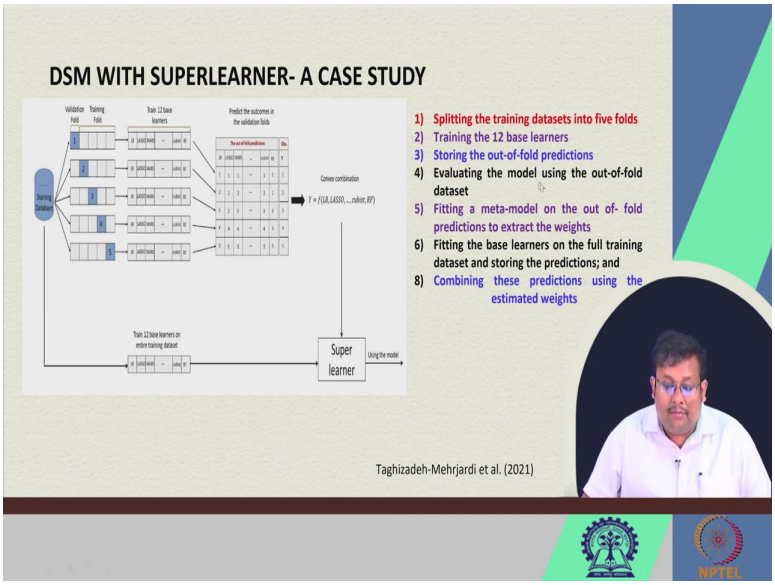
validation. So, in the 5 fold cross validation, they have kept 1 validation 4 fold and 4 fold at the training.

So, and although also using these training set, they have training a trained all these 12 base learners starting from linear regression Lasso, Mars 2, Cubist Random Forest. So, for each of this step, they kept out one fold and they trained these 12 models using the rest of the training folds and after each of them, they will produce the outcome based on these holdout validation fold.

So, for example, here you can see the using the holdout validation fold, they have predicted the value as 1 in case of linear regression in case of Lasso, they have produced 1 as the predicted value and then Mars 1 and all these things. So, this will be the predicted outcomes in the validation folds. And ultimately the observation will be combined with these are will be considered as the predictors to predict the original observation using a convex combination in this case, so it will be called a meta learning model.

So, using this meta learning model, there weights will be optimised. So, here you can see using these metrics of the results, the out of fold predictions, then they will produce these the meta learning meta learning model at the same time, they will also train these 12 base learners on the entire data set. And then they will adjust their using these predicted values, they will use these weights to do final to produce the final result. So, using the weights which they will optimise in this state, they will assign these weights to this final prediction and ultimately, they will get these results.

(Refer Slide Time: 25:06)



