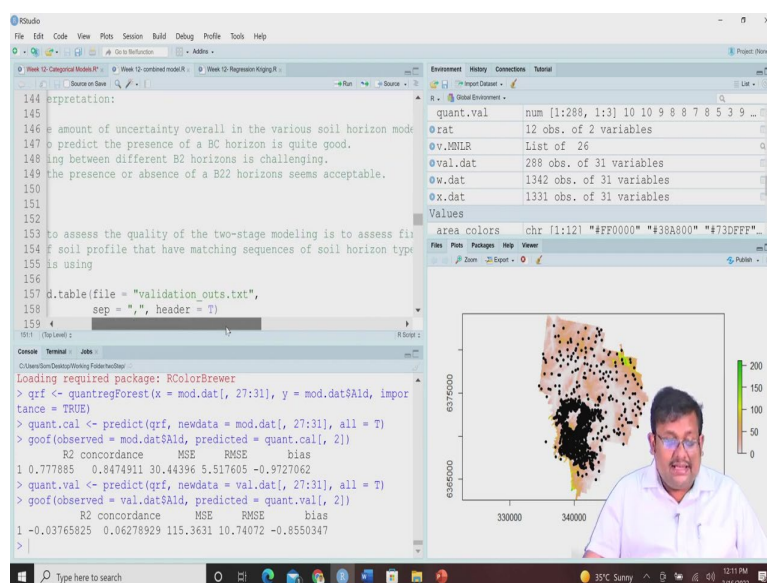
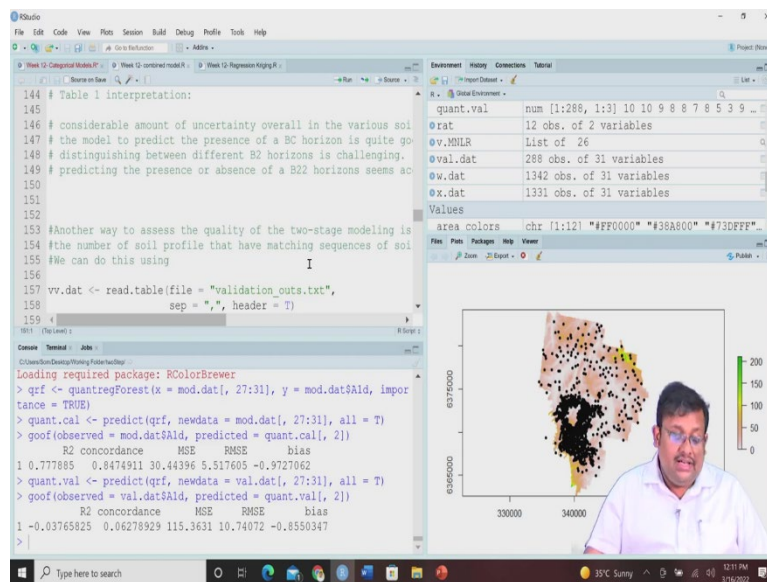


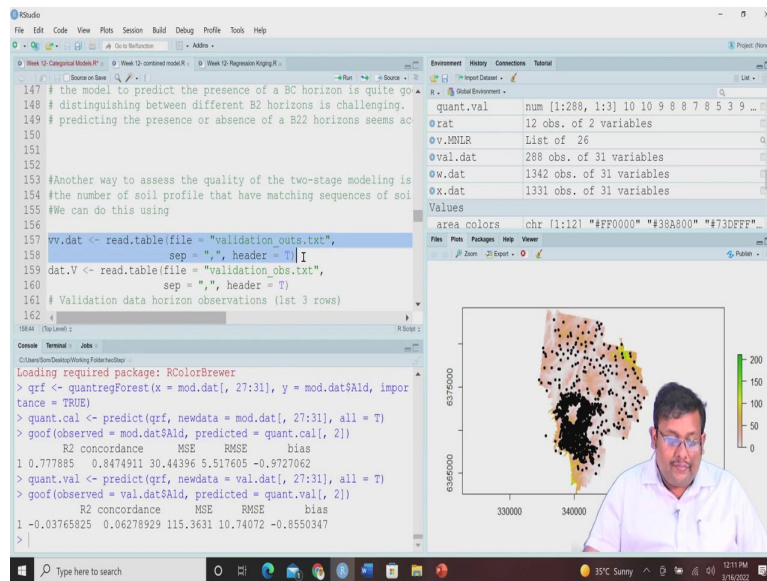
**Machine Learning for Soil and Crop Management**  
**Professor Somsubhra Chakraborty**  
**Agricultural and Food Engineering Department**  
**Indian Institute of Technology Kharagpur**  
**Lecture 60**

**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)

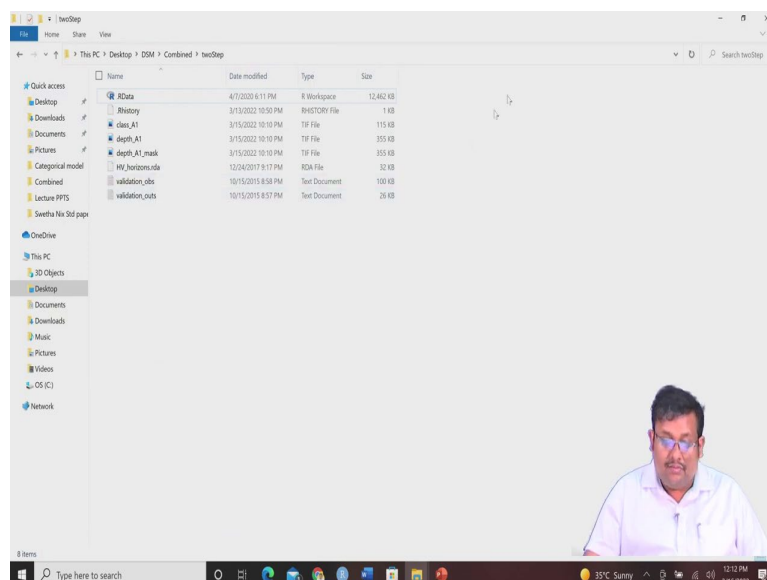




So, let us go back to the script and finish it and so, we have discussed that how to see the presence or absence of a particular horizon using the multinomial logistic regression model and then we are going to we have we have seen that how to using the quantile regression forest, we have seen there the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



RStudio interface showing R code in the editor and a map of soil profile locations. The code includes:

```
157 vv.dat <- read.table(file = "validation_obs.txt",
158 sep = ",", header = T)
159 dat.V <- read.table(file = "validation_obs.txt",
160 sep = ",", header = T)
161 # Validation data horizon observations (list 3 rows)
162 dat.V[1:3, c(1, 4:14)]
163
164 # Associated model predictions (list 3 rows)
165 vv.dat[1:3, 1:12]
```

The console shows the output of `goof` and `predict` functions, including R2 concordance, MSE, RMSE, and bias values. The environment pane lists objects like `dat.V`, `DSM_data`, `ext`, `hunterCovaria`, `ohv.C5`, `ohv.MNLR`, `ohv.RF`, and `ohv.TerronDat`. The map shows a geographical area with a color scale from 0 to 200.

RStudio interface showing R code for matching soil profiles and a map of soil profile locations. The code includes:

```
167 # matched soil profiles
168 sum(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.dat$a2 & dat.V$a3 == vv.dat$a3 &
169 dat.V$b1 == vv.dat$b1 & dat.V$b2 == vv.dat$b2 & dat.V$b3 == vv.dat$b3 &
170 dat.V$b4 == vv.dat$b4 & dat.V$b5 == vv.dat$b5 & dat.V$b6 == vv.dat$b6 &
171 dat.V$b7 == vv.dat$b7 & dat.V$b8 == vv.dat$b8 & dat.V$b9 == vv.dat$b9 &
172 dat.V$b10 == vv.dat$b10 & dat.V$b11 == vv.dat$b11 & dat.V$b12 == vv.dat$b12)
```

The console shows the output of `dat.V[1:3, c(1, 4:14)]` and `vv.dat[1:3, 1:12]`, displaying soil profile data in a table format. The environment pane and map are identical to the previous screenshot.

RStudio interface showing R code for reading validation data and a map of soil profile locations. The code includes:

```
159 read.table(file = "validation_obs.txt",
160 sep = ",", header = T)
161 # Validation data horizon observations (list 3 rows)
162 dat.V[1, 4:14]
163
164 # model predictions (list 3 rows)
165 vv.dat[1:12]
```

The console shows the output of `dat.V[1:3, c(1, 4:14)]` and `vv.dat[1:3, 1:12]`, displaying soil profile data in a table format. The environment pane and map are identical to the previous screenshots.

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)

The screenshot shows the RStudio interface. The script editor contains R code for matching soil profiles. The console shows the execution of the code, resulting in a matrix of 0s and 1s. The Environment pane on the right lists various objects, including a matrix of 333 observations and 45 variables. A scatter plot in the bottom right shows the spatial distribution of the matched profiles, with a color scale from 0 to 200.

