

Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology Kharagpur
Lecture 59

Digital Soil Mapping with Categorical Variables (Contd.)

Welcome friends to this 59th lecture of NPTEL online certification course of Machine Learning for Soil and Crop Management. In this week we are talking about Digital Soil Mapping with the Categorical Variables. In our previous lectures, we have already discussed about different types of categorical models like multinomial logistic model, then C5 decision tree and we have seen what are the important steps of developing the random forest.

We have seen how these bootstrap samples are developed for creating a tree in the random forest and what is M try and what are the features of this random forest model and then how you know out of bag sample is being used for validating the model performance we have already seen.

Now, in this lecture first we are going to run the script, R script of random forest and then we are going to discuss another very important topic that is combining the regression as well as classification model. So, that is called combined model. Now, let us first see how to execute the random forest codes and then we will go for the combined model discussion.

(Refer Slide Time: 01:57)



So, these are the concepts which we are going to cover in this lecture apart from running the script of the random forest regression model categorical model, we are also going to discuss the combined continuous and categorical model.

(Refer Slide Time: 02:12)

KEYWORDS

- Combined model
- Soil horizon
- Ap horizon
- Quantile regression forest
- nnet

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, in this lecture these are the keywords like Combined model, Soil Horizon, AP horizon, Quantile regression forest, nnet these are the 5 keywords which we are going to discuss in this lecture.

(Refer Slide Time: 02:25)

```
166
167
168-##### Random Forest #####
169 library(randomForest)
170 hv.rf <- randomForest(terror ~ AACN + Drainage.Index + Light.In
171                      Gamma.Total.Count, data = DSM_data[tra
172 # Output random forest model diagnostics
173 print(hv.rf)
174 # output relative importance of each covariate
175 importance(hv.rf)
176 # Prediction of classes
177 predict(hv.rf, type = "response", newdata = DSM_data[training,
178 #calibration
179 C.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[training, ])
181
```

The screenshot shows the RStudio interface with a script editor on the left containing R code for training a random forest model. The Environment pane on the right shows objects like 'con.mat', 'DSM_data', and 'hv.rf'. The console at the bottom shows the execution of the code, including the output of 'library(randomForest)' and 'randomForest 4.6-14'.

RStudio interface showing the following code in the editor:

```

166
167
168 #####
169
170 AACN + Drainage.Index + Light.Insolation + TWI +
171 total.Count, data = DSM_data(training, ), ntree = 500, mtry = 5)
172 diagnostics
173
174 # Output random forest model diagnostics
175 # Output relative importance of each covariate
176 importance(hv.RF)
177 # Prediction of classes
178 predict(hv.RF, type = "response", newdata = DSM_data(training,
179
180 #calibration
181 C.pred.hv.RF <- predict(hv.RF, newdata = DSM_data(training, ))
182
183 ##### Random Forest #####
184
185 library(randomForest)
186 randomForest(4.6-14
187 Type rfNews() to see new features/changes/bug fixes.
188
189

```

The Environment pane on the right shows the following objects:

- con.mat: num [1:4, 1:4] 5 0 1 2 0 15 0 5 0 1 ...
- DSM_data: 1000 obs. of 8 variables
- hunterCovaria...: Large RasterStack (2902500 elements, 2...
- hv.C5: List of 16
- hv.MNLR: List of 26
- hv.RF: List of 19
- hvTerrorDat: Formal class 'SpatialPointsDataFrame'

The Console shows the output of the library() function:

```

> ##### Random Forest #####
> library(randomForest)
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
>

```

RStudio interface showing the execution of the randomForest function. The code in the editor is:

```

166
167
168 ##### Random Forest #####
169 library(randomForest)
170 hv.RF <- randomForest(terror ~ AACN + Drainage.Index + Light.In
171 Gamma.Total.Count, data = DSM_data(trai
172 # Output random forest model diagnostics
173 print(hv.RF)
174 # Output relative importance of each covariate
175 importance(hv.RF)
176 # Prediction of classes
177 predict(hv.RF, type = "response", newdata = DSM_data(training,
178
179 #calibration
180 C.pred.hv.RF <- predict(hv.RF, newdata = DSM_data(training, ))
181
182 ##### Random Forest #####
183
184 library(randomForest)
185 randomForest(4.6-14
186 Type rfNews() to see new features/changes/bug fixes.
187
188

```

The Console displays the output of the randomForest function:

```

Call: randomForest(x = DSM_data(training, ), y = terror, ntree = 500, mtry = 5,
                    importance = TRUE, random.seed = 1000)
Random Forest model
Number of trees: 500
Node size: 5
mtry: 5
Maximum depth of trees: 12
Cost function: cross-entropy
OOB estimate of error rate: 49.29%
Confusion matrix:
 1  2  3  4  5  6  7  8  9 10 11 12 class.error
1  9  2  1  7  0  0  0  0  0  0  0  0  0.5263158
2  4  2  3  0  0  0  0  0  0  0  0  0  0.7777778
3  0  3  39  5  0  0  9  0  1  0  3  5  0.4000000
4  6  0  4  25  0  0  6  15  1  6  1  0.5454545
5  0  0  1  0  50 13  6  9  3  7  5  0  0.4680851
6  0  0  0  1  8  50  0  6  0  1  0  0  0.2424242
7  0  0  15  3  5  0  79 11  1  6  2  2  0.3629032
8  0  0  1  2  8  2  15  64  3  4  1  1  0.3786408
9  0  1  2  6  8  1  7  6  8  0  3  0  0.8095238
10 0  0  1  0 13  2 10  6  0 17  1  3  0.6792453
11 0  0 10  6  3  1  5  6  1  1  3  2  0.9210526
12 0  0  3  0  0  0 14  0  0  5  1  9  0.7187500

```

RStudio interface showing the confusion matrix and OOB estimate of error rate. The code in the editor is:

```

166
167
168 ##### Random Forest #####
169 library(randomForest)
170 hv.RF <- randomForest(terror ~ AACN + Drainage.Index + Light.In
171 Gamma.Total.Count, data = DSM_data(trai
172 # Output random forest model diagnostics
173 print(hv.RF)
174 # Output relative importance of each covariate
175 importance(hv.RF)
176 # Prediction of classes
177 predict(hv.RF, type = "response", newdata = DSM_data(training,
178
179 #calibration
180 C.pred.hv.RF <- predict(hv.RF, newdata = DSM_data(training, ))
181
182 ##### Random Forest #####
183
184 library(randomForest)
185 randomForest(4.6-14
186 Type rfNews() to see new features/changes/bug fixes.
187
188

```

The Console displays the output of the randomForest function:

```

Call: randomForest(x = DSM_data(training, ), y = terror, ntree = 500, mtry = 5,
                    importance = TRUE, random.seed = 1000)
Random Forest model
Number of trees: 500
Node size: 5
mtry: 5
Maximum depth of trees: 12
Cost function: cross-entropy
OOB estimate of error rate: 49.29%
Confusion matrix:
 1  2  3  4  5  6  7  8  9 10 11 12 class.error
1  9  2  1  7  0  0  0  0  0  0  0  0  0.5263158
2  4  2  3  0  0  0  0  0  0  0  0  0  0.7777778
3  0  3  39  5  0  0  9  0  1  0  3  5  0.4000000
4  6  0  4  25  0  0  6  15  1  6  1  0.5454545
5  0  0  1  0  50 13  6  9  3  7  5  0  0.4680851
6  0  0  0  1  8  50  0  6  0  1  0  0  0.2424242
7  0  0  15  3  5  0  79 11  1  6  2  2  0.3629032
8  0  0  1  2  8  2  15  64  3  4  1  1  0.3786408
9  0  1  2  6  8  1  7  6  8  0  3  0  0.8095238
10 0  0  1  0 13  2 10  6  0 17  1  3  0.6792453
11 0  0 10  6  3  1  5  6  1  1  3  2  0.9210526
12 0  0  3  0  0  0 14  0  0  5  1  9  0.7187500

```

```

166
167
168 ##### Random Forest #####
169 library(randomForest)
170 hv.rf <- randomForest(terror ~ AACN + Drainage.Index + Light.In
171                       Gamma.Total.Count, data = DSM_data[tra
172 # Output random forest model diagnostics
173 print(hv.rf)
174 # output relative importance of each covariate
175 importance(hv.rf)
176 # Prediction of classes
177 predict(hv.rf, type = "response", newdata = DSM_data[training,
178
179 #calibration
180 C.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[training, ])
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

```

Console Terminal Jobs

```

C:\Users\DSM\Desktop\DSM\Categorical model
10 0 0 1 0 13 2 10 6 0 17 1 3 0.6792453
11 0 0 10 6 3 1 5 6 1 1 3 2 0.9210526
12 0 0 3 0 0 0 14 0 0 5 1 9 0.7187500
> # output relative importance of each covariate
> importance(hv.rf)
      MeanDecreaseGini
AACN                127.21252
Drainage.Index       118.33232
Light.Insolation     86.54248
TWI                  182.35299
Gamma.Total.Count    107.73101

```

For executing the random forest classification model. Again, you have to call the library random forest, we have already installed this random forest previously, so I am just going to, call this library random forest and as you know our target is terror. And then we are using these covariates like Altitude Above Channel Network, Drainage Index, Light Insulation, TWI, Gamma Total Count all these for predicting the terror class and our data is of course, the training data. And here you can see we are specifying the ntree that is we are going to build 500 number of trees with a number of mtry equal to 5.

So, in each tree random tree, we are going to use 5 randomly selected features for splitting criteria for identifying the splitting criteria. So, here you can see the mtry is 5 and ntree is 500. So, let us just run it and once we run it, let us see the output of the random forest model diagnostics. So, we are using this print function to see the model output and you can see here the, this is the confusion matrix for all the 12 classes. And of course, the class header is given here.

Now, the output if we want to check the output of the relative importance of each of these 5 covariates, which we are using like AACN, Drainage Index, Light Insulation and then TWI and a Gamma Total Count these 5 we want to see their importance in the classification. Again, we are going to use this importance function and then we are going to see that according to the mean decrease in Gini impurity criteria, you can see here that TWI is having the highest impact on this model, random forest categorical model followed by Altitude Above Channel Network and then Drainage Index and Gamma Total Count.

(Refer Slide Time: 04:43)

RStudio interface showing R code for random forest model diagnostics and prediction. The code includes:

```
173 # output random forest model diagnostics
174 int(hv.RF)
175 # output relative importance of each covariate
176 importance(hv.RF)
177
178 # prediction of classes
179 hdict(hv.RF, type = "response", newdata = DSM_data(training, ))
180
181 # calibration
182 hred.hv.RF <- predict(hv.RF, newdata = DSM_data(training, ))
183 hfcats(observed = DSM_data$terrain(training), predicted = C.pred.h
184
185 # external validation
186 vred.hv.RF <- predict(hv.RF, newdata = DSM_data[-training, ])
187 vfcats(observed = DSM_data$terrain[-training], predicted = V.pred.
188
```

The Environment pane shows variables like `area_colors`, `C.pred.hv.CS`, `C.pred.hv.RF`, `pred.hv.MNLR`, `training`, `V.pred.hv.CS`, `V.pred.hv.MNLR`, and `V.pred.hv.RF`.

RStudio interface showing the output of the `hdict` function. The output is a matrix of predicted values for each observation:

```
77 977 608 541 84 528 195 821 281 637 168 970 605
10 12 7 7 6 8 4 7 12 3 11 8 3
858 710 748 424 827 722 579 793 228 650 901 768 476
12 3 5 8 3 8 5 7 8 7 5 3 8
619 505 852 955 557 349 240 369 513 709 141 498 958
5 6 9 8 8 9 7 8 9 12 7 7 5 7
218 833 884 363 348 811 657 225 164 212 602 75 501
8 3 6 6 6 5 12 6 4 4 7 1 3 7
816 916 759 659 855 794 946 104 12 888 306
4 9 5 3 10 5 7 5 3 10 7
Levels: 1 2 3 4 5 6 7 8 9 10 11 12
```

RStudio interface showing the output of the `predict` function. The output is a matrix of predicted values for each observation:

```
> predict(hv.RF, type = "response", newdata = DSM_data(training, ))
787 267 623 984 587 836 44 544 107 239 480 378 49
6 8 4 8 8 2 10 7 5 10 12 11 7
487 548 662 90 890 667 277 728 262 779 410 302 942
9 7 10 6 7 8 9 4 7 7 10 1 5
876 428 953 411 869 223 383 362 881 466 213 523 736
3 8 8 10 5 3 8 10 5 8 5 9 4
353 507 106 593 52 156 814 304 766 631 993 652 298
9 12 5 7 10 8 4 4 6 11 4 8 6
653 517 972 575 634 291 385 196 489 672 853 572 935
4 8 4 4 7 11 7 12 4 7 3 9 8
880 771 440 442 294 936 405 60 434 65 470 601 351
```

RStudio interface showing R code for a random forest model and its diagnostics. The code includes training and validation steps for a model named 'hv.RF'.

```

173 random forest model diagnostics
174 RF
175 relative importance of each covariate
176 importance(hv.RF)
177
178 # Prediction of classes
179 hv.RF, type = "response", newdata = DSM_data(training, )
180
181 # calibration
182 C.pred.hv.RF <- predict(hv.RF, newdata = DSM_data(training, ))
183 observed = DSM_data$terron(training), predicted = C.pred.hv.RF
184
185 # external validation
186 V.pred.hv.RF <- predict(hv.RF, newdata = DSM_data[-training, ])
187 observed = DSM_data$terron[-training], predicted = V.pred.hv.RF
188
189 Random Forest :

```

Environment pane showing variable types:

```

area_colors cnr [1:12] "#f20000" "#38a800" "#7300ff" ...
C.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",...
C.pred.hv.RF Factor w/ 12 levels "1","2","3","4",...
pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",...
training int [1:700] 787 267 623 984 587 836 44 5...
V.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",...
V.pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",...
V.pred.hv.RF Factor w/ 12 levels "1","2","3","4",...