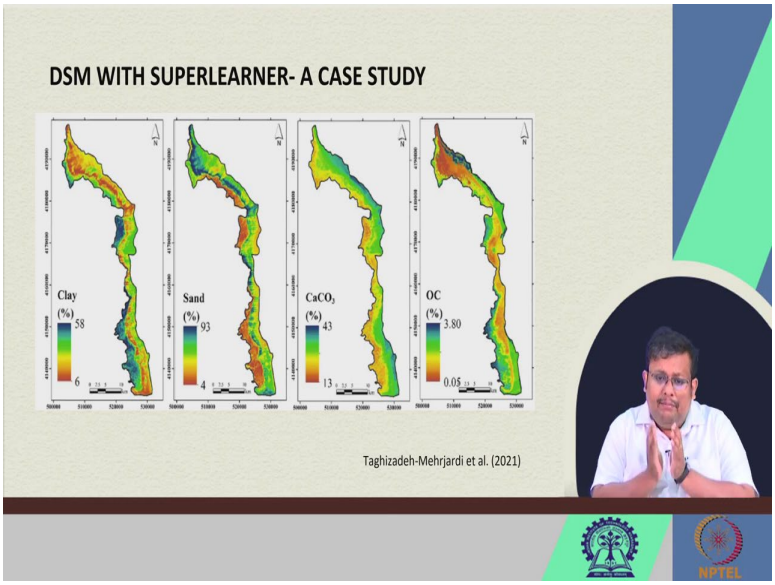

So, these are the steps. First of all splitting the training data set into 5 folds, we have seen that, and then training these 12 base learners, and then storing the out of fold predictions just like here, we are storing these out of fold predictions. And then in the forth step, we are evaluating the model using these out of fold predictions. Then we are evaluating this model out using the out of fold prediction. And then in the fifth stage, we are doing the meta model using the on the out of fold predictions to extract the weights.

So, from here from this meta model, they have used the generalised linear model in this instant, but you can use any other model. So, using this meta model, they have extracted their weights, and then they have fitted these base learners on the full training data set and stored their predictions, and then combine these predictions. So, from each of these base learners, they have got these predictions. And then finally, they will combine this prediction using the estimated weights from this meta learning model.

So, this is how they will be finally predicting the results. So, this is the workflow of the super learner. So, again, first we do the validation, splitting the training dataset into folds, several folds. And then we train the models, all the base learners using the out of fold predictions, and then evaluate based on the out of fold data set.

And then we fit the meta model to the original observation using the predict and then we extract their weights once we extract their weights, again, we will train the data set entire data set using this 12 base learners. And then their predictions will be combined together by using these extracted weights. And this is the super learner strategy.

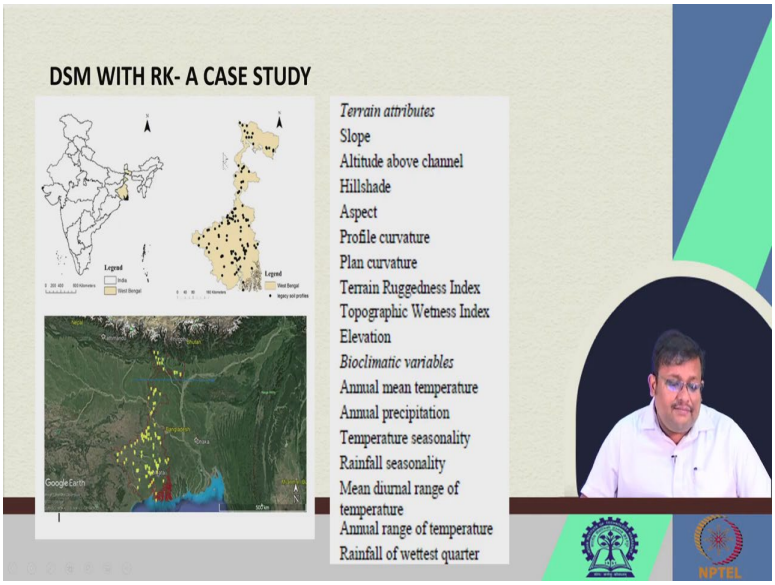
(Refer Slide Time: 27:05)




So, using these SuperLearner model, they have produced high resolution maps of multiple soil properties, I am just giving here for examples, the clay percentage map, sand percentage map, calcium carbonate percentage map and organic carbon percentage map. So, now a days people are trying new, new methods, new, new advanced methods in AI domain as well as machine learning domains.

So, that they can produce high resolution maps and they can improve one of the best advantage of the SuperLearner is it always produce better results it never produce any result which will be worse from the best performing base learner. So, here you can see there are 12 performing best learners. So, it is assumed that the SuperLearner will always perform better than these best learners, it will never perform worse than that of any I mean the best individual best learner. So, this is the advantage of SuperLearner.