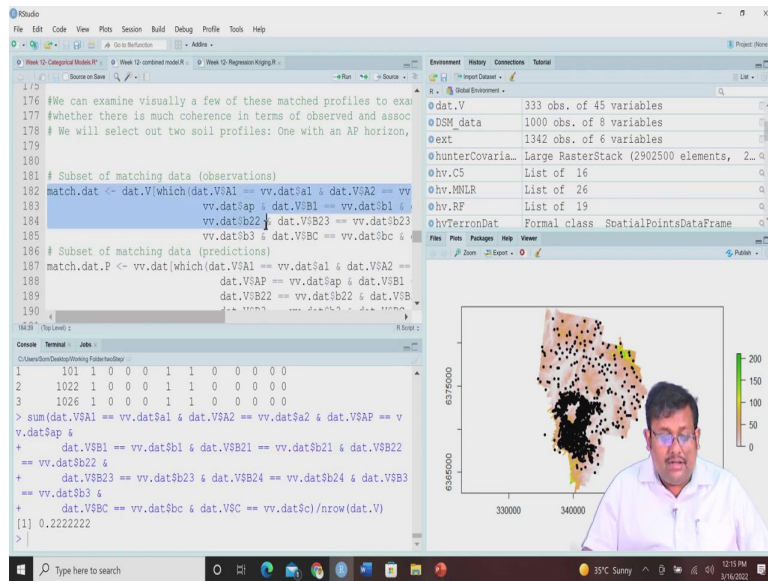
```
166 # matched soil profiles
167 sum(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.dat$a2 & dat.V$AP ==
168   dat.V$b1 == vv.dat$b1 & dat.V$b21 == vv.dat$b21 & dat.V$b
169   dat.V$b23 == vv.dat$b23 & dat.V$b24 == vv.dat$b24 & dat.V
170   dat.V$bC == vv.dat$bC & dat.V$c == vv.dat$c)/nrow(dat.V)
171
172
173
174 # just over 20% of validation soil profiles have matched sequen
175
176 #We can examine visually a few of these matched profiles to exa
177 #whether there is much coherence in terms of observed and assoc
178 # We will select out two soil profiles: One with an AP horizon,
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

The screenshot shows the RStudio interface. The script editor contains R code for matching soil profiles. The console shows the execution of the code, resulting in a matrix of 0s and 1s. The Environment pane on the right lists various objects, including a matrix of 333 observations and 45 variables. A scatter plot in the bottom right shows the spatial distribution of the matched profiles, with a color scale from 0 to 200.

```
166
167
168 == vv.dat$a2 & dat.V$AP == vv.dat$a2 &
169 b1 == vv.dat$b1 & dat.V$b21 == vv.dat$b21 &
170 b24 == vv.dat$b24 & dat.V$b3 == vv.dat$b3 &
171 == vv.dat$c/nrow(dat.V)
172
173
174 files have matched sequences of horizons
175
176 #e matched profiles to examine
177 rms of observed and associated predicted horizon depths.
178 : One with an AP horizon, and the other with an AI horizon.
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

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)



This screenshot shows the RStudio interface with the following content:

```
176 #We can examine visually a few of these matched profiles to exa
177 #whether there is much coherence in terms of observed and assoc
178 # We will select out two soil profiles: One with an AP horizon,
179
180
181 # Subset of matching data (observations)
182 match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv
183 vv.dat$a2 & dat.V$b1 == vv.dat$b1 &
184 vv.dat$b2 & dat.V$b3 == vv.dat$b3
185 vv.dat$b3 & dat.V$b3 == vv.dat$b3 &
186 # Subset of matching data (predictions)
187 match.dat.P <- vv.dat[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 ==
188 dat.V$a2 & dat.V$b1 == vv.dat$b1
189 dat.V$b2 == vv.dat$b2 & dat.V$b
190 dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 &
191
192
193
194
```

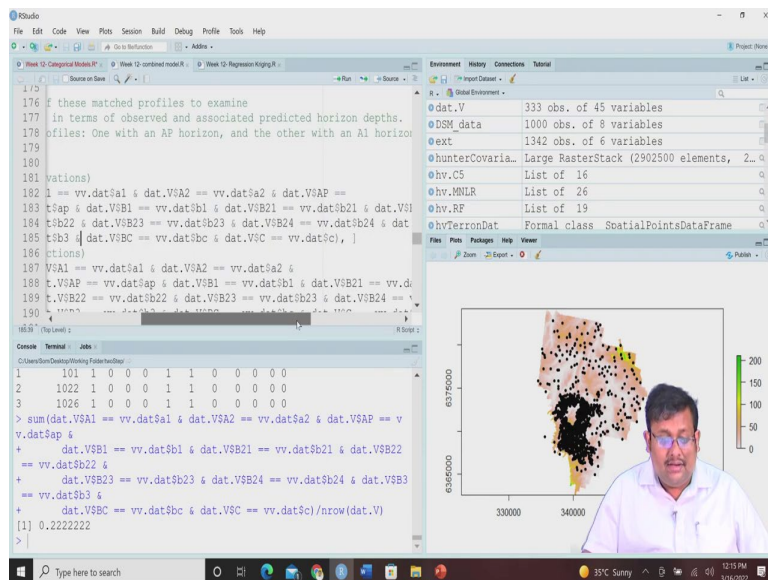
The console output shows the following:

```
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
> sum(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.dat$a2 & dat.V$a2 == v
v.dat$a2 &
+ dat.V$b1 == vv.dat$b1 & dat.V$b2 == vv.dat$b2 & dat.V$b2 ==
+ vv.dat$b2 &
+ dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 ==
+ vv.dat$b3 &
+ dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 ==
[1] 0.2222222
```

The Environment pane lists the following objects:

- @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.CS: List of 16
- @hv.MNLR: List of 26
- @hv.RF: List of 19
- @hvTerrainDat: Formal class 'SpatialPointsDataFrame'

The map shows sampling locations on a grid with coordinates ranging from 330000 to 340000 on the x-axis and 635000 to 637500 on the y-axis. A color scale on the right ranges from 0 to 200.



This screenshot shows the RStudio interface with the following content:

```
176 # these matched profiles to examine
177 # in terms of observed and associated predicted horizon depths.
178 # profiles: One with an AP horizon, and the other with an AI horizo
179
180
181 # (observations)
182 l == vv.dat$a1 & dat.V$a2 == vv.dat$a2 & dat.V$a2 == vv.dat$a2 &
183 t.V$a2 == vv.dat$a2 & dat.V$b1 == vv.dat$b1 & dat.V$b1 == vv.dat$b1 &
184 t.V$b1 == vv.dat$b1 & dat.V$b2 == vv.dat$b2 & dat.V$b2 == vv.dat$b2 &
185 t.V$b2 == vv.dat$b2 & dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 &
186 # (predictions)
187 V$A1 == vv.dat$a1 & dat.V$a2 == vv.dat$a2 &
188 t.V$a2 == vv.dat$a2 & dat.V$b1 == vv.dat$b1 & dat.V$b1 == vv.d
189 t.V$b1 == vv.dat$b1 & dat.V$b2 == vv.dat$b2 & dat.V$b2 == vv.d
190 t.V$b2 == vv.dat$b2 & dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.d
191
192
193
194
```

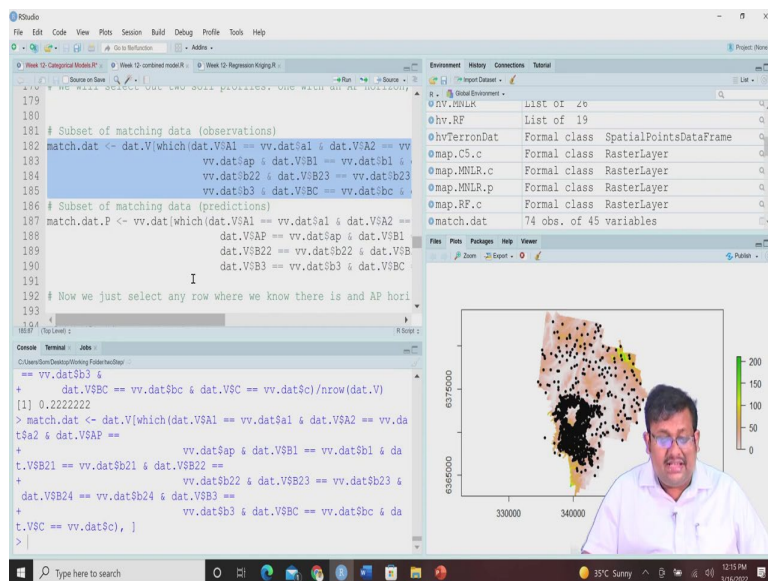
The console output shows the following:

```
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
> sum(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.dat$a2 & dat.V$a2 == v
v.dat$a2 &
+ dat.V$b1 == vv.dat$b1 & dat.V$b2 == vv.dat$b2 & dat.V$b2 ==
+ vv.dat$b2 &
+ dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 ==
+ vv.dat$b3 &
+ dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 ==
[1] 0.2222222
```

The Environment pane lists the following objects:

- @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.CS: List of 16
- @hv.MNLR: List of 26
- @hv.RF: List of 19
- @hvTerrainDat: Formal class 'SpatialPointsDataFrame'

The map shows sampling locations on a grid with coordinates ranging from 330000 to 340000 on the x-axis and 635000 to 637500 on the y-axis. A color scale on the right ranges from 0 to 200.