```

Console output showing a confusion matrix:

```

C:\Users\CM\Desktop\DSM\Categorical model >
  10  12  7  7  6  8  4  7  12  3  11  8  3
858 710 748 424 827 722 579 793 228 650 901 768 476
  12  3  5  8  3  8  5  7  8  7  5  3  8
619 505 852 955 557 349 240 369 513 709 141 498 958
  5  6  9  8  9  7  8  9  12  7  7  5  7
218 833 884 363 348 811 657 225 164 212 602 75 501
  8  3  6  6  5  12  6  4  4  7  1  3  7
816 916 759 659 855 794 946 104 12 888 306
  4  9  5  3  10  5  7  5  3  10  7
Levels: 1 2 3 4 5 6 7 8 9 10 11 12
> C.pred.hv.RF <- predict(hv.RF, newdata = DSM_data(training, ))
>

```



RStudio interface showing the same R code as above, but with the console output displaying accuracy metrics.

```

189 Random Forest :

```

Environment pane showing variable types:

```

area_colors cnr [1:12] "#f20000" "#38a800" "#7300ff" ...
C.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",...
C.pred.hv.RF Factor w/ 12 levels "1","2","3","4",...
pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",...
training int [1:700] 787 267 623 984 587 836 44 5...
V.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",...
V.pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",...
V.pred.hv.RF Factor w/ 12 levels "1","2","3","4",...


```

Console output showing accuracy metrics:

```

C:\Users\CM\Desktop\DSM\Categorical model >
$producers_accuracy
 1  2  3  4  5  6  7  8  9  10 11 12
100 100 100 100 100 100 100 100 100 100 100
$users_accuracy
 1  2  3  4  5  6  7  8  9  10 11 12
100 100 100 100 100 100 100 100 100 100 100
$kappa
[1] 1
>

```



RStudio interface showing the same R code as above, but with the console output displaying overall accuracy and kappa values.

```

189 Random Forest :

```

Environment pane showing variable types:

```

area_colors cnr [1:12] "#f20000" "#38a800" "#7300ff" ...
C.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",...
C.pred.hv.RF Factor w/ 12 levels "1","2","3","4",...
pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",...
training int [1:700] 787 267 623 984 587 836 44 5...
V.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",...
V.pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",...
V.pred.hv.RF Factor w/ 12 levels "1","2","3","4",...


```

Console output showing overall accuracy and kappa values:

```

C:\Users\CM\Desktop\DSM\Categorical model >
$overall_accuracy
[1] 100
$producers_accuracy
 1  2  3  4  5  6  7  8  9  10 11 12
100 100 100 100 100 100 100 100 100 100 100
$users_accuracy
 1  2  3  4  5  6  7  8  9  10 11 12
100 100 100 100 100 100 100 100 100 100 100
$kappa
[1] 100
>

```

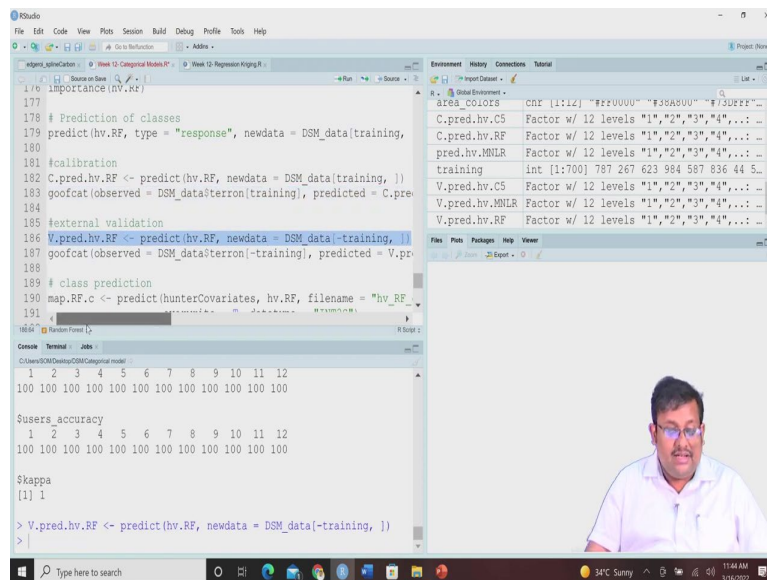


So, now we have predicted the importance we have seen the importance of all the variables. Now, the next step is to use the prediction of the classes. So, for the prediction of the class, we are going to use this type equal to response and then we are going to use the same model to see the prediction of all the classes.

So, you can see that the 787 observation will have the fifth terron and then 267th will have the eighth terron and then 623 will be having the fourth terron and so on. And then we are going to see the goodness of fit for the calibration samples. So, you can see here we are predicting based on the training samples, the predicted values, and at the same time, we are going to run this goofcat function to see the goodness of fit statistics.

So, here you can see that this is the confusion matrix. And of course, the overall accuracy then producer's accuracy and users accuracy and also the Kappa coefficient. So, it looks like from the calibration model the model is almost perfect. Now, to check whether this calibration model is okay or not, we are going to use the categorical validation or goodness of fit of the validation data. So, we are going to use these validation data that is minus training samples.

(Refer Slide Time: 06:25)



```
importance(nv.rf)
177
178 # Prediction of classes
179 predict(hv.rf, type = "response", newdata = DSM_data[training,
180
181 #calibration
182 C.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[training, ])
183 goofcat(observed = DSM_data$terror[training], predicted = C.pre
184
185 #external validation
186 V.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[-training, ])
187 goofcat(observed = DSM_data$terror[-training], predicted = V.pr
188
189 # class prediction
190 map.rf.c <- predict(hunterCovariates, hv.rf, filename = "hv_rf
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
26
```

```

176 }
177 }
178 } on of classes
179 }, RF, type = "response", newdata = DSM_data[training, ])
180 }
181 } on
182 } RF <- predict(hv.RF, newdata = DSM_data[training, ])
183 } served = DSM_data$terrain[training], predicted = C.pred.hv.RF)
184 }
185 } validation
186 } RF <- predict(hv.RF, newdata = DSM_data[-training, ])
187 } served = DSM_data$terrain[-training], predicted = V.pred.hv.RF)
188 }
189 } ediction
190 } - predict(hunterCovariates, hv.RF, filename = "hv_RF_class.tif")
191 }

```

```

Console Terminal Jobs
C:\Users\CM\Desktop\DSM\Categorical model
1 2 3 4 5 6 7 8 9 10 11 12
100 100 100 100 100 100 100 100 100 100 100 100

$users_accuracy
 1  2  3  4  5  6  7  8  9 10 11 12
100 100 100 100 100 100 100 100 100 100 100 100

$kappa
[1] 1

> V.pred.hv.RF <- predict(hv.RF, newdata = DSM_data[-training, ])

```

Environment: Global Environment, cnr [1:12] "#f20000" "#38a800" "#73dfff"..., area_colors Factor w/ 12 levels "1", "2", "3", "4", ..., C.pred.hv.CS Factor w/ 12 levels "1", "2", "3", "4", ..., C.pred.hv.RF Factor w/ 12 levels "1", "2", "3", "4", ..., pred.hv.MNLR Factor w/ 12 levels "1", "2", "3", "4", ..., training int [1:700] 787 267 623 984 587 836 44 5..., V.pred.hv.CS Factor w/ 12 levels "1", "2", "3", "4", ..., V.pred.hv.MNLR Factor w/ 12 levels "1", "2", "3", "4", ..., V.pred.hv.RF Factor w/ 12 levels "1", "2", "3", "4", ...

```

176 }
177 }
178 } on of classes
179 }, RF, type = "response", newdata = DSM_data[training, ])
180 }
181 } on
182 } RF <- predict(hv.RF, newdata = DSM_data[training, ])
183 } served = DSM_data$terrain[training], predicted = C.pred.hv.RF)
184 }
185 } validation
186 } RF <- predict(hv.RF, newdata = DSM_data[-training, ])
187 } served = DSM_data$terrain[-training], predicted = V.pred.hv.RF)
188 }
189 } ediction
190 } - predict(hunterCovariates, hv.RF, filename = "hv_RF_class.tif")
191 }

```

```

Console Terminal Jobs
C:\Users\CM\Desktop\DSM\Categorical model
1 2 3 4 5 6 7 8 9 10 11 12
25 0 36 50 74 70 61 39 13 48 20 39

$producers_accuracy
 1  2  3  4  5  6  7  8  9 10 11 12
25 0 48 34 53 63 45 49 100 55 23 63

$users_accuracy
 1  2  3  4  5  6  7  8  9 10 11 12
100 100 100 100 100 100 100 100 100 100 100 100

$kappa
[1] 0.4008742

>

```

Environment: Global Environment, cnr [1:12] "#f20000" "#38a800" "#73dfff"..., area_colors Factor w/ 12 levels "1", "2", "3", "4", ..., C.pred.hv.CS Factor w/ 12 levels "1", "2", "3", "4", ..., C.pred.hv.RF Factor w/ 12 levels "1", "2", "3", "4", ..., pred.hv.MNLR Factor w/ 12 levels "1", "2", "3", "4", ..., training int [1:700] 787 267 623 984 587 836 44 5..., V.pred.hv.CS Factor w/ 12 levels "1", "2", "3", "4", ..., V.pred.hv.MNLR Factor w/ 12 levels "1", "2", "3", "4", ..., V.pred.hv.RF Factor w/ 12 levels "1", "2", "3", "4", ...

```

176 }
177 }
178 } on of classes
179 }, RF, type = "response", newdata = DSM_data[training, ])
180 }
181 } on
182 } RF <- predict(hv.RF, newdata = DSM_data[training, ])
183 } served = DSM_data$terrain[training], predicted = C.pred.hv.RF)
184 }
185 } validation
186 } RF <- predict(hv.RF, newdata = DSM_data[-training, ])
187 } served = DSM_data$terrain[-training], predicted = V.pred.hv.RF)
188 }
189 } ediction
190 } - predict(hunterCovariates, hv.RF, filename = "hv_RF_class.tif")
191 }

```

```

Console Terminal Jobs
C:\Users\CM\Desktop\DSM\Categorical model
11 0 0 3 1 1 0 0 0 1 0 1 2 0
12 0 0 0 0 0 0 0 1 0 0 2 0 5

$producers_accuracy
 1  2  3  4  5  6  7  8  9 10 11 12
25 0 36 50 74 70 61 39 13 48 20 39

$users_accuracy
 1  2  3  4  5  6  7  8  9 10 11 12
100 100 100 100 100 100 100 100 100 100 100 100

$kappa
[1] 0.47

>

```

Environment: Global Environment, cnr [1:12] "#f20000" "#38a800" "#73dfff"..., area_colors Factor w/ 12 levels "1", "2", "3", "4", ..., C.pred.hv.CS Factor w/ 12 levels "1", "2", "3", "4", ..., C.pred.hv.RF Factor w/ 12 levels "1", "2", "3", "4", ..., pred.hv.MNLR Factor w/ 12 levels "1", "2", "3", "4", ..., training int [1:700] 787 267 623 984 587 836 44 5..., V.pred.hv.CS Factor w/ 12 levels "1", "2", "3", "4", ..., V.pred.hv.MNLR Factor w/ 12 levels "1", "2", "3", "4", ..., V.pred.hv.RF Factor w/ 12 levels "1", "2", "3", "4", ...


```

183 goofcat(observed = DSM_data$terrortraining, predicted = C.p
184
185 #external validation
186 V.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[-training, ])
187 goofcat(observed = DSM_data$terrortraining, predicted = V.pr
188
189 # class prediction
190 map.rf.c <- predict(hunterCovariates, hv.rf, filename = "hv_rf_
191               overwrite = T, datatype = "INT2S")
192 map.rf.c <- as.factor(map.rf.c)
193 rat <- levels(map.rf.c)[1]
194 rat[["terrortraining"]] <- c("HVT_001", "HVT_002", "HVT_003", "HVT_004"
195                            "HVT_006", "HVT_007", "HVT_008", "HVT_009"
196 levels(map.rf.c) <- rat
197 # plot
198
199 Random Forest :
200
Console Terminal Jobs
C:\Users\GMM\Desktop\GMM\Catgical model
$Kappa
[1] 0.4008742
> map.rf.c <- predict(hunterCovariates, hv.rf, filename = "hv_rf_cla
ss.tif", format = "GTiff",
+                   overwrite = T, datatype = "INT2S")
Warning message:
In .gd_SetProject(object, ...) : NOT UPDATED FOR PROJ >= 6
> map.rf.c <- as.factor(map.rf.c)
> rat <- levels(map.rf.c)[1]
>