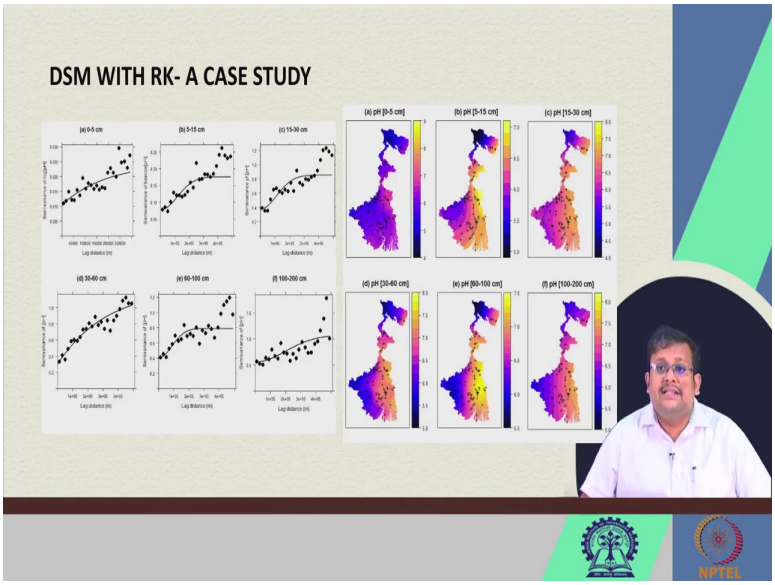
(Refer Slide Time: 28:08)




And another case study we have we have in our group, we have collected, we have gathered the soil data information from the state of West Bengal of India and then we are extracted the different types of terrain parameters and bioclimatic variables are been the terrain parameters we have extracted the slope or Altitude Above Channel Network, Hill shade, Aspect, Profile curvature, Plane curvature and then Terrain Ruggedness Index, Topographic Wetness Index and Elevation all of these were extracted from the DEM file for this region of West Bengal state of India.

And also we have downloaded the Annual mean temperature, Annual precipitation, Temperature seasonality, Rainfall seasonality, Mean diurnal range of temperature, Annual range of temperature, Rainfall of a wettest quarter, so we have downloaded this data bioclimatic variables as well as the terrain variables.

(Refer Slide Time: 29:08)




Using these covariates, we have produced the regression Kriging of different soil properties soil pH at different depths as you can see here, 0 to 5 centimetre, 5 to 15 centimetre, 15 to 30 centimetre 30 to 60, centimetres 60 to 100 centimetre, 100 to 200 centimetre in this regression Kriging we have used the random forest model to predict these high resolution maps of soil pH.

So, guys, this makes the end of this course and I hope that within this limited time period, I tried to as much as I can I try to cover all these machine learnings giving their basic overview and try to show you some agriculture related examples of course, this is the inception, this is basically an eye opener for those who are new to this domain and you should explore more and more, I have already discussed different types of the references, and there are plenty of literature, plenty of research papers and plenty of online courses that they are which are focusing on these machine learning models.

You please go through these machine learning courses for enrich, more enrich yourself and then you can also utilise those data set for the, your own data set for agriculture related data set both with the crop dataset as well as the soil data set, you explore, and if you can explore and you will see there is a huge sea of information in this domain of artificial intelligence and machine learning, which you can utilise in your own study.

So, guys, I really thank you for staying, for paying your attention in these lectures. And you have asked so many good questions in our first live session, I hope that you will be also asking some good questions in our second live session.

(Refer Slide Time: 31:18)


And again, and finally, I would like to thank my PhD student, Mr Subhadip Dasgupta and Mr. Madan Jatiya, for helping me for offering this course and they have helped me for drafting the assignments, weekly assignments. And so, guys I hope that this course has met your expectation.

And, again, I will request to explore more in this domain and utilise in your research, to gain more and more knowledge and to advance the agricultural operations and there is a huge amount of opportunities in the agricultural sector, specifically crop and soil and utilise this knowledge for advancement of crop and soil characterization. Thank you guys.

(Refer Slide Time: 32:22)

And these are the references. And I thank you again for your participation in this course. And I wish you all the best for your final exam. Thank you.