This screenshot shows the RStudio interface with the following content:

```
179
180
181 # Subset of matching data (observations)
182 match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv
183 vv.dat$a2 & dat.V$b1 == vv.dat$b1 &
184 vv.dat$b2 & dat.V$b3 == vv.dat$b3
185 vv.dat$b3 & dat.V$b3 == vv.dat$b3 &
186 # Subset of matching data (predictions)
187 match.dat.P <- vv.dat[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 ==
188 dat.V$a2 & dat.V$b1 == vv.dat$b1
189 dat.V$b2 == vv.dat$b2 & dat.V$b
190 dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 &
191
192
193
194
```

The console output shows the following:

```
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
> sum(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.dat$a2 & dat.V$a2 == v
v.dat$a2 &
+ dat.V$b1 == vv.dat$b1 & dat.V$b2 == vv.dat$b2 & dat.V$b2 ==
+ vv.dat$b2 &
+ dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 ==
+ vv.dat$b3 &
+ dat.V$b3 == vv.dat$b3 & dat.V$b3 == vv.dat$b3 & dat.V$b3 ==
[1] 0.2222222
```

The Environment pane lists the following objects:

- @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.CS: List of 16
- @hv.MNLR: List of 26
- @hv.RF: List of 19
- @hvTerrainDat: Formal class 'SpatialPointsDataFrame'
- @map.CS.c: Formal class 'RasterLayer'
- @map.MNLR.c: Formal class 'RasterLayer'
- @map.MNLR.p: Formal class 'RasterLayer'
- @map.RF.c: Formal class 'RasterLayer'
- @match.dat: 74 obs. of 45 variables

The map shows sampling locations on a grid with coordinates ranging from 330000 to 340000 on the x-axis and 635000 to 637500 on the y-axis. A color scale on the right ranges from 0 to 200.

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)

The screenshot shows the RStudio interface with the following R code in the editor:

```

179 # We will select out two soil profiles one with an AP horizon
180
181 # Subset of matching data (observations)
182 match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv
183   vv.dat$a2 & dat.V$b1 == vv.dat$b1 &
184   vv.dat$b2 & dat.V$b3 == vv.dat$b3 &
185   vv.dat$b3 & dat.V$bC == vv.dat$bC &
186 # Subset of matching data (predictions)
187 match.dat.P <- vv.dat[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 ==
188   dat.V$a2 & dat.V$b1 == vv.dat$b1 &
189   dat.V$b2 == vv.dat$b2 & dat.V$b3
190   dat.V$b3 == vv.dat$b3 & dat.V$bC
191
192 # Now we just select any row where we know there is an AP hori
193
194
195

```

The console shows the execution of the following code:

```

== vv.dat$b3 &
+   dat.V$bC == vv.dat$bC & dat.V$c == vv.dat$c)/nrow(dat.V)
[1] 0.2222222
> match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.da
t$a2 & dat.V$aP ==
+   vv.dat$aP & dat.V$b1 == vv.dat$b1 & da
t.V$b21 == vv.dat$b21 & dat.V$b22 ==
+   vv.dat$b22 & dat.V$b23 == vv.dat$b23 &
+   dat.V$b24 == vv.dat$b24 & dat.V$b3 ==
+   vv.dat$b3 & dat.V$bC == vv.dat$bC & da
t.V$c == vv.dat$c), ]
>

```

The environment pane on the right shows the following objects:

- Global Environment
- hv.MNLR: List of 26
- hv.RF: List of 19
- hv.TerronDat: Formal class 'SpatialPointsDataFrame'
- map.C5.c: Formal class 'RasterLayer'
- map.MNLR.c: Formal class 'RasterLayer'
- map.MNLR.p: Formal class 'RasterLayer'
- map.RF.c: Formal class 'RasterLayer'
- match.dat: 74 obs. of 45 variables

The plot pane shows a spatial plot of the data points on a map of India, with a color scale ranging from 0 to 200. A video feed of the presenter is overlaid on the bottom right of the plot.

The screenshot shows the RStudio interface with the following R code in the editor:

```

179 # We will select out two soil profiles one with an AP horizon
180
181 # Subset of matching data (observations)
182 match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv
183   vv.dat$a2 & dat.V$b1 == vv.dat$b1 &
184   vv.dat$b2 & dat.V$b3 == vv.dat$b3 &
185   vv.dat$b3 & dat.V$bC == vv.dat$bC &
186 # Subset of matching data (predictions)
187 match.dat.P <- vv.dat[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 ==
188   dat.V$a2 & dat.V$b1 == vv.dat$b1 &
189   dat.V$b2 == vv.dat$b2 & dat.V$b3 == vv.dat$b3 &
190   dat.V$b3 & dat.V$bC == vv.dat$bC &
191   dat.V$c == vv.dat$c), ]
192 # Now we just select any row where we know there is an AP hori
193
194
195

```

The console shows the execution of the following code:

```

== vv.dat$b3 &
+   dat.V$bC == vv.dat$bC & dat.V$c == vv.dat$c)/nrow(dat.V)
[1] 0.2222222
> match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$a2 == vv.da
t$a2 & dat.V$aP ==
+   vv.dat$aP & dat.V$b1 == vv.dat$b1 & da
t.V$b21 == vv.dat$b21 & dat.V$b22 ==
+   vv.dat$b22 & dat.V$b23 == vv.dat$b23 &
+   dat.V$b24 == vv.dat$b24 & dat.V$b3 ==
+   vv.dat$b3 & dat.V$bC == vv.dat$bC & da
t.V$c == vv.dat$c), ]
>

```

The environment pane on the right shows the following objects:

- Global Environment
- hv.MNLR: List of 26
- hv.RF: List of 19
- hv.TerronDat: Formal class 'SpatialPointsDataFrame'
- map.C5.c: Formal class 'RasterLayer'
- map.MNLR.c: Formal class 'RasterLayer'
- map.MNLR.p: Formal class 'RasterLayer'
- map.RF.c: Formal class 'RasterLayer'
- match.dat: 74 obs. of 45 variables

The plot pane shows a spatial plot of the data points on a map of India, with a color scale ranging from 0 to 200. A video feed of the presenter is overlaid on the bottom right of the plot.

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)

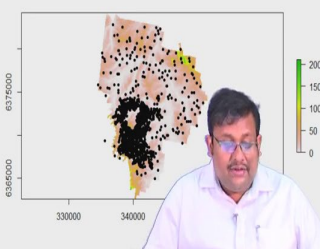
RStudio environment showing R code for data manipulation and analysis. The code includes:

```
186 # Subset of matching data (predictions)
187 match.dat.P <- vv.dat[which(dat.V$A1 == vv.dat$A1 & dat.V$A2 ==
188 dat.V$A3 & dat.V$A4 == vv.dat$A4 & dat.V$A5 ==
189 dat.V$A6 & dat.V$A7 == vv.dat$A7 & dat.V$A8 ==
190 dat.V$A9 & dat.V$A10 == vv.dat$A10 & dat.V$A11 ==
191 dat.V$A12 & dat.V$A13 == vv.dat$A13 & dat.V$A14 ==
192 dat.V$A15 & dat.V$A16 == vv.dat$A16 & dat.V$A17 ==
193 dat.V$A18 & dat.V$A19 == vv.dat$A19 & dat.V$A20 ==
194 vv.dat$A20), ] #observation
195
196
197 match.dat.P[49, ] #prediction
198
199 #We can see in these two profiles, the sequence of horizons is A
200 # Using the horizon classes together with the associated depths
201
202 H1 <- c("A1", "B1", "B2", "B3")
203
```

The console output shows:

```
> match.dat[49, ] #observation
      FID      e      n A1 A2 AP B1 B2 B3 B4 B5 BC C A1d
195 642 338096 6372259 0 0 1 0 1 1 0 0 0 0 1 0 NA
    A2d APd B1d B21d B22d B23d B24d B3d Bcd Cd ID totalCount
195 NA 10 NA 30 15 NA NA NA 45 NA 735 446.7597
      tppm Terrain_Ruggedness_Index      slope SAGA_wetness_index
195 7.192239          0.846727 0.697118          13.34301
      r57 r37 r32 PC2 PC1 ndvi
195 1.955892 0.794118 1.542857 -1.89351 -2.239939 -0.076923
      MRVBF MRRTF Mid_slope Positon light_insolation
```

The Environment pane on the right lists objects such as `@hv.TerronDat`, `@map.C5.c`, `@map.MNLR.c`, `@map.MNLR.p`, `@map.RF.c`, `@match.dat`, and `@match.dat.P`.