```

```

183 goofcat(observed = DSM_data$terrortraining, predicted = C.p
184
185 #external validation
186 V.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[-training, ])
187 goofcat(observed = DSM_data$terrortraining, predicted = V.pr
188
189 # class prediction
190 map.rf.c <- predict(hunterCovariates, hv.rf, filename = "hv_rf_
191               overwrite = T, datatype = "INT2S")
192 map.rf.c <- as.factor(map.rf.c)
193 rat <- levels(map.rf.c)[1]
194 rat[["terrortraining"]] <- c("HVT_001", "HVT_002", "HVT_003", "HVT_004"
195                            "HVT_006", "HVT_007", "HVT_008", "HVT_009"
196 levels(map.rf.c) <- rat
197 # plot
198
199 Random Forest :
200
Console Terminal Jobs
C:\Users\GMM\Desktop\GMM\Catgical model
$Kappa
[1] 0.4008742
> map.rf.c <- predict(hunterCovariates, hv.rf, filename = "hv_rf_cla
ss.tif", format = "GTiff",
+                   overwrite = T, datatype = "INT2S")
Warning message:
In .gd_SetProject(object, ...) : NOT UPDATED FOR PROJ >= 6
> map.rf.c <- as.factor(map.rf.c)
> rat <- levels(map.rf.c)[1]
> View(rat)
>

```

```

183 goofcat(observed = DSM_data$terrortraining, predicted = C.p
184
185 #external validation
186 V.pred.hv.rf <- predict(hv.rf, newdata = DSM_data[-training, ])
187 goofcat(observed = DSM_data$terrortraining, predicted = V.pr
188
189 # class prediction
190 map.rf.c <- predict(hunterCovariates, hv.rf, filename = "hv_rf_
191               overwrite = T, datatype = "INT2S")
192 map.rf.c <- as.factor(map.rf.c)
193 rat <- levels(map.rf.c)[1]
194 rat[["terrortraining"]] <- c("HVT_001", "HVT_002", "HVT_003", "HVT_004"
195                            "HVT_006", "HVT_007", "HVT_008", "HVT_009"
196 levels(map.rf.c) <- rat
197 # plot
198
199 Random Forest :
200
Console Terminal Jobs
C:\Users\GMM\Desktop\GMM\Catgical model
+                   overwrite = T, datatype = "INT2S")
Warning message:
In .gd_SetProject(object, ...) : NOT UPDATED FOR PROJ >= 6
> map.rf.c <- as.factor(map.rf.c)
> rat <- levels(map.rf.c)[1]
> View(rat)
> rat[["terrortraining"]] <- c("HVT_001", "HVT_002", "HVT_003", "HVT_004",
+                             "HVT_005",
+                             "HVT_006", "HVT_007", "HVT_008", "HVT_009",
+                             "HVT_010", "HVT_011", "HVT_012")
> levels(map.rf.c) <- rat
>

```

Now, class prediction for predict the class again we are going to run those scripts which using the, which uses the hex colour codes as we have seen previously, so we are going to use this predict function for the class prediction and then we are going to map it, now we are going to add a new column called raster in the raster attribute table. So, you will see this rat you can see 12 observation of one variable. So, we are going to add these terrons with these indicators. So, if we just click on this, you will see that.

(Refer Slide Time: 07:52)

The screenshot shows the RStudio interface. The Environment pane on the right displays the variable 'rat' with 12 observations of 2 variables. The console shows the following R code and output:

```

Warning message:
In .gd_setProject(object, ...) : NOT UPDATED FOR PROJ >= 6
> map.RF.c <- as.factor(map.RF.c)
> rat <- levels(map.RF.c)[1:12]
> View(rat)
> rat[["terron"]] <- c("HVT_001", "HVT_002", "HVT_003", "HVT_004",
+ "HVT_005",
+ "HVT_006", "HVT_007", "HVT_008", "HVT_009",
+ "HVT_010", "HVT_011", "HVT_012")
> levels(map.RF.c) <- rat
> View(rat)

```

The screenshot shows the RStudio interface with R code for class prediction and a resulting raster plot. The console shows the following R code and output:

```

187 goofcat(observed = DSM_data$terron[-training], predicted = V.pred)
188 # class prediction
189 map.RF.c <- predict(hunterCovariates, hv.RF, filename = "hv_RF_
190                   overwrite = T, datatype = "INT2S")
191
192 map.RF.c <- as.factor(map.RF.c)
193 rat <- levels(map.RF.c)[1:12]
194 rat[["terron"]] <- c("HVT_001", "HVT_002", "HVT_003", "HVT_004"
195                   "HVT_006", "HVT_007", "HVT_008", "HVT_009"
196 levels(map.RF.c) <- rat
197 # plot
198 area_colors <- c("#FF0000", "#38A800", "#73DFFF", "#FFBAP", "#
199 "#FFA77F", "#7AF5CA", "#D7B09E", "#CCCCCC", "#
200 levelplot(map.RF.c, col.regions = area_colors, xlab = "", ylab
201

```

The plot shows a map with colored regions corresponding to the 'rat' variable. The x-axis ranges from 335000 to 350000 and the y-axis ranges from 630000 to 6375000. A legend on the right shows the color coding for the 12 observations.

Now, rat will be developed with two variables one is the ID and the second one is the terron. So, from this ID and terron we can see that these are the indicators. Now, let us go back and then we have assigned these plot colours using these hex colour codes. And then we are

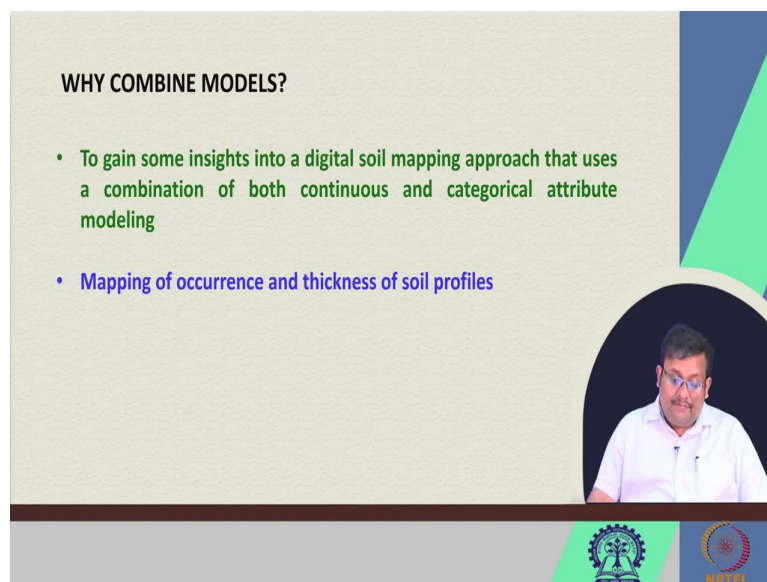
going to use this just like previously we are going to use this for predicting and mapping the properties of the terrons in the area of interest.

So, guys, what we have done again just like previously, we have first call the library of random forest and we then fitted the model random forest model specifying the number of trees as well as the number of features and once we have done that the model will produce the results we have seen they are variable importance.

Apart from that, we have produced the predicted based on our calibration data set, we have seen the categorical goodness of fit and then we have seen the validation goodness of fit by predicting their values and then using these goofcat function after that, using this random forests module, we have prepared the map and also we have included a new raster attribute table where we have indicated the serial numbers of the terron classes starting from Hunter Valley Terron 001 to Hunter Valley Terron 012.

And then we assign them each of these terron class we have assigned them a colour using the hex colour codes and then we have plotted them just like here. So, this shows that how we can develop this soil classification map using the random forest classification, random forests classification algorithm.

(Refer Slide Time: 10:10)



WHY COMBINE MODELS?

- To gain some insights into a digital soil mapping approach that uses a combination of both continuous and categorical attribute modeling
- Mapping of occurrence and thickness of soil profiles

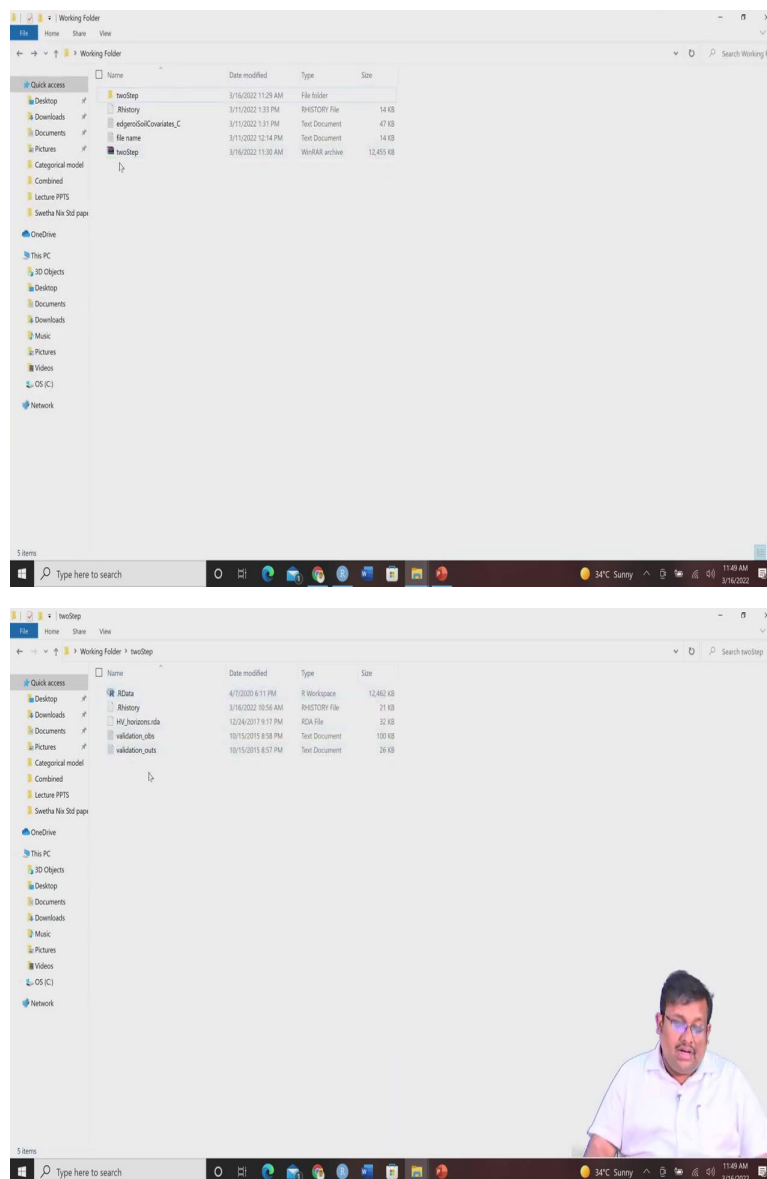
The slide features a video inset of a man in a white shirt and glasses speaking. At the bottom, there are logos for a university and NPTEL.

Now, let us go back to our Combine Model, now, what is the motivation between this Combine Model, why we go for the combining this model here, I will show you how we can combine this categorical model and then we can also, how we can combine this categorical

model and continuous model together in digital soil mapping using different types of machine learning algorithms.

So, our objective is to gain insight into digital soil mapping approach that uses a combination of both continuous and categorical attribute modelling first of all, we are going to see the occurrence that means presence or absence of particular soil horizon in a soil profile and then if they are present then we are going to measure we are going to predict their depth using a regression model. So, first we are going to use the categorical model, then we are going to use the continuous model.