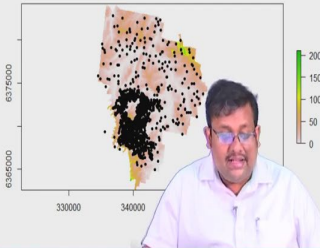
RStudio environment showing R code for data manipulation and analysis. The code includes:

```
188 dt.V$AP == vv.dat$ap & dt.V$B1
189 dt.V$B2 == vv.dat$b22 & dt.V$B
190 dt.V$B3 == vv.dat$b3 & dt.V$B
191
192 # Now we just select any row where we know there is and AP hori
193
194 match.dat[49, ] #observation
195
196
197 match.dat.P[49, ] #prediction
198
199 #We can see in these two profiles, the sequence of horizons is A
200 # Using the horizon classes together with the associated depths
201
202 H1 <- c("A1", "B21", "B22", "BC")
203
```

The console output shows:

```
> match.dat[49, ] #observation
      MRVBF MRRTF Mid_slope Positon light_insolation
195 0.111123 3.746326          0.130692          1716.388
    kperc Filled_DEM drainage_2011
195 0.5863795 142.8293 3.909594
      Altitude_Above_Channel_Network
195          25.52147
> match.dat.P[49, ] #prediction
      dt.V.FID a1 a2 ap b1 b21 b22 b23 b24 b3 bc c a1d a2d
195 642 0 0 1 0 1 0 1 1 0 0 0 0 1 0 18 16
    apd b1.1 b21d b22d b23d b24d b3d bcd cd
195 21.92308 18 31 27 20 15.41176 NA 32 NA
```

The Environment pane on the right lists objects such as `@hv.TerronDat`, `@map.C5.c`, `@map.MNLR.c`, `@map.MNLR.p`, `@map.RF.c`, `@match.dat`, and `@match.dat.P`.



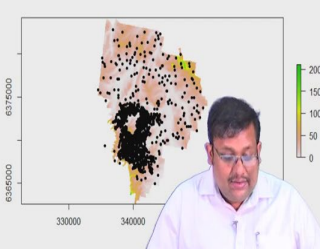
RStudio environment showing R code for data manipulation and analysis. The code includes:

```
188 dt.V$AP == vv.dat$ap & dt.V$B1
189 dt.V$B2 == vv.dat$b22 & dt.V$B
190 dt.V$B3 == vv.dat$b3 & dt.V$B
191
192 # Now we just select any row where we know there is and AP hori
193
194 match.dat[49, ] #observation
195
196
197 match.dat.P[49, ] #prediction
198
199 #We can see in these two profiles, the sequence of horizons is A
200 # Using the horizon classes together with the associated depths
201
202 H1 <- c("A1", "B21", "B22", "BC")
203
```

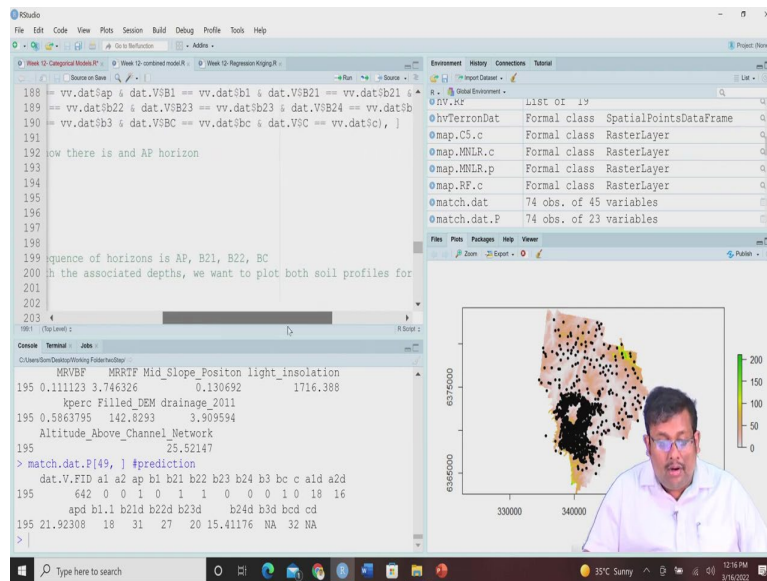
The console output shows:

```
> match.dat[49, ] #observation
      FID      e      n A1 A2 AP B1 B2 B3 B4 B5 BC C A1d
195 642 338096 6372259 0 0 1 0 1 1 0 0 0 0 1 0 NA
    A2d APd B1d B21d B22d B23d B24d B3d Bcd Cd ID totalCount
195 NA 10 NA 30 15 NA NA NA 45 NA 735 446.7597
      tppm Terrain_Ruggedness_Index      slope SAGA_wetness_index
195 7.192239          0.846727 0.697118          13.34301
      r57 r37 r32 PC2 PC1 ndvi
195 1.955892 0.794118 1.542857 -1.89351 -2.239939 -0.076923
      MRVBF MRRTF Mid_slope Positon light_insolation
195 0.111123 3.746326          0.130692          1716.388
    kperc Filled_DEM drainage_2011
```

The Environment pane on the right lists objects such as `@hv.TerronDat`, `@map.C5.c`, `@map.MNLR.c`, `@map.MNLR.p`, `@map.RF.c`, `@match.dat`, and `@match.dat.P`.



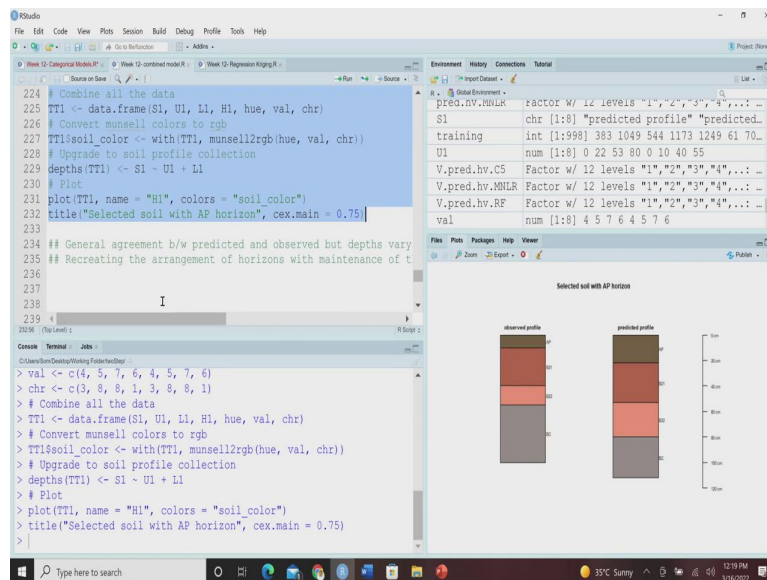
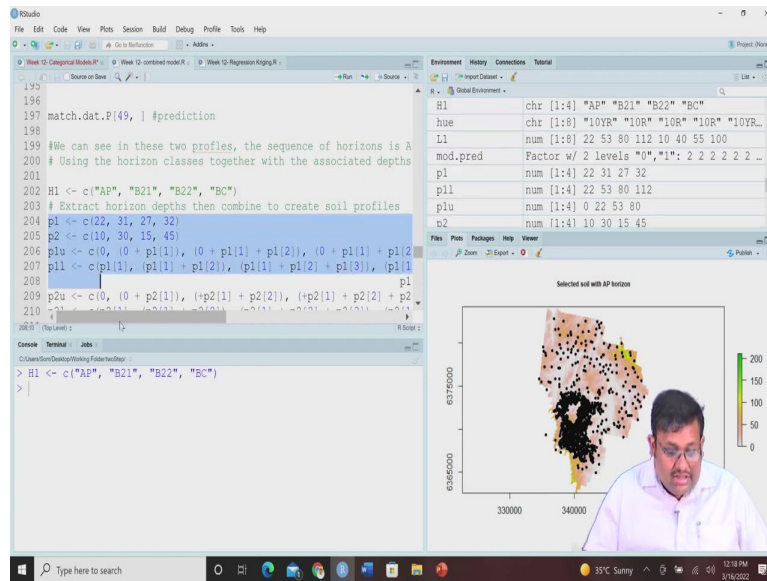




So, we know that on 1549 observation there will be an AP horizon and then let us see, so this is the observation. So, in the observation you can see here, these are observed data, so, there is an AP horizon as we denoted by 1 and if we want to see the prediction, so, here you can see, they are two, one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

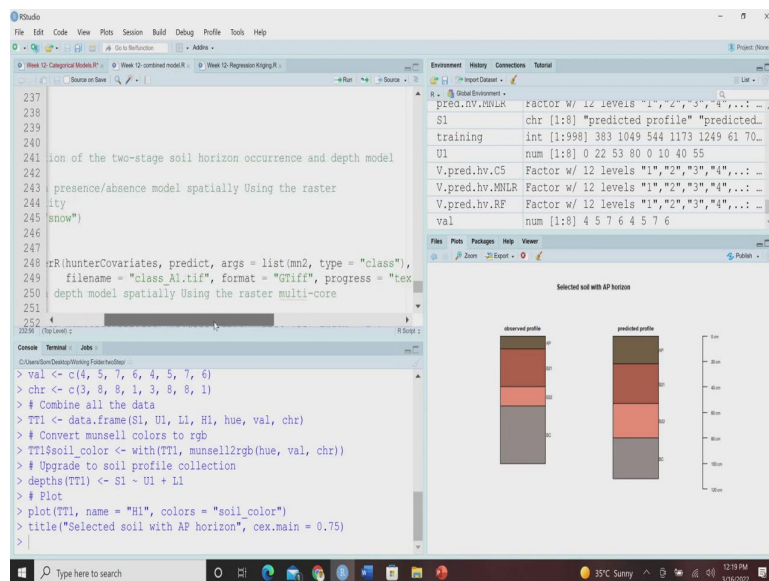
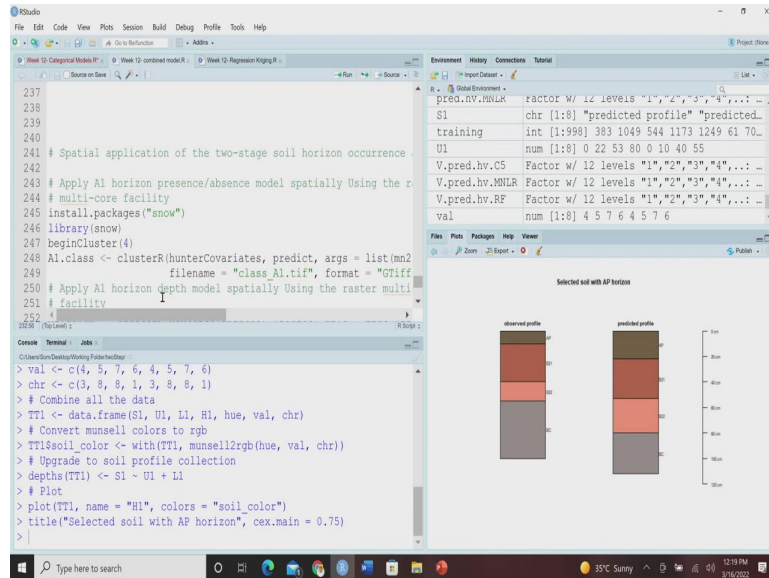


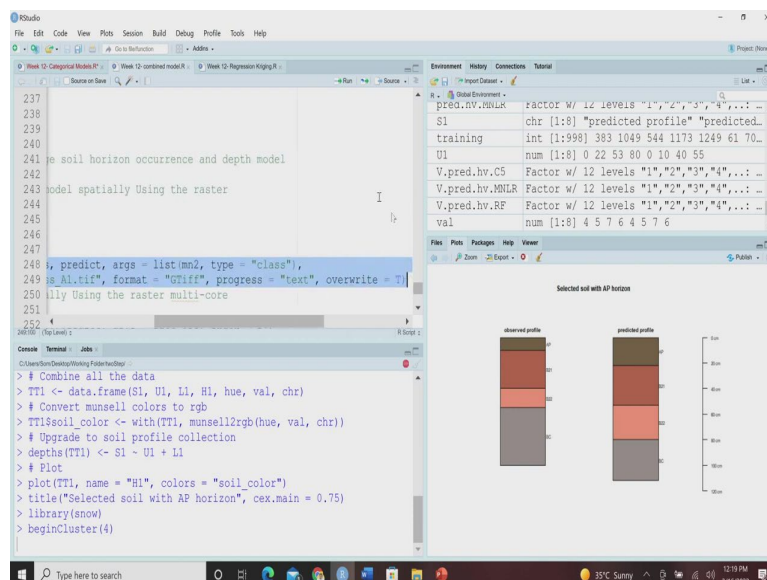
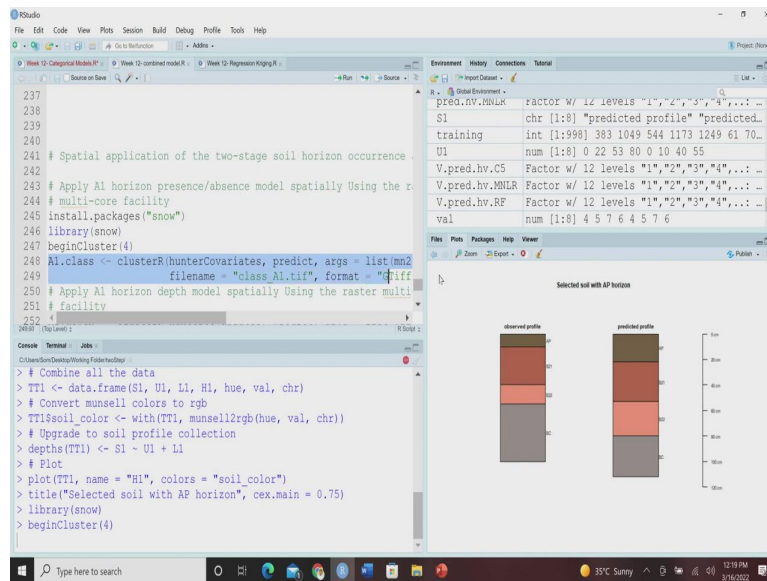
Next is we want to extract these horizon depths and then we combine to create the soil profile will produce the plots for a side by side comparison. So, let us run these codes and see how this has looked like. So, if we do this, they will produce these this is an observed profile and you can see this 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 predicted 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)

RStudio interface showing R code for soil classification. The code includes loading the 'snow' library, clustering soil data, and writing the results to TIF files. The console shows progress bars for writing 'class\_A1.tif' (100%) and 'depth\_A1.tif' (12%). A warning message is displayed: 'Warning message: In .gd\_setProject(object, ...) : NOT UPDATED FOR PROJ >= 6'. The Environment pane shows objects like 'AI.class', 'con.mat', 'odat', 'odat.C', 'odat.V', 'DSM\_data', and 'oext'. A plot titled 'Selected soil with AP horizon' shows two soil profiles: 'observed profile' and 'predicted profile'.

RStudio interface showing R code for soil classification. The code includes loading the 'snow' library, clustering soil data, and writing the results to TIF files. The console shows progress bars for writing 'class\_A1.tif' (100%) and 'depth\_A1.tif' (100%). A warning message is displayed: 'Warning message: In .gd\_setProject(object, ...) : NOT UPDATED FOR PROJ >= 6'. The Environment pane shows objects like 'AI.class', 'con.mat', 'odat', 'odat.C', 'odat.V', 'DSM\_data', and 'oext'. A plot titled 'Selected soil with AP horizon' shows two soil profiles: 'observed profile' and 'predicted profile'.

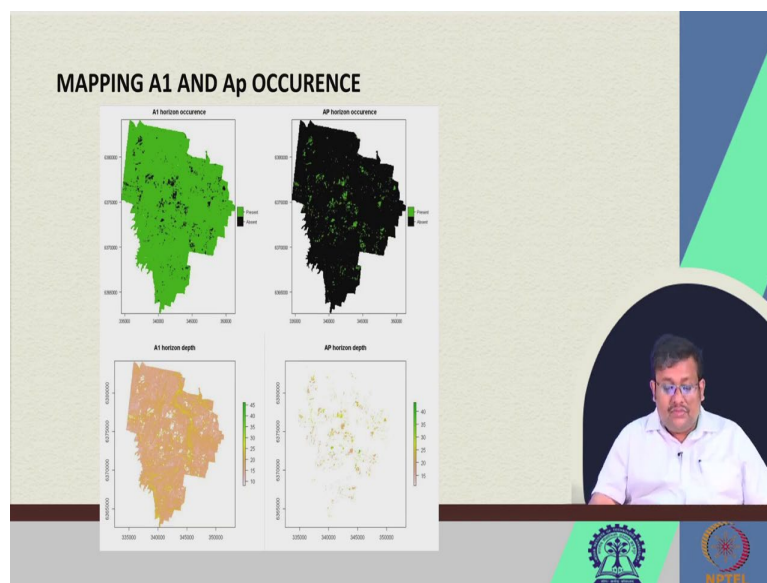
Windows File Explorer showing a directory listing of files. The files are listed in a table with columns for Name, Date modified, Type, and Size. The files include 'RData', 'Rhistory', 'class\_A1', 'depth\_A1', 'depth\_A1\_mask', 'MY\_horizons.rda', 'validation\_obs', and 'validation\_outs'. The 'depth\_A1\_mask' file is highlighted.