(Refer Slide Time: 11:20)

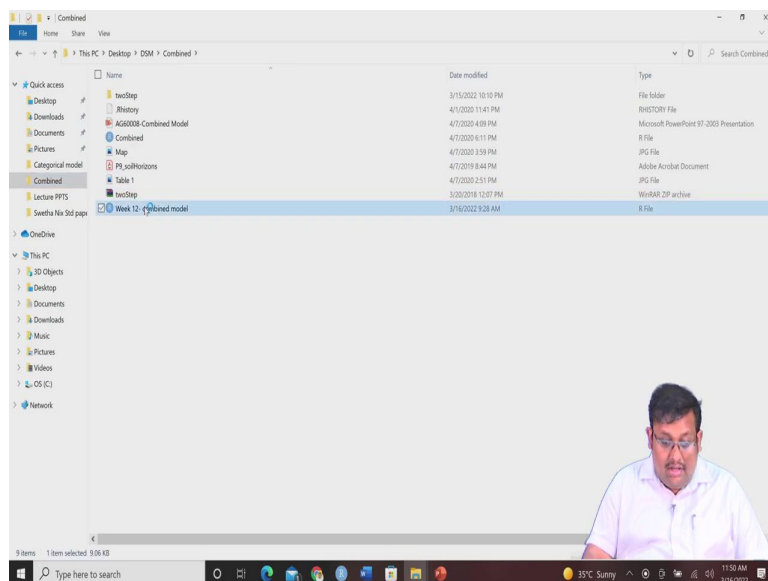
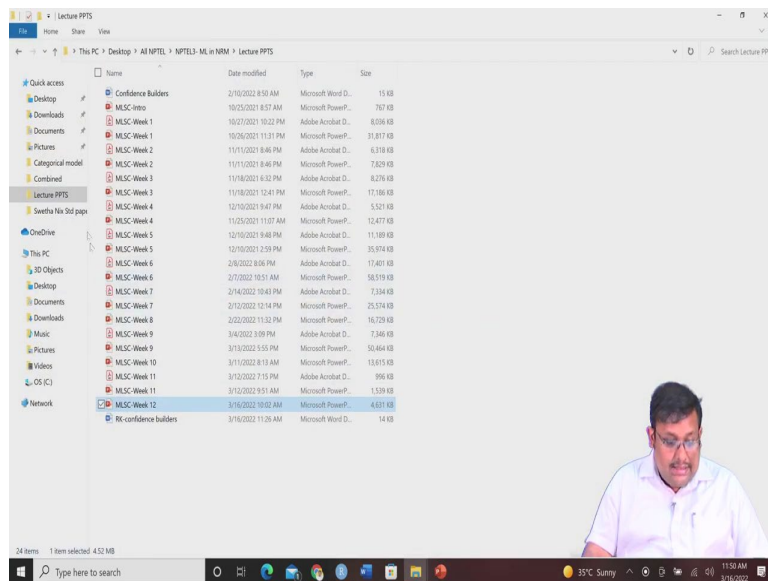


Now, for this, remember guys, we have to use some files. So, this file I will attach in our class forum, please download it. So, if I go back to our working folder, in the working folder,

you will see there is a file called twoStep. So, this twoStep model, I will attach, I will post it in our class forum, and then you can extract this into your working folder. So, if you open this twoStep folder, you will see there is a data file and also there will be the observation, validation observation as well as the validation outputs.

So, these will be required for different operation in this combined model. So, let me just go through and these codes and walk you through these codes and let you know, how we use these things. So, once you download this code of this twoStep, you will just download this folder and extract it in your working folder, so that you can work with it.

(Refer Slide Time: 12:25)



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

```

246 library(snow)
247 beginCluster(4)
248 Al.class <- clusterR(hunterCovariates, predict, args = list(mm2
249                       filename = "class_Al.tif", format = "GTiff
250 # Apply AI horizon depth model spatially using the raster multi
251 # facility
252 Al.depth <- clusterR(hunterCovariates, predict, args = list(qrf
253                       filename = "depth_Al.tif", format = "GTiff
254 endCluster()
255 # Mask out areas where horizon is absent
256 Al.class[Al.class == 0] <- NA
257 mr <- mask(Al.depth, Al.class)
258 writeRaster(mr, filename = "depth_Al_mask.tif", format = "GTiff
259
260
261

```

The Environment pane shows the following objects:

- rat: 12 obs. of 2 variables
- ov.MNLR: List of 26
- area_colors: chr [1:12] "#FF0000" "#38A800" "#73DFFF"...
- C.pred.hv.C5: Factor w/ 12 levels "1","2","3","4",... ..
- C.pred.hv.RF: Factor w/ 12 levels "1","2","3","4",... ..
- pred.hv.MNLR: Factor w/ 12 levels "1","2","3","4",... ..
- training: int [1:7001] 787 267 623 984 587 836 44 5...

The map shows a geographical region with a color scale from 7 to 12. A person is visible in the bottom right corner of the map area.

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

```

1 #1342 soil profile and core description
2
3 # We want to predict spatial distribution of the occurrence of
4 # And then where those horizons occur, predict their depth.
5
6
7 # library
8 library(ihvir)
9 library(sp)
10 library(raster)
11 install.packages("acp")
12 library(acp)
13 library(randomForest)
14
15 setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
16

```

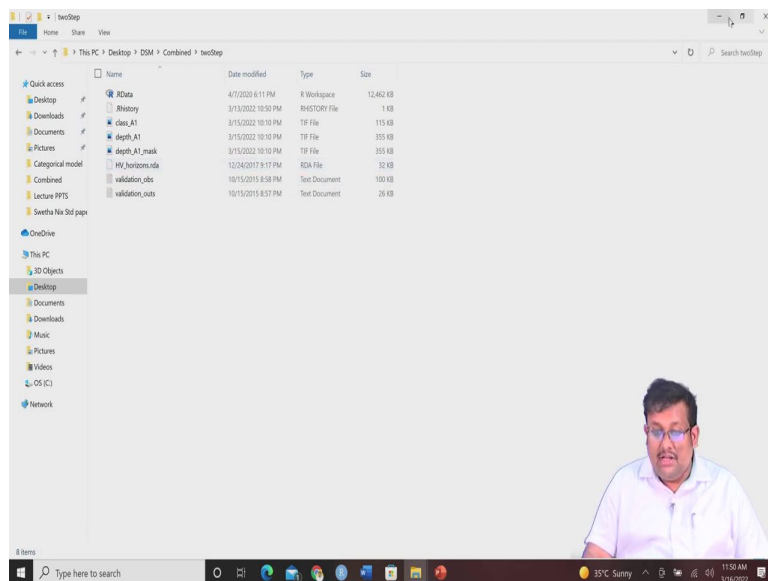
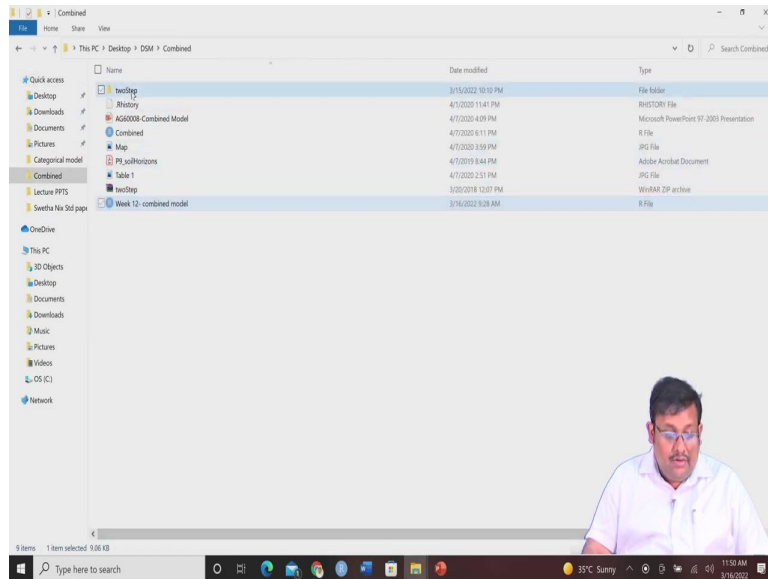
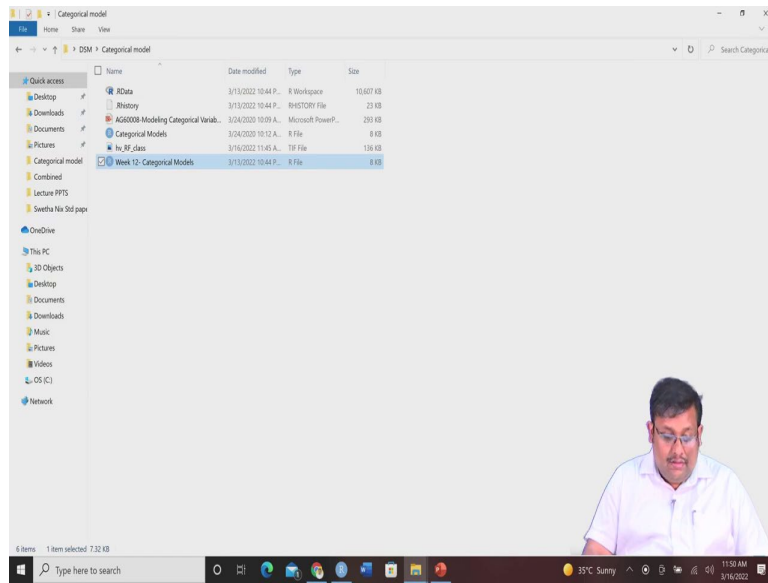
The Environment pane shows the following objects:

- rat: 12 obs. of 2 variables
- ov.MNLR: List of 26
- area_colors: chr [1:12] "#FF0000" "#38A800" "#73DFFF" "#FFEB3F" "#A8A800" "#4070FF" "#FFA77F" "#7AF5CA" "#D7B09E" "#CCCCCC" "#B4D79E" "#FFFF00"
- C.pred.hv.C5: Factor w/ 12 levels "1","2","3","4",... ..
- C.pred.hv.RF: Factor w/ 12 levels "1","2","3","4",... ..
- pred.hv.MNLR: Factor w/ 12 levels "1","2","3","4",... ..
- training: int [1:7001] 787 267 623 984 587 836 44 5...

The map shows a geographical region with a color scale from 7 to 12. A person is visible in the bottom right corner of the map area.

So, let me just go ahead and use the combined model. So, these are the scripts for the combined model.

(Refer Slide Time: 12:42)



So, in this data set of this twoStep model, if you go to the twoStep model, you will see that there will be HV horizons data.

(Refer Slide Time: 12:51)

The screenshot shows the RStudio interface. The script in the editor includes the following code:

```
1 #1342 soil profile and core description
2
3 # We want to predict spatial distribution of the occurrence of
4 # And then where those horizons occur, predict their depth.
5
6
7 # library
8 library(ithir)
9 library(sp)
10 library(raster)
11 install.packages("agg")
12 library(agg)
13 library(randomForest)
14
15 setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
16
```

The console output shows the execution of the script, including the creation of a raster object 'rat' and the assignment of color values to 'area_colors'. The environment pane on the right shows the loaded packages and variables.

The map in the bottom right corner displays a geographical area with a color-coded legend on the right side, ranging from 7 to 12. A person's face is visible in the bottom right corner of the map area.

The screenshot shows the RStudio interface. The script in the editor includes the following code:

```
1
2
3 ince of A1, A2, AB, B1, B21,B22, B23, B24, BC, and C horizons,
4 h.
5
6
7
8
9
10
11
12
13
14
15
16
```

The console output shows the execution of the script, including the creation of a raster object 'rat' and the assignment of color values to 'area_colors'. The environment pane on the right shows the loaded packages and variables.

The map in the bottom right corner displays a geographical area with a color-coded legend on the right side, ranging from 7 to 12. A person's face is visible in the bottom right corner of the map area.

RStudio interface showing R code execution and environment details.

```

4 # And then where those horizons occur, predict their depth.
5
6
7 # library
8 library(ithir)
9 library(sp)
10 library(raster)
11 install.packages("agg")
12 library(agg)
13 library(randomForest)
14
15 setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
16
17 # data
18 load("HV_horizons.rda")
19

```

Environment pane shows:

- dat: 12 obs. of 2 variables
- ov.MNLR: List of 26
- area_colors: chr [1:12] "#FF0000" "#38A800" "#73DFFF"...
- C.pred.hv.C5: Factor w/ 12 levels "1","2","3","4",... ..
- C.pred.hv.RF: Factor w/ 12 levels "1","2","3","4",... ..
- pred.hv.MNLR: Factor w/ 12 levels "1","2","3","4",... ..
- training: int [1:7001] 787 267 623 984 587 836 44 5...

RStudio interface showing R code execution and environment details.

```

4 # And then where those horizons occur, predict their depth.
5
6
7 # library
8 library(ithir)
9 library(sp)
10 library(raster)
11 install.packages("agg")
12 library(agg)
13 library(randomForest)
14
15 setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
16
17 # data
18 load("HV_horizons.rda")
19

```

Environment pane shows:

- dat: 12 obs. of 2 variables
- ov.MNLR: List of 26
- area_colors: chr [1:12] "#FF0000" "#38A800" "#73DFFF"...
- C.pred.hv.C5: Factor w/ 12 levels "1","2","3","4",... ..
- C.pred.hv.RF: Factor w/ 12 levels "1","2","3","4",... ..
- pred.hv.MNLR: Factor w/ 12 levels "1","2","3","4",... ..
- training: int [1:7001] 787 267 623 984 587 836 44 5...