Name	Date modified	Type	Size
RData	4/7/2020 6:11 PM	R Workspace	12,462 KB
Rhistory	3/13/2022 10:50 PM	R HISTORY File	1 KB
class_A1	3/15/2022 10:10 PM	TIF File	115 KB
depth_A1	3/15/2022 10:10 PM	TIF File	355 KB
depth_A1_mask	3/15/2022 10:10 PM	TIF File	355 KB
MY_horizons.rda	11/24/2017 9:17 PM	RDA File	32 KB
validation_obs	11/24/2017 9:17 PM	Text Document	100 KB
validation_outs	11/24/2017 9:17 PM	Text Document	26 KB

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.

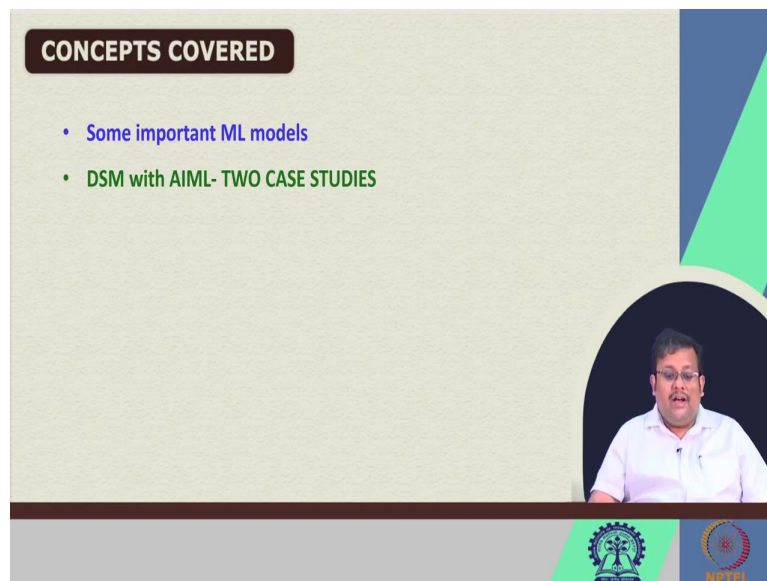
(Refer Slide Time: 10:00)



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)



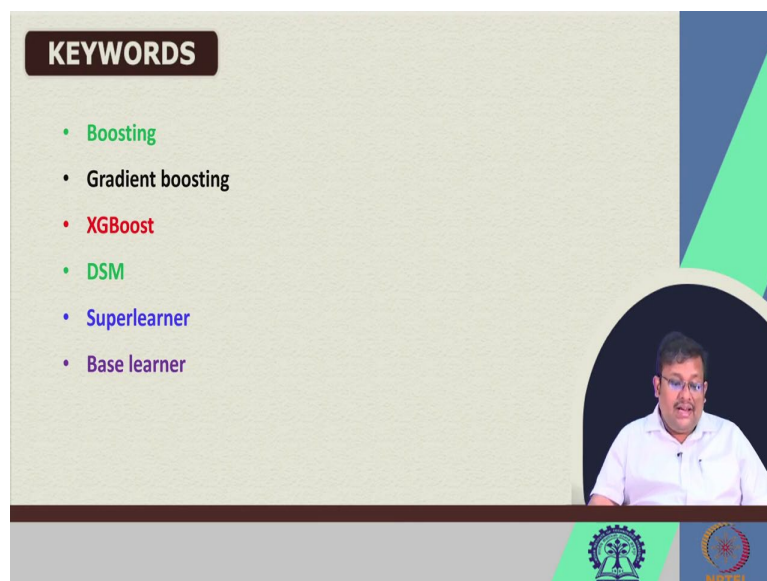
**CONCEPTS COVERED**

- Some important ML models
- DSM with AIML- TWO CASE STUDIES

The slide features a video inset of a man in a white shirt speaking. At the bottom, there are logos for IIT Bombay and NPTEL.

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)



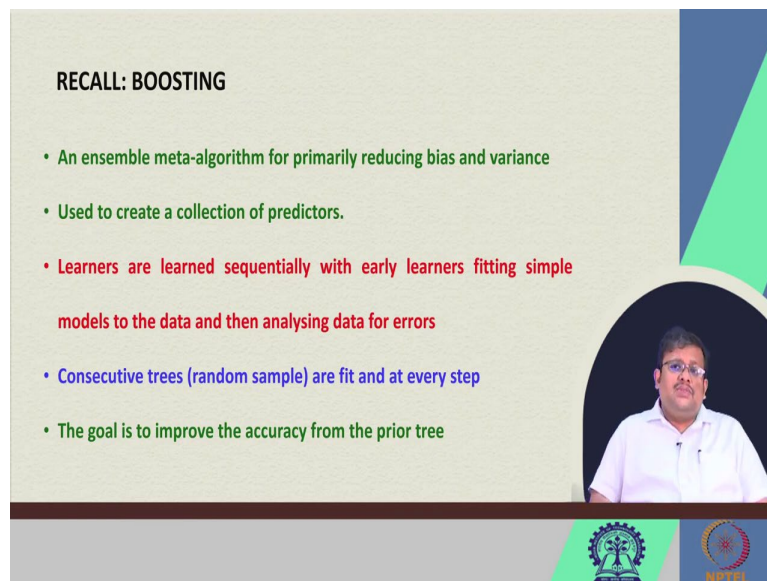
**KEYWORDS**

- Boosting
- Gradient boosting
- XGBoost
- DSM
- Superlearner
- Base learner

The slide features a video inset of the same man in a white shirt speaking. At the bottom, there are logos for IIT Bombay and NPTEL.

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)



**RECALL: BOOSTING**

- An ensemble meta-algorithm for primarily reducing bias and variance
- Used to create a collection of predictors.
- Learners are learned sequentially with early learners fitting simple models to the data and then analysing data for errors
- Consecutive trees (random sample) are fit and at every step
- The goal is to improve the accuracy from the prior tree

The slide features a video inset of a man in a white shirt speaking. The background is light green with a blue and green geometric design on the right. Logos for institutions are visible at the bottom.

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)

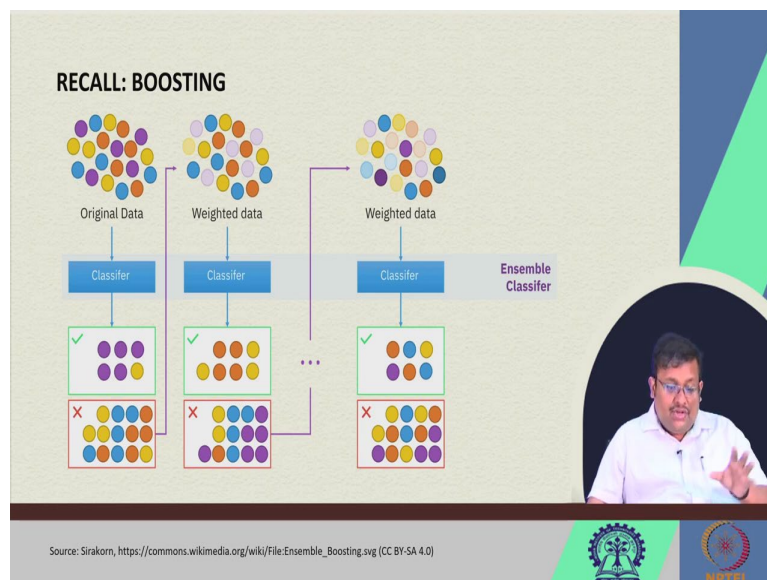
**RECALL: BOOSTING**

- When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly
- This process converts weak learners into better performing model



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)



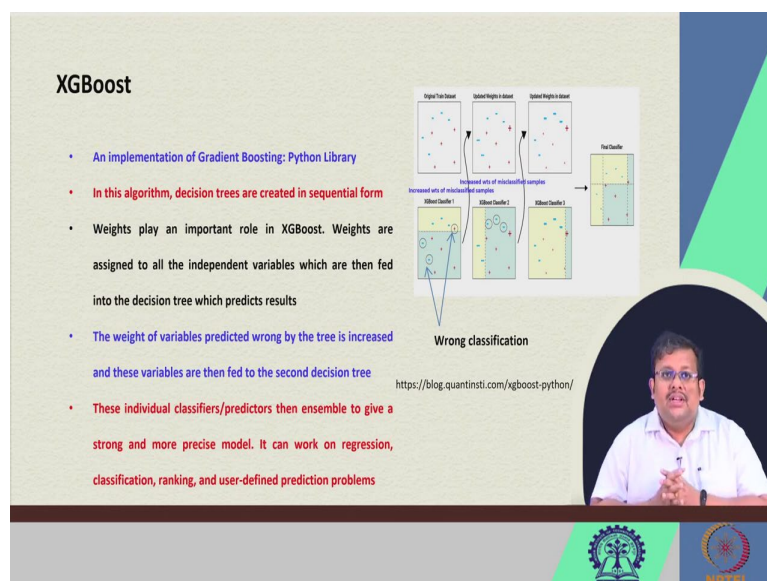
**GRADIENT BOOSTING**

- A popular boosting algorithm
- In gradient boosting, each predictor corrects its predecessor's error
- Each predictor is trained using the residual errors of predecessor as labels

The slide features a speaker in a video inset at the bottom right. The background is a light green wall with a blue and green geometric design on the right side. Logos for IIT Bombay and NPTEL are visible at the bottom.

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)



**XGBoost**

- An implementation of Gradient Boosting: Python Library
- In this algorithm, decision trees are created in sequential form
- Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results
- The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree
- These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems

The slide includes a diagram illustrating the sequential nature of XGBoost. It shows three decision trees: 'Original Tree', 'Original Tree + 1st Stage', and 'Original Tree + 2nd Stage'. The diagram highlights 'Increased sets of misclassified samples' and 'Increased sets of misclassified samples' leading to 'Wrong classification'. A URL <https://blog.quantinsti.com/xgboost-python/> is provided. A speaker is visible in a video inset at the bottom right. The background is a light green wall with a blue and green geometric design on the right side. Logos for IIT Bombay and NPTEL are visible at the bottom.

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)

### DSM: APPLICATION OF AIML

Table 1  
Non-exhaustive list with numbers of case studies in which machine learning algorithms are used for digital soil mapping.

Spatial extent	Sample size	Sampling design	Number of covariates	Machine learning model	Covariate selection	Resampling strategy	Map quality indicator	Uncertainty quantification	Reference
Quantitative maps									
Field	350	grid-based	19	cubist, RF	no	no	R <sup>2</sup> , RMSE	no	Dubois et al. (2019)
Local	47	stratified random	41	RF	yes	yes	RMSE, COR	yes	Blanco et al. (2019)
Local	71	GIS	13	cubist	no	no	RMSE, RMSE, R, CCC	yes	Leone et al. (2016)
Local	75	grid-based	9	ANN	no	no	R <sup>2</sup> , RMSE	no	Kachhikera et al. (2018)
Local	98	targeted	173	RF	yes	no	RMSE, R <sup>2</sup>	no	Shi et al. (2018)
Local	135	single-random	20	RF	no	no	R <sup>2</sup> , RMSE, CCC	no	Chen et al. (2017)
Local	137	not specified	133	GBM	yes	yes	R <sup>2</sup> , RMSE, MAE	yes	Chen et al. (2018)
Local	137	not specified	402	cubist	yes	no	RMSE, R <sup>2</sup> , R <sub>adj</sub>	no	Miller et al. (2018)
Local	138	stratified random	25	ANN, SRT	no	no	RMSE, R <sup>2</sup> , RMSE	no	Wang et al. (2018)
Local	137	systematic random	25	ANN, GBM	yes	yes	RMSE, R <sup>2</sup> , RMSE	no	Wahedi et al. (2019)
Local	138	not specified	15	RF	yes	no	RMSE, R <sup>2</sup> , CCC	no	Shi et al. (2019)
Local	138	grid-based	not specified	ANN	no	yes	validation coefficient, R <sup>2</sup> , RMSE	no	Shayegh et al. (2019)
Local	18	not specified	not specified	not specified	no	no	Williams' index of agreement, EPQ	no	Alamirani et al. (2019)
Local	132	grid-based	26	RF	yes	no	RMSE, R <sup>2</sup>	no	Tahiri et al. (2019)
Local	139/34	not specified	37	RF, cubist, CNN, NN, enbNet, chm, vme, GCM, SVM, RF, SVM	yes	no	R <sup>2</sup> , RMSE, MAE, RMSE	yes	Kudrycki et al. (2018)
Local	305	stratified random	18	RF	no	yes	RMSE, RMSE	no	Chen et al. (2016)
Local	575 profiles	GIS	19	RF	no	no	RMSE, RMSE, R <sup>2</sup>	no	Toussaint-Balajout et al. (2018)
Local	598 profiles	GIS	16	ANN, SVM, SVM, RF, RF	no	yes	RMSE, CCC	no	Toussaint-Balajout et al. (2018)
Local	226	not specified	409	cubist	yes	no	RMSE, R <sup>2</sup>	yes	Miller et al. (2016)
Local	339 profiles	not specified	12	RF, ANN, boost square	no	yes	R <sup>2</sup> , R <sub>adj</sub> , RMSE, relative RMSE	no	Chen et al. (2017)
Local	339	single-random	10	SVM	no	yes	RMSE	no	Toussaint et al. (2019)
Local	339	GIS	16	RF, GBM	no	yes	RMSE, MAE, RMSE, R <sup>2</sup>	no	Toussaint et al. (2019)
Local	342/32	-	14	cubist, RF, RF	yes	no	R <sup>2</sup> , RMSE	no	Blanco et al. (2018)
Local	342/32	-	14	RF	no	yes	R <sup>2</sup>	no	Blanco et al. (2018)
Local	399	not specified	12	SVM	no	no	R <sup>2</sup> , RMSE	no	de Silva Chapin et al. (2018)
Local	440	targeted	19	RF, SVM, ANN	no	yes	RMSE, MAE	no	Nien et al. (2015)
Local	460	grid-based	21	RF	no	yes	RMSE, RMSE	no	Pedroni et al. (2017)
Local	568	single-random	35	GBM	no	no	R <sup>2</sup> , RMSE, range-normalized RMSE	yes	Alamirani et al. (2019)
Local	1104	targeted	20	RF, SVM, SVM	no	yes	RMSE, MAE	no	Fontaine et al. (2017)
Local	2002	200	300-500	RF, RF	yes	yes	RMSE, RMSE, R <sup>2</sup>	no	Fontaine et al. (2018)
Local	2396	targeted	3	CNN, RF	no	yes	RMSE, RMSE, R <sup>2</sup> , CCC	no	Muller et al. (2019)
Regional	not specified	not specified	20	cubist	no	no	R <sup>2</sup> , RMSE, RMSE, CCC	yes	Dubois et al. (2019)
Regional	120 profiles	probabilistic	11	RF, RF	no	no	RMSE, RMSE, R <sup>2</sup> , CCC	yes	Tang et al. (2016)
Regional	240	grid-based	4	ANN	no	no	RMSE, MAE, RMSE, CCC	no	Yang et al. (2016)
Regional	339/362	targeted	40	GBM	no	no	R <sup>2</sup> , RMSE	yes	Fontaine and Desjardins (2019)
Regional	402 profiles	not specified	1	CNN	no	yes	R <sup>2</sup> , RMSE	no	Fontaine et al. (2019)
Regional	508	not specified	13	RF, RF	yes	no	R <sup>2</sup> , RMSE	no	Regier et al. (2017)

Continued on next page

Wadoux et al. (2020)

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)

### DSM: APPLICATION OF AIML

Table 1 (continued)