Console output:

```

combine

The following objects are masked from 'package:raster':
  metadata, metadata<-

The following object is masked from 'package:stats':
  filter

```

RStudio interface showing R code execution and environment details.

```

8 library(ithir)
9 library(sp)
10 library(raster)
11 install.packages("agg")
12 library(agg)
13 library(randomForest)
14
15 setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
16
17 # data
18 load("HV_horizons.rda")
19 View(dat)
20
21 #soil profile data dat is arranged in a flat file where each ro
22 #then there are 11 further columns that are binary indicators
23

```

Environment pane shows:

- dat: 1342 obs. of 25 variables
- DSM_data: 1000 obs. of 8 variables
- hunterCovaria: Large RasterStack (2902500 elements, 2...
- ohv.C5: List of 16
- ohv.MNLR: List of 26
- ohv.RF: List of 19
- ohvTerrainDat: Formal class 'SpatialPointsDataFrame'
- map.C5.c: Formal class 'RasterLayer'

Console output:

```

metadata, metadata<-

The following object is masked from 'package:stats':
  filter

> library(agg)
> library(randomForest)
> setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
> # data
> load("HV_horizons.rda")

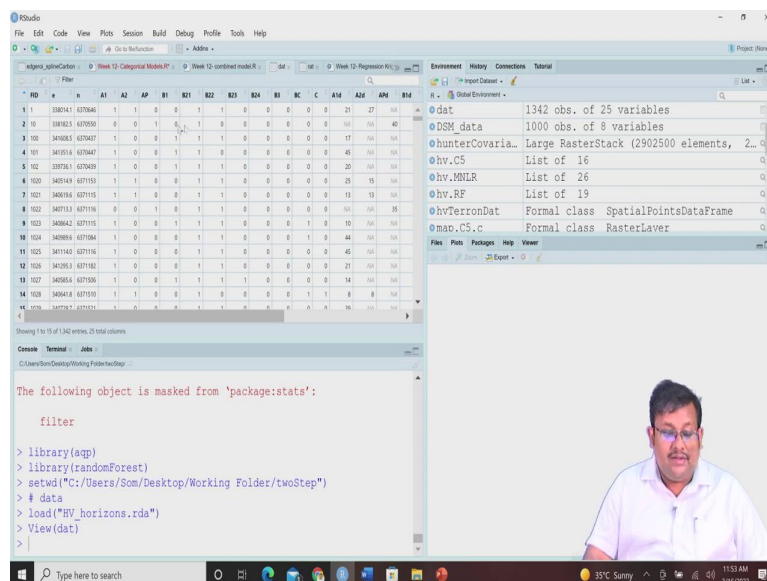
```

So, these HV horizon data is basically having 1342 soil profile and code description. So, we are going to use the, this data set and we are going to we want to predict the spatial distribution of their occurrence specifically on, for A1, A2, AP and then B1, B21, B22, B23, B24, BC and C horizons and where they are presents, when they are present we are going to predict their depth. So, let us call all the required libraries.

So, the required library already you know that is ithir library is required, the library sp will be required, library raster will be required. And then you have to install this package called a aqp once you install then call this library aqp, aqp is basically having the is useful for doing some soil DSMs calculations. So, this is very important. Now, once we have, call this library aqp the next is calling the library random forest because we are going to use this random forest model.

So, let us call it after this we have to set our working directory, we can either manually go and see set the twoStep folder or we can give directly the path of this twoStep folder here I have used the path of this twoStep folder it is contained within this working folder and for this I have used the set working directory function. So, let me just go and select this working directory twoStep because all the maps and all these outputs will be produced in this twoStep folder now. Now, in the twoStep folder there is HV underscore horizon dot rda file. So, let us call these things and then let us view the file.

(Refer Slide Time: 14:52)



The screenshot shows the RStudio interface. The main window displays a data frame with 1342 rows and 25 columns. The columns are labeled with soil horizon codes: A1, A2, AP, B1, B2, B22, B23, B24, B3, BC, C, A1d, A2d, APd, B1d, B2d, B22d, B23d, B24d, B3d, BCd, Cd. The console window shows the following R code:

```

> library(agg)
> library(randomForest)
> setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
> # data
> load("HV_horizons.rda")
> View(dat)

```

This screenshot is identical to the one above, showing the RStudio interface with the same data frame and console window.

So, if you see this file, this data set of 1340 observation around so you will see there will be easting northing and then A1, A2, AP, B1, B2 and B22, B23, B24, B3, BC, C and all these horizons. So, this 1 and 0 basically shows the presence and absence. So, presence is denoted by 1 and absence is denoted by 0.

So, here you can see these are the samples the number of samples and you can see these A1, A2 are basically this is showing each row is showing a soil profile. So, this is for soil profile these another soil profile and the soil profile it shows the arrangement of the horizon. So, the first soil profile gives you the arrangement of the depth of these horizons and second profile gives you the arrangement of these horizons and so, on.

And after this C horizon, you will see these A1d, A2d and all these things, so the d stands for the depth. So, when there are present so, for example, A1 is present, so, the A1 depth is 21

centimetre, then A2 is present, so A2 depth is 27 centimetre and when there are absent there having the any values. So, for this you can see, this is the dataset. So, once we have this data set, let us go back and see.

(Refer Slide Time: 16:17)

The screenshot shows the RStudio interface. The script editor contains the following R code:

```
13 library(randomForest)
14
15 setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
16
17 # data
18 load("HV_horizons.rda")
19 View(dat)
20
21 #soil profile data dat is arranged in a flat file where each ro
22 #then there are 11 further columns that are binary indicators
23 # of whether a horizon class is present or not (indicated as 1
24 #The following 11 columns after the binary columns indicate the
25
26
27 str(dat)
28
```

The console shows the following output:

```
The following object is masked from 'package:stats':
  filter

> library(agg)
> library(randomForest)
> setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
> # data
> load("HV_horizons.rda")
> View(dat)
>
```

The Environment pane on the right lists the following objects:

- dat: 1342 obs. of 25 variables
- DSM_data: 1000 obs. of 8 variables
- hunterCovaria_: Large RasterStack (2902500 elements, 2...
- hv.C5: List of 16
- hv.MNLR: List of 26
- hv.RF: List of 19
- hvTerrorDat: Formal class 'SpatialPointsDataFrame'
- map.C5.c: Formal class 'RasterLayer'

A video feed of a man in a white shirt is visible in the bottom right corner of the RStudio window.

The screenshot shows the RStudio interface. The script editor contains the following R code:

```
13
14
15 /Working Folder/twoStep")
16
17
18
19
20
21 ranged in a flat file where each row is a soil profile.
22 columns that are binary indicators
23 is present or not (indicated as 1 and 0 respectively).
24 ter the binary columns indicate the horizon depth for the given
25
26
27
28
```

The console shows the following output:

```
The following object is masked from 'package:stats':
  filter

> library(agg)
> library(randomForest)
> setwd("C:/Users/Som/Desktop/Working Folder/twoStep")
> # data
> load("HV_horizons.rda")
> View(dat)
>
```

The Environment pane on the right lists the following objects:

- dat: 1342 obs. of 25 variables
- DSM_data: 1000 obs. of 8 variables
- hunterCovaria_: Large RasterStack (2902500 elements, 2...
- hv.C5: List of 16
- hv.MNLR: List of 26
- hv.RF: List of 19
- hvTerrorDat: Formal class 'SpatialPointsDataFrame'
- map.C5.c: Formal class 'RasterLayer'

A video feed of a man in a white shirt is visible in the bottom right corner of the RStudio window.

RStudio interface showing code execution. The console displays the following output:

```
$ AP : int 0 1 0 0 0 0 0 1 0 0 ...
$ B1 : int 0 0 1 1 1 0 0 0 1 0 ...
$ B21 : int 1 1 1 1 1 1 1 1 1 1 ...
$ B22 : int 1 0 1 0 1 1 1 1 1 1 ...
$ B23 : int 0 0 0 0 0 0 0 0 0 0 ...
$ B24 : int 0 0 0 0 0 0 0 0 0 0 ...
$ B3 : int 0 0 0 0 0 0 0 0 0 0 ...
$ BC : int 0 0 0 0 0 0 0 0 1 1 ...
$ C : int 0 0 0 0 0 0 0 0 0 0 ...
$ A1d : num 21 NA 17 45 20 25 13 NA 10 44 ...
$ A2d : num 27 NA NA NA NA 15 13 NA NA NA ...
$ APd : num NA 40 NA NA NA NA 35 NA NA ...
```

The Environment pane on the right shows the following objects:

- dat: 1342 obs. of 25 variables
- DSM_data: 1000 obs. of 8 variables
- hunterCovaria: Large RasterStack (2902500 elements, 2...)
- ohv.C5: List of 16
- ohv.MNLR: List of 26
- ohv.RF: List of 19
- ohv.TerronDat: Formal class SpatialPointsDataFrame
- map.C5.c: Formal class RasterLayer

RStudio interface showing code execution. The console displays the following output:

```
$ data.frame: 1342 obs. of 25 variables:
 $ FID : Factor w/ 1342 levels "1","10","100",...: 1 2 3 4 5 6 7 8 9
 10 ...
 $ e : num 338014 338183 341609 341352 339736 ...
 $ n : num 6370446 6370550 6370437 6370447 6370439 ...
 $ A1 : int 1 0 1 1 1 1 1 0 1 1 ...
 $ A2 : int 1 0 0 0 0 1 1 0 0 0 ...
 $ AP : int 0 1 0 0 0 0 0 1 0 0 ...
 $ B1 : int 0 0 1 1 1 0 0 0 1 0 ...
 $ B21 : int 1 1 1 1 1 1 1 1 1 1 ...
 $ B22 : int 1 0 1 0 1 1 1 1 1 1 ...
 $ B23 : int 0 0 0 0 0 0 0 0 0 0 ...
 $ B24 : int 0 0 0 0 0 0 0 0 0 0 ...
```

The Environment pane on the right shows the same objects as in the previous screenshot.

RStudio interface showing code execution. The console displays the following output:

```
$ B3 : int 0 0 0 0 0 0 0 0 0 0 ...
$ BC : int 0 0 0 0 0 0 0 0 1 1 ...
$ C : int 0 0 0 0 0 0 0 0 0 0 ...
$ A1d : num 21 NA 17 45 20 25 13 NA 10 44 ...
$ A2d : num 27 NA NA NA NA 15 13 NA NA NA ...
$ APd : num NA 40 NA NA NA NA 35 NA NA ...
$ B1d : num NA NA 25 25 30 NA NA NA 30 NA ...
$ B21d : num 26 60 26 30 25 58 40 20 38 ...
$ B22d : num 26 NA 32 NA 20 20 16 25 18 ...
$ B23d : int NA NA NA NA NA NA NA NA NA ...
$ B24d : int NA NA NA NA NA NA NA NA NA ...
$ B3d : int NA NA NA NA NA NA NA NA NA ...
```

The Environment pane on the right shows the same objects as in the previous screenshots.

So, you can see here, the soil profile data is arranged in a flat file, where each row is a soil profile, then there is 11 further columns who are, that they are binary indicators 1 or 0 I have already told you whether horizon is class is present or not they indicated by 1 and 0 and the following 11 columns are showing the binary column after this binary columns are showing the horizon depth.

Now, once we have seen these data set let us see the structure of the datasets, so the structure of the data set will be again very clear starting from the factor then e and n of course, these are the numerical variables and then the integer variables are there and finally, their depth which is also numerical variables. Now, the next now it is a tabular format it is a data frame format next we have to convert it to the spatial objects.

(Refer Slide Time: 17:15)

The screenshot shows an RStudio session with the following code in the editor:

```

30 # convert data to spatial object
31 coordinates(dat) <- ~e + n
32
33
34 # covariates
35 data(hunterCovariates)
36
37
38 # covariates
39 names(hunterCovariates)
40
41 # resolution
42 res(hunterCovariates)
43
44 #These covariates are all registered to the common spatial reso
45
46

```

The Environment pane on the right lists the following objects:

- @dat: Formal class SpatialPointsDataFrame
- @DSM_data: 1000 obs. of 8 variables
- @hunterCovaria_: Large RasterStack (2902500 elements, 2...
- @hv.C5: List of 16
- @hv.MNLR: List of 26
- @hv.RF: List of 19
- @hvTerrorDat: Formal class SpatialPointsDataFrame
- @mac.C5.c: Formal class RasterLayer

The console shows the output of the code:

```

> # convert data to spatial object
> coordinates(dat) <- ~e + n
> # covariates
> data(hunterCovariates)
> # covariates
> names(hunterCovariates)
[1] "AACN" "Drainage.Index" "Light.Insolation"
[4] "TWI" "I" "Gamma.Total.Count"
>

```

The screenshot shows an RStudio session with the following code in the editor:

```

41 # resolution
42 res(hunterCovariates)
43
44 #These covariates are all registered to the common spatial reso
45
46 #For a quick check, lets overlay the soil profile points onto t
47
48 plot(hunterCovariates[["AACN"]])
49 points(dat, pch = 20)
50
51
52 #take the covariate intersection of the soil profile data, and
53
54 # Covariate extract
55 ext <- extract(hunterCovariates, dat, df = T, method = "simple"
56
57

```

The Environment pane on the right lists the following objects:

- @dat: Formal class SpatialPointsDataFrame
- @DSM_data: 1000 obs. of 8 variables
- @hunterCovaria_: Large RasterStack (2902500 elements, 2...
- @hv.C5: List of 16
- @hv.MNLR: List of 26
- @hv.RF: List of 19
- @hvTerrorDat: Formal class SpatialPointsDataFrame
- @mac.C5.c: Formal class RasterLayer

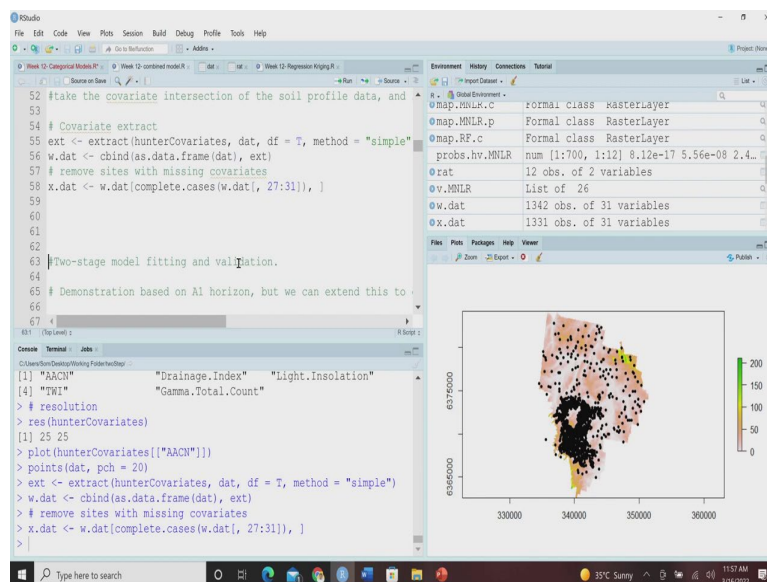
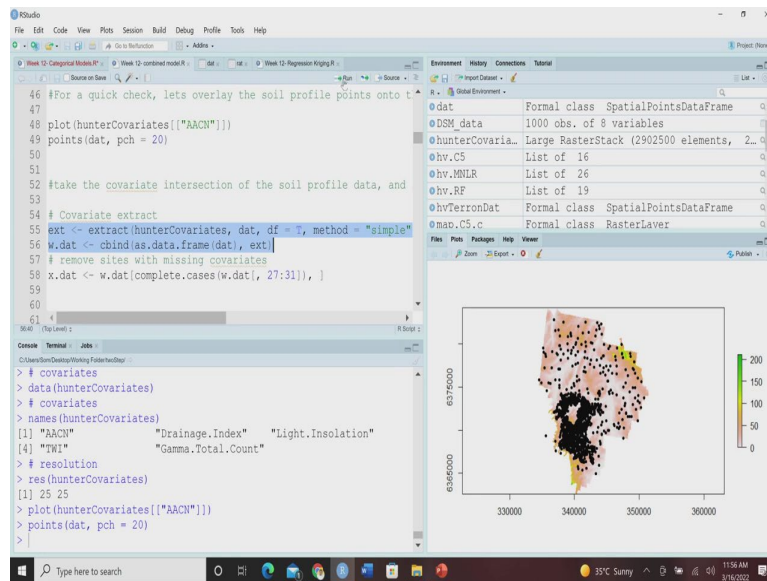
The console shows the output of the code:

```

> # covariates
> data(hunterCovariates)
> # covariates
> names(hunterCovariates)
[1] "AACN" "Drainage.Index" "Light.Insolation"
[4] "TWI" "Gamma.Total.Count"
> # resolution
> res(hunterCovariates)
[1] 25 25
> plot(hunterCovariates[["AACN"]])
> points(dat, pch = 20)
>

```

A map of the study area is displayed in the bottom right, showing a spatial distribution of values with a legend on the right ranging from 0 to 200. The map axes are labeled with coordinates: 330000, 340000, 6375000, and 6395000.



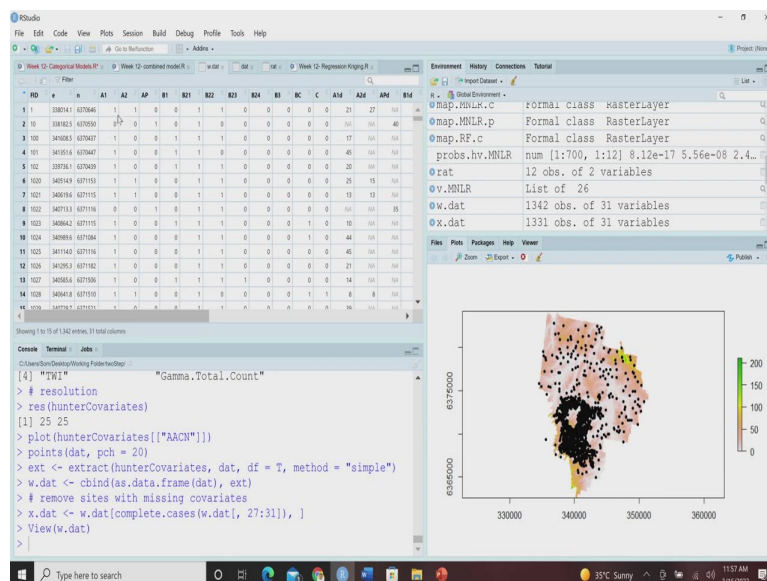
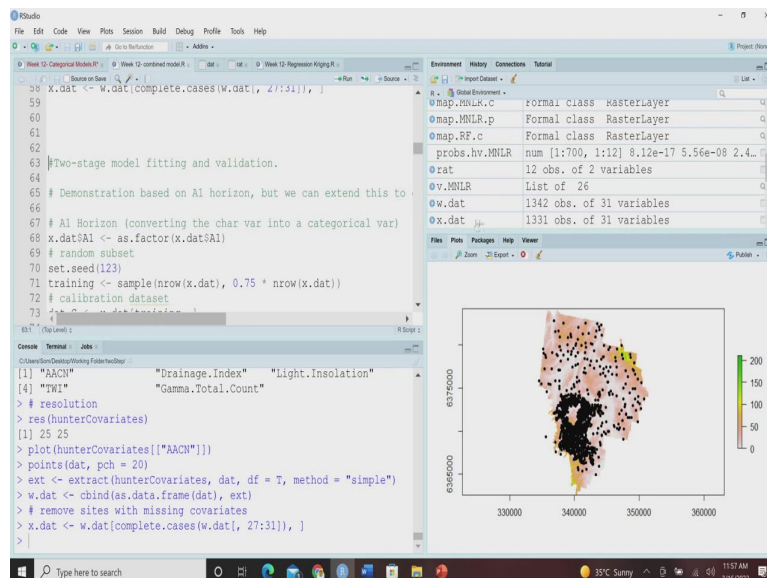
So, we must understand we must instruct R that consider these e and n as the coordinates. So, R now, you will see there is a spatial points data frame. Now, once we have converted this into spatial points data frame, we are downloading these Hunter Covariates data after we download the Hunter Covariates data of course, let us see what are the names of these there will be 5, 1, 2, 3, 4, 5 ACNN, Drainage Index, Light Installation and then TWI and Gamma Total Count and let us see this resolution.

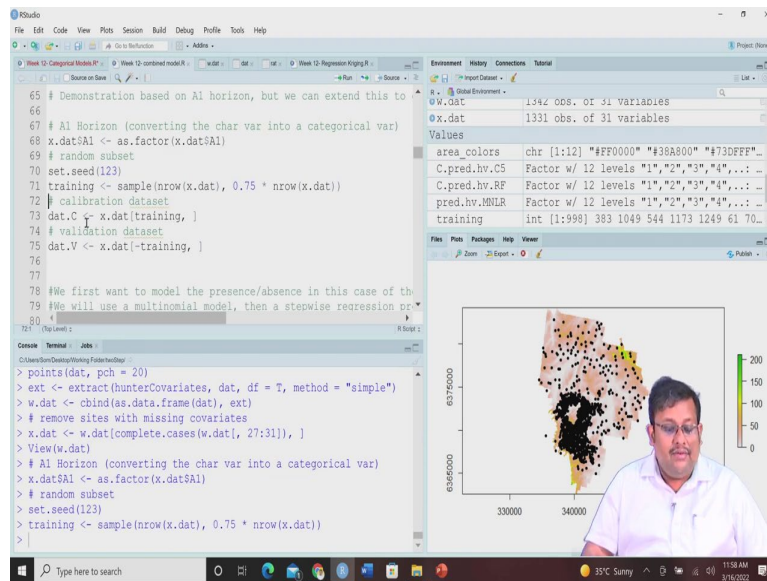
So, it is a raster stack and you will see the resolution is 25 metre by 25. So, it is a large raster stack where we have stacked all these 5 covariate and the resolution is 25 by 25 they are having all the common spatial resolution. So, for a quick display, let us first plot these for example, this ACNN and display the points and overlay the points, sampling points and this will look like this.

So, we are plotting first the Altitude Above Channel Network raster file and overlaying the samples over there, the point samples over there. So, 1342 samples whatever we may have. So, it is yes 1342 samples, so 1342 I did not been plotted here. Now, once we have these the next step is to extract these covariates.

So, there are 5 covariates just like previously, we are going to extract these covariates and then we are we want to combine these extracted covariates with the original data frame of 1342 observation and we are giving this name w dot dat which will be already created here and you can see here 1342 observation of 31 variables. Now, let us remove the sides with the missing covariates. So, where there are missing covariates we are removing those sides. Now, our data is now prepared for two stage model fitting and validation.

(Refer Slide Time: 19:23)





Now, we are going to demonstrate these two-stage model fitting for based on the A1 horizon. But remember that we can extend to any other horizons. So, you can perform this operation with any other horizon. Now, that in the data file 1341 observation so you can see these are considered as a numerical variable by R at this point of time, but we have to indicate R that these are the character variables because one indicates the presence of a proof of a horizon and 0 indicates the absence of that particular soil horizon.

So, we are going to first this converting we have to first convert this character variable into categorical variable. So, for that we are going we are going to use this x dot factor function for these A1. So, now it will understand R will understand that 1 and 0 are the two characters.

Now, again just like previously, we are going to set the random seed 123. We are going to now select 75 percent of the data, you can do it with the 70 percent data also. So, you can use the 75 percent of the data as the training samples and then you can use this data set and you can rename this data set as calibration data and of course, the validation data you can need a name as dat dot V. So, dat dot C and dat dot V are showing the calibration data and validation data. So, once we did, once we have separated this calibration and validation file.

(Refer Slide Time: 21:05)

```
76  
77  
78 #We first want to model the presence/absence in this case of th  
79 #We will use a multinomial model, then a stepwise regression pr  
80  
81 library(nnet)  
82 install.packages("MASS")  
83 library(MASS)  
84 library(aqp)  
85  
86  
87 # A1 presence or absence model  
88 mnl <- multinom(formula = A1 ~ AACN + Drainage.Index + Light.In  
89                 Gamma.Total.Count, data = dat.C)  
90 # stepwise variable selection  
91  
92 (The Level):
```

Environment: Global Environment
@dat.V: 333 obs. of 31 variables
@DSM_data: 1000 obs. of 8 variables
@ext: 1342 obs. of 6 variables
@HunterCovaria...: Large RasterStack (2902500 elements, 2...
@hvr.C5: List of 16
@hvr.MNLR: List of 26
@hvr.RF: List of 19
@hvrTerrorDat: Formal class 'SpatialPointsDataFrame'

Console:
> x.dat\$A1 <- as.factor(x.dat\$A1)
> # random subset
> set.seed(123)
> training <- sample(nrow(x.dat), 0.75 * nrow(x.dat))
> # calibration dataset
> dat.C <- x.dat[training,]
> # validation dataset
> dat.V <- x.dat[-training,]
> library(nnet)
> library(MASS)
> library(aqp)

Figure: A map of a region with a color scale from 0 to 200. A person's face is overlaid on the map.

```
76  
77  
78 #del the presence/absence in this case of the A1 horizon.  
79 #nomial model, then a stepwise regression procedure in order to  
80  
81 library(MASS)  
82 library(aqp)  
83  
84  
85  
86  
87 #ence model  
88 mnl = A1 ~ AACN + Drainage.Index + Light.Insolation + TWI +  
89 Gamma.Total.Count, data = dat.C]  
90 selection  
91  
92 (The Level):
```

Environment: Global Environment
@hvr.MNLR: List of 26
@hvr.RF: List of 19
@hvrTerrorDat: 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'
@mnl: List of 25

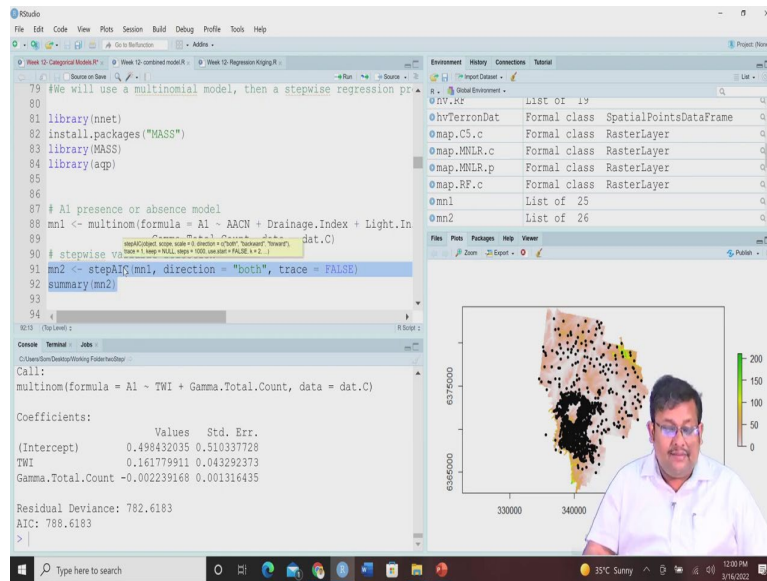
Console:
> library(MASS)
> library(aqp)
> mnl <- multinom(formula = A1 ~ AACN + Drainage.Index + Light.Inso
+ Gamma.Total.Count, data = dat.C)
weights: 7 (6 variable)
initial value 691.760886
iter 10 value 417.223876
iter 20 value 390.094046
final value 389.954432
converged

Figure: A map of a region with a color scale from 0 to 200. A person's face is overlaid on the map.

Now, suppose our first objective is to model the presence or absence in this case we are going to use this a A1 horizon. So, for this we are going to use the multinomial logistic regression model. So, we are going to use this nnet function and then we are going to install this mass package and then library aqp.

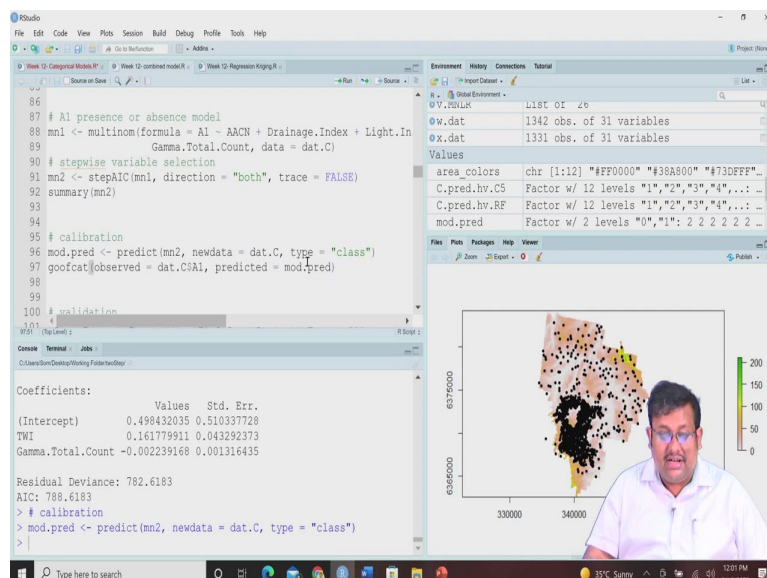
So, for predicting the presence or absence of the predicting the presence or absence of A1 horizon, we are going to use this multinom function and let us do this just like previously here our target is A1 presence of A1 or absence of A1 and here AACN, Drainage Index and then Light Insulation, TWI, Gamma Total Count all these are given here.

(Refer Slide Time: 22:15)



And then for the next step we are going to select the important variables using the stepwise regression. So, stepwise variable selection we are going to use step AIC function and then let us see what are the variables or features this algorithm can select. So, you can see here TWI and Gamma Total Count have been selected as important predictors and rest of them were discarded. So, will go with this mn2 model, which will be more parsimonious model than mn1, for subsequent statistical operations and mapping operations.

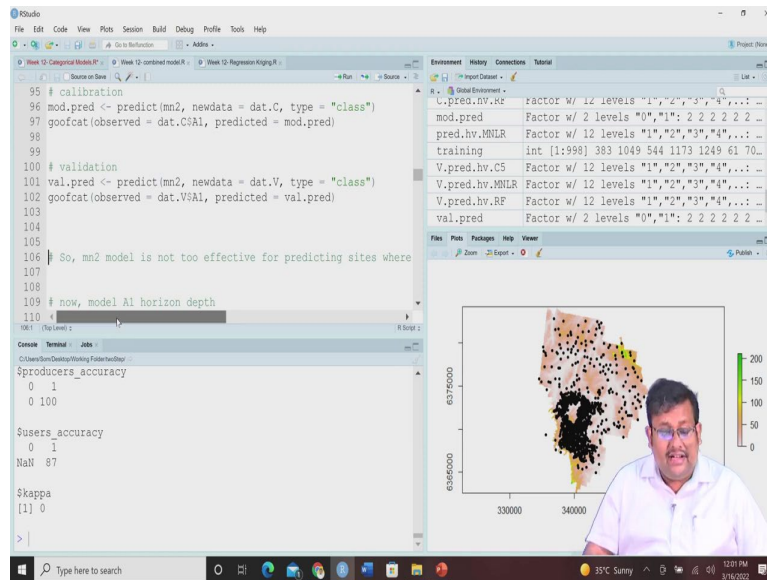
(Refer Slide Time: 22:50)



So, here you can see we are keeping these mn2 model and then we are going to predict the values for these training samples and we are going to predict the class using this calibration data using this mn2 model. So, we are doing this and then we are using this gofcat function

to produce the classification accuracy and you can see here again the confusion matrix overall accuracy produces accuracy users accuracy and Kappa coefficient are generated and then we are also in the next step we are doing the prediction of the validation samples.

(Refer Slide Time: 23:30)



This screenshot shows an RStudio session. The script editor contains the following code:

```
95 # calibration
96 mod.pred <- predict(mn2, newdata = dat.C, type = "class")
97 goofcat(observed = dat.C$A1, predicted = mod.pred)
98
99
100 # validation
101 val.pred <- predict(mn2, newdata = dat.V, type = "class")
102 goofcat(observed = dat.V$A1, predicted = val.pred)
103
104
105
106 # So, mn2 model is not too effective for predicting sites where
107
108
109 # now, model A1 horizon depth
110
111
```

The console output shows the following accuracy metrics:

```
$producers_accuracy
0 1
0 100

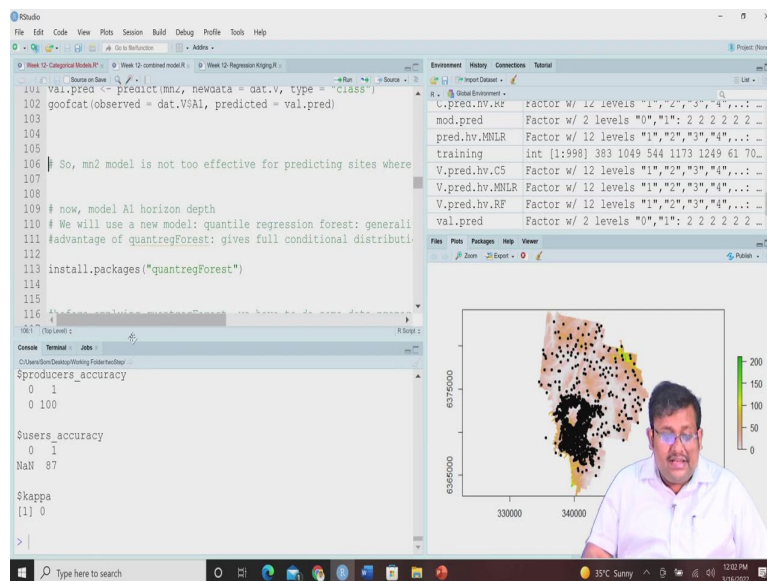
$users_accuracy
0 1
NaN 87

$kappa
[1] 0
```

The Environment pane shows the following objects:

Object	Class	Levels
C.pred.hv.RF	Factor w/ 12 levels	"1", "2", "3", "4", ...
mod.pred	Factor w/ 2 levels	"0", "1": 2 2 2 2 2 ...
pred.hv.MNLR	Factor w/ 12 levels	"1", "2", "3", "4", ...
training	int	[1:998] 383 1049 544 1173 1249 61 70...
V.pred.hv.CS	Factor w/ 12 levels	"1", "2", "3", "4", ...
V.pred.hv.MNLR	Factor w/ 12 levels	"1", "2", "3", "4", ...
V.pred.hv.RF	Factor w/ 12 levels	"1", "2", "3", "4", ...
val.pred	Factor w/ 2 levels	"0", "1": 2 2 2 2 2 ...

The plot pane shows a map of India with a color scale from 0 to 200. A video feed of a man is overlaid on the bottom right of the plot.



This screenshot shows an RStudio session. The script editor contains the following code:

```
101 val.pred <- predict(mn2, newdata = dat.V, type = "class")
102 goofcat(observed = dat.V$A1, predicted = val.pred)
103
104
105
106 # So, mn2 model is not too effective for predicting sites where
107
108
109 # now, model A1 horizon depth
110 # We will use a new model: quantile regression forest: general
111 # advantage of quantregForest: gives full conditional distributi
112
113 install.packages("quantregForest")
114
115
116
```

The console output shows the following accuracy metrics:

```
$producers_accuracy
0 1
0 100

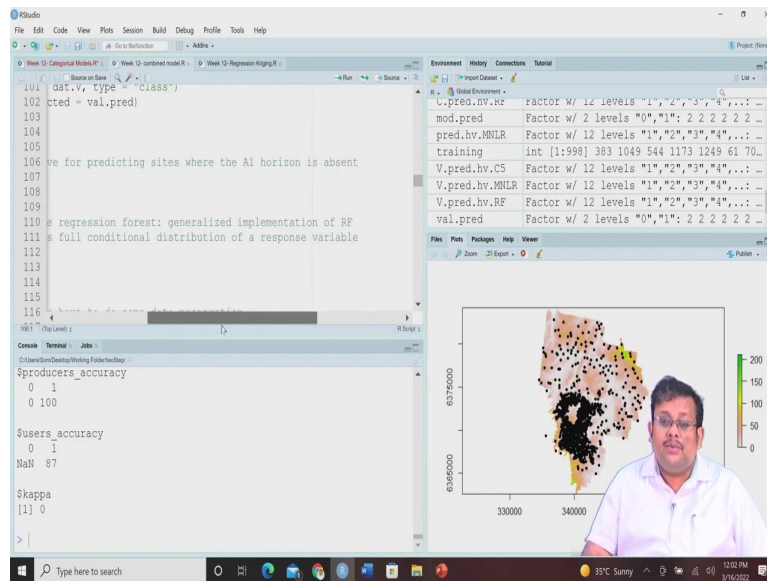
$users_accuracy
0 1
NaN 87

$kappa
[1] 0
```

The Environment pane shows the following objects:

Object	Class	Levels
C.pred.hv.RF	Factor w/ 12 levels	"1", "2", "3", "4", ...
mod.pred	Factor w/ 2 levels	"0", "1": 2 2 2 2 2 ...
pred.hv.MNLR	Factor w/ 12 levels	"1", "2", "3", "4", ...
training	int	[1:998] 383 1049 544 1173 1249 61 70...
V.pred.hv.CS	Factor w/ 12 levels	"1", "2", "3", "4", ...
V.pred.hv.MNLR	Factor w/ 12 levels	"1", "2", "3", "4", ...
V.pred.hv.RF	Factor w/ 12 levels	"1", "2", "3", "4", ...
val.pred	Factor w/ 2 levels	"0", "1": 2 2 2 2 2 ...

The plot pane shows a map of India with a color scale from 0 to 200. A video feed of a man is overlaid on the bottom right of the plot.



So, we are predicting the validation samples and we are again using this goofcat function to see that is also from this result, this is the confusion matrix overall accuracy produces accuracy, users accuracy and Kappa coefficient. So, we can see clearly from this that mn2 to model is not too effective for predicting the sides where even horizon is absent. So, here you can see when the A1 horizon is absent in both the cases these mn2 model is not very much effective for predicting those absence of A1 horizons.

Now, once we have done this categorical model, next step is to predict the A1 horizon depth. So, we will use a new model that is called quantile regression forest. It is generalised implementation of the random forests we know that random forest, how to use these random forests to predict the continuous properties, continuous variables.

However, we are going to now use the quantile regression for this time, we are going to show you the quantile regression for just to give you an idea about the the variety of different types of machine learning models which we use for digital soil mapping purposes. So, what is the major advantage of using these quantile regression forest. So, for basic advantage of this quantile regression forest is it gives the full conditional distribution of a response variable. So, I have already given these these annotations here so, you can follow these.

(Refer Slide Time: 25:15)

RStudio interface showing R code for quantregForest calibration. The code includes data preparation, model fitting, and validation steps. The console shows the output of the calibration process, including the removal of missing values and the fitting of the quantile regression forest model. A map of India is displayed in the background, with a video overlay of a person speaking.

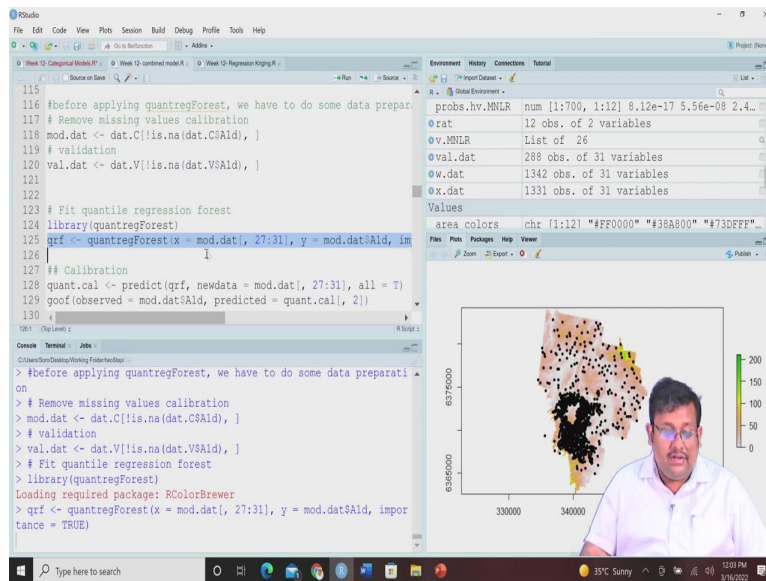
```
115  
116 #before applying quantregForest, we have to do some data prepar  
117 # Remove missing values calibration  
118 mod.dat <- dat.C[!is.na(dat.C$Aid), ]  
119 # validation  
120 val.dat <- dat.V[!is.na(dat.V$Aid), ]  
121  
122  
123 # Fit quantile regression forest  
124 library(quantregForest)  
125 qrf <- quantregForest(x = mod.dat[, 27:31], y = mod.dat$Aid, im  
126  
127 ## Calibration  
128 quant.cal <- predict(qrf, newdata = mod.dat[, 27:31], all = T)  
129 goof(observed = mod.dat$Aid, predicted = quant.cal[, 2])  
130  
131  
132  
133  
134 (No Level):  
R Script:  
NaN 87  
$kappa  
[1] 0  
> #before applying quantregForest, we have to do some data preparati  
> on  
> # Remove missing values calibration  
> mod.dat <- dat.C[!is.na(dat.C$Aid), ]  
> # validation  
> val.dat <- dat.V[!is.na(dat.V$Aid), ]  
>  
> # Fit quantile regression forest  
> library(quantregForest)
```

RStudio interface showing R code for quantregForest prediction. The code includes data preparation, model fitting, and validation steps. The console shows the output of the prediction process, including the removal of missing values and the fitting of the quantile regression forest model. A map of India is displayed in the background, with a video overlay of a person speaking.

```
115  
116 #before applying quantregForest, we have to do some data prepar  
117 # Remove missing values calibration  
118 mod.dat <- dat.C[!is.na(dat.C$Aid), ]  
119 # validation  
120 val.dat <- dat.V[!is.na(dat.V$Aid), ]  
121  
122  
123 # Fit quantile regression forest  
124 library(quantregForest)  
125 qrf <- quantregForest(x = mod.dat[, 27:31], y = mod.dat$Aid, im  
126  
127 ## Calibration  
128 quant.cal <- predict(qrf, newdata = mod.dat[, 27:31], all = T)  
129 goof(observed = mod.dat$Aid, predicted = quant.cal[, 2])  
130  
131  
132  
133  
134 (No Level):  
R Script:  
NaN 87  
$kappa  
[1] 0  
> #before applying quantregForest, we have to do some data preparati  
> on  
> # Remove missing values calibration  
> mod.dat <- dat.C[!is.na(dat.C$Aid), ]  
> # validation  
> val.dat <- dat.V[!is.na(dat.V$Aid), ]  
> # Fit quantile regression forest  
> library(quantregForest)  
Loading required package: RColorBrewer  
>
```

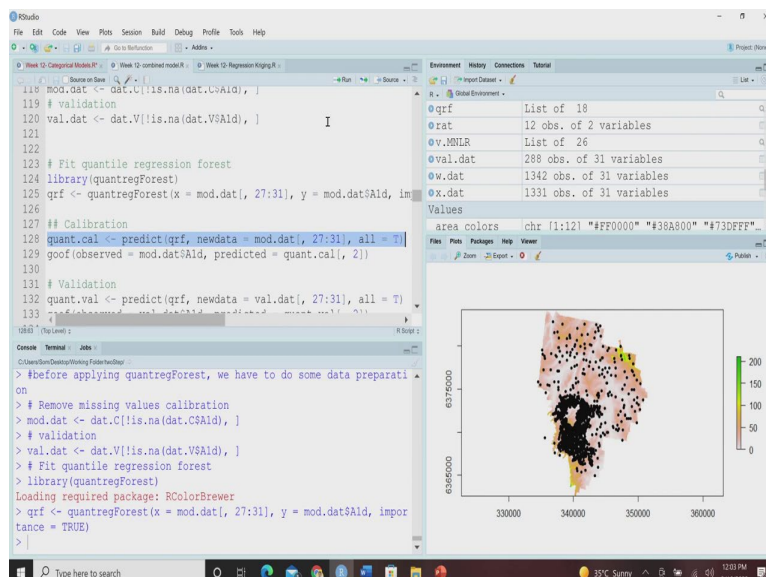
RStudio interface showing R code for quantregForest prediction. The code includes data preparation, model fitting, and validation steps. The console shows the output of the prediction process, including the removal of missing values and the fitting of the quantile regression forest model. A map of India is displayed in the background, with a video overlay of a person speaking.

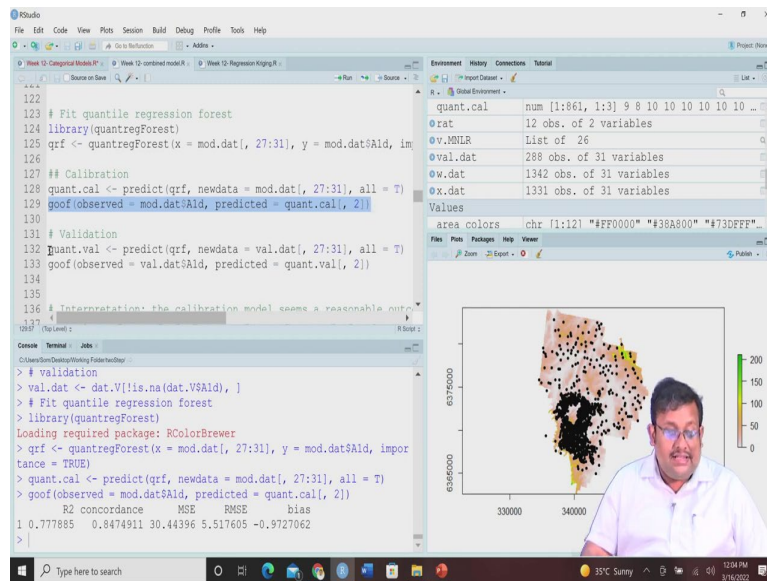
```
115  
116 #before applying quantregForest, we have to do some data preparati  
117 # calibration  
118 dat.C$Aid, ]  
119  
120 dat.V$Aid, ]  
121  
122  
123 forest  
124  
125 mod.dat[, 27:31], y = mod.dat$Aid, importance = TRUE)  
126  
127  
128 newdata = mod.dat[, 27:31], all = T)  
129 Aid, predicted = quant.cal[, 2])  
130  
131  
132  
133  
134 (No Level):  
R Script:  
[1] 0  
> #before applying quantregForest, we have to do some data preparati  
> on  
> # Remove missing values calibration  
> mod.dat <- dat.C[!is.na(dat.C$Aid), ]  
> # validation  
> val.dat <- dat.V[!is.na(dat.V$Aid), ]  
> # Fit quantile regression forest  
> library(quantregForest)  
Loading required package: RColorBrewer  
>
```



Now, for this we need to install this for this quantile regression forest we need to install this quant regression forest, and then before applying this quant regression forest, we have to do some data preparation. So, we are going to remove the missing data from the model data as well as from the validation data, and then we are going to call the library quant regression forest and then we are going to fit this quant regression forest model and then we are going to predict there, here you can see, our x is model data and our y is also model data with the depth of A1 horizon.

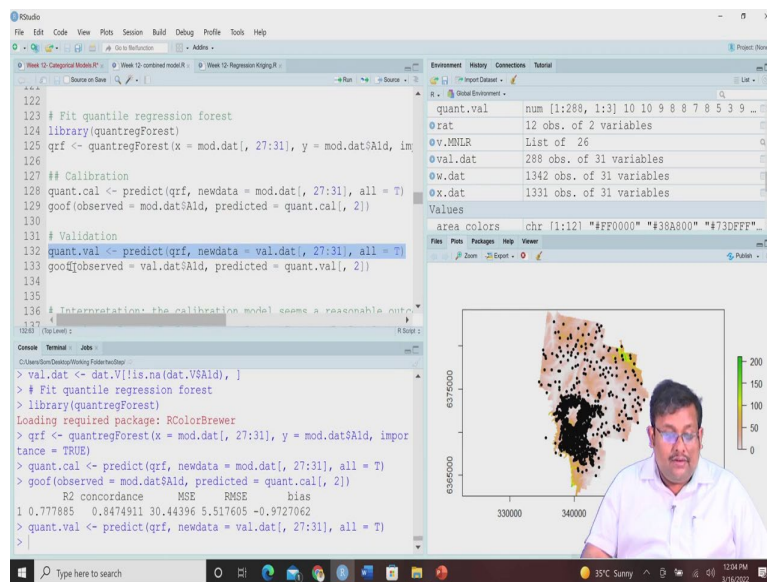
(Refer Slide Time: 25:59)

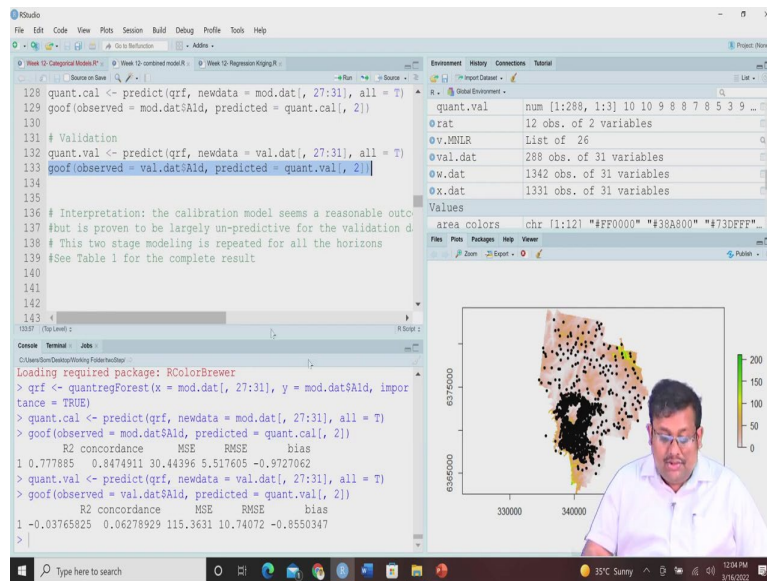




So, we are going to run it and let us see how it looks like. So, the model has been implemented. Now, we are going to calibrate based on our calibration data. So, our calibration data we have, we have predicted and now, we are going to use this goof function since it is a continuous model, we will be using the simple goof function and the results are as you can see here, 0.77 concordance 0.284.

(Refer Slide Time: 26:27)





Now, let us see that, what is the validation output? So, from this validation output, you can again we are going to use these goof function to see the results. So, we can see here from this that our interpretation should be the calibration model seems a reasonable outcome for predicting the depth of the A1 horizon. However, it is largely unpredicted for the validation samples. So, this is this two-stage modelling is then repeated for all the horizons, and will see table 1 for the complete result.

(Refer Slide Time: 27:08)

MODEL PERFORMANCE

Table 1: Selected model validation diagnostics returned for each horizon class and associated depth model.

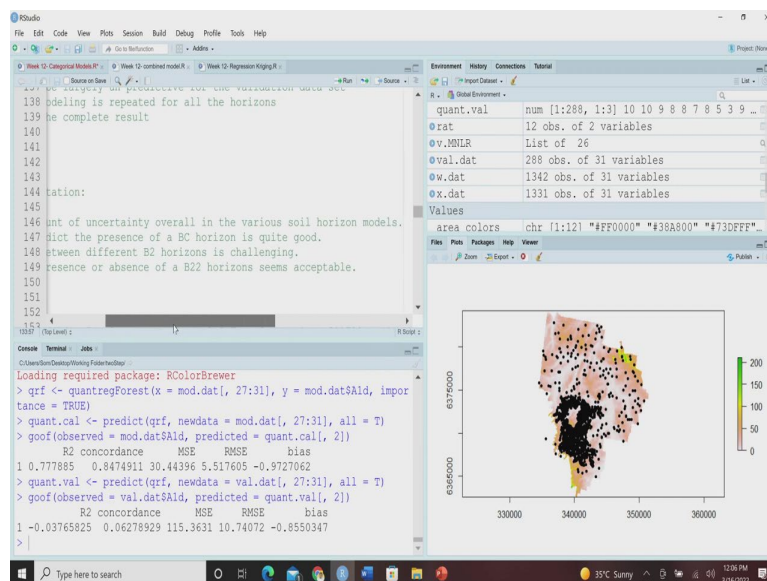