Study/area	Sample size	Sampling design	Number of attributes	Machine learning model	Criteria selection	Parameter tuning	Map quality indicator	Uncertainty quantification	Reference
Regional	528	ad hoc from a systematic grid	18	RNN	yes	no	RMSLE, R <sup>2</sup> , bias, coefficient of variance	no	Maney et al. (2014)
Regional	795	simple random	16	RF, BRT, SVM	no	yes	R <sup>2</sup> , MAE, RMSE, CCC	yes	Yang et al. (2018)
Regional	979 profiles	not specified	24	RF	no	no	R <sup>2</sup> , MAE, RMSE, CCC	no	Alpar et al. (2014)
Regional	1,014	stratified random	32	CART, BRT, RF, RF, SVM	yes	no	R <sup>2</sup> , ERMS, RM, RMQ	no	Kutlu et al. (2019)
Regional	1,136	not specified	81	SVM	no	no	R <sup>2</sup> , MAE, RMSE	no	Adhikari and Chell (2016)
Regional	1,366	not specified	6	RF	no	no	CCC, RMSE	yes	Mohamed et al. (2018)
Regional	1,524	not specified	40	SVM	no	yes	R <sup>2</sup> , MSE	no	Wu et al. (2016)
Regional	2,024	topsoil data	16	QRF	no	no	ME, RMSE, R <sup>2</sup> , accuracy plot	yes	Vijayar and Lakshmin (2017)
Regional	2,024	topsoil data	16	QRF	no	yes	MSE, R <sup>2</sup>	no	Vijayar and Lakshmin (2015)
Regional	2,042	two-stage systematic	37	QRF, RF	no	yes	ME, RMSE, R <sup>2</sup> , CCC	yes	Wahdan (2019)
Regional	4,639	not specified	26	QRF	no	no	MAE, RMSE, accuracy plot	yes	Samuel et al. (2018)
Regional	4,839	not specified	32	QRF	no	no	MAE, RMSE, accuracy plot	yes	Samuel and Pinar (2019)
Regional	5,266	stratified random	6	CNN, SVM	no	no	R <sup>2</sup> , MSE, CCC	no	Shrivastava et al. (2019)
Regional	13,000	not specified	18	RF	no	no	R <sup>2</sup>	yes	Koch et al. (2019)
Regional	13,761	two-stage systematic	19	RF	no	no	MAE	no	Mohamed et al. (2018)
Regional	22,491	topsoil soil data	74	RF, CNN, SVM	yes	yes	R <sup>2</sup> , RMSE, MAE	yes	Gomez et al. (2019)
Regional	22,812,242	stratified random	34	CNN	no	yes	CCC, RMSE, MSE, MAE	yes	Vijayaraj et al. (2020)
Global	36,624	stratified random	> 200	RF, SVM	no	yes	R <sup>2</sup> , MAE, RMSE	yes	Karichanev et al. (2018)
Global	11,268	topsoil soil data	118	SVM, boost, weighted SVM, RF	yes	no	RMSE, R <sup>2</sup>	yes	Gomez et al. (2016)
Global	180,000	topsoil soil data	> 200	RF, SVM	no	yes	R <sup>2</sup>	no	Wang et al. (2017)
Categorical maps	not specified	not specified	128	SVM	no	no	Accuracy, recall, precision	no	Samuel et al. (2018)
Local	31 profiles	not specified	16	RF, SVM	no	no	not specified	no	Maney et al. (2016)
Local	160,207 (17 profiles)	GIS	139	RNN, SVM, CC, BRT, RF	yes	yes	RMSE, accuracy, bias, error, visual inspection, confusion table	yes	Wang et al. (2015)
Local	121 profiles	GIS	17	SVM, NN, SVM	no	no	map quality, Cohen's kappa, Bhattacharyya index, relative parity, relative diversity	no	Jainzadeh et al. (2017)
Local	151	not specified	not specified	SVM	no	no	RMSE, error averaged (1) accuracy, kappa coefficient	no	Karichanev et al. (2018)
Local	175, 41 profiles	stratified random	27	RNN, SVM, RF, RF	no	no	GA, GA, GA, kappa coefficient, RMSE	no	Mohamed and Van Halbeek (2017)
Local	452 profiles	regular grid	6	SVM, RF	yes	no	GA, GA, GA, kappa coefficient of agreement	no	Barbieri et al. (2019)
Local	971	grid based	33	RF	yes	no	RMSE index	no	Hindiyeh et al. (2018)
Local	1123	topographic, soil data, soil weighted, and soil weighted with random noise sampling	33	CART, CART with bagging, RF, ANN, NN, CNN, LSTM, SVM	no	yes	overall agreement, quantity disagreement, allocation disagreement, soil disagreement	no	Thapa et al. (2018)

(continued on next page)

Wadoux et al. (2020)

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)

### DSM WITH SUPERLEARNER- A CASE STUDY

- 12 base learners were used to create a superlearner

Model	Description	Hyperparameters
RF	Random forest	none
LR	Linear regression	none
LR2D	The least squares shrinkage and selection operator	lambda
LR3D	The adaptive elastic net regression operator	lambda, alpha
SVM	Support vector machine	C, gamma
QRF	Quantile regression	lambda
GP	Gaussian process	kernel, sigma
ANN	Artificial neural network	number, size of the hidden layer, learning rate, momentum, weight
ALM	Adaptive learning machine	learning rate
GBM	Gradient boosting machine	learning rate, number of trees
ET	Extreme tree	learning rate
ET2	Extreme tree with ensemble	learning rate, number of trees
ET3	Extreme tree with ensemble and bagging	learning rate, number of trees, bagging
ET4	Extreme tree with ensemble and bagging and boosting	learning rate, number of trees, bagging, boosting
Super learner	Super learner	learning rate, number of trees, bagging, boosting, ensemble

Fig. 2. Four examples of environmental variables: (A) elevation derived from digital elevation model, (B) normalized difference vegetation index (NDVI) calculated from Landsat 4 images, (C) depth to the groundwater calculated by regression kriging model, and (D) geomorphic map based on the binary classification system.

Taghizadeh-Mehrjardi et al. (2021)

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)

**DSM WITH SUPERLEARNER- A CASE STUDY**

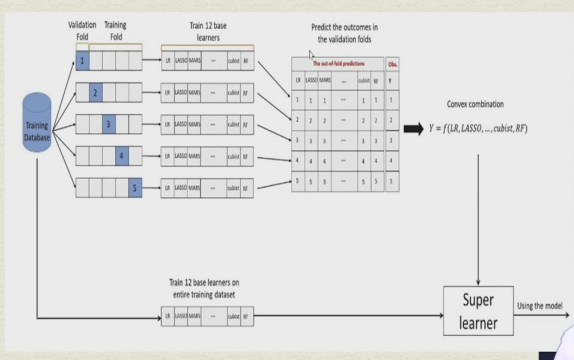
**SuperLearner** is an algorithm that uses cross-validation to estimate the performance of multiple machine learning models, or the same model with different settings. It then creates an optimal weighted average of those models, aka an "ensemble", using the test data performance





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)

**DSM WITH SUPERLEARNER- A CASE STUDY**



Taghizadeh-Mehrjardi et al. (2021)



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.

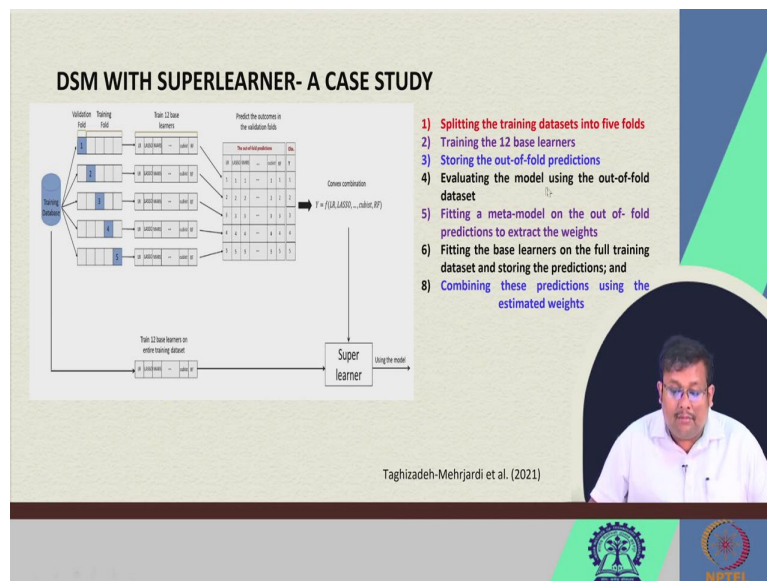
So, and although also using these training set, they have training a trained all these 12 base learners starting from linear regression Lasso, Mars 2, Cubist Random Forest. So, for each of this step, they kept out one fold and they trained these 12 models using the rest of the training folds and after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



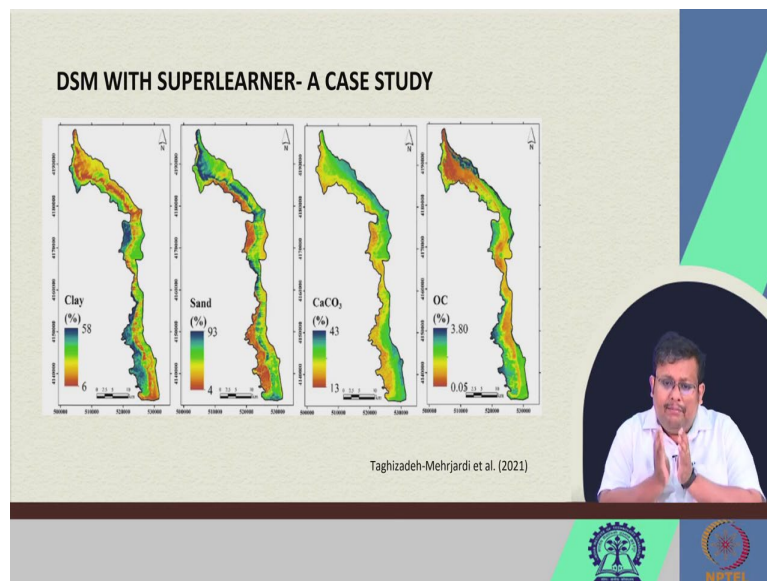
So, these are the steps. First of all splitting the training data set into 5 folds, we have seen that, and then training these 12 base learners, and then storing the out of fold predictions just like here, we are storing these out of fold predictions. And then in the fourth step, we are evaluating the 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)

**DSM WITH RK- A CASE STUDY**

*Terrain attributes*

- Slope
- Altitude above channel
- Hillshade
- Aspect
- Profile curvature
- Plan curvature
- Terrain Ruggedness Index
- Topographic Wetness Index
- Elevation

*Bioclimatic variables*

- Annual mean temperature
- Annual precipitation
- Temperature seasonality
- Rainfall seasonality
- Mean diurnal range of temperature
- Annual range of temperature
- Rainfall of wettest quarter

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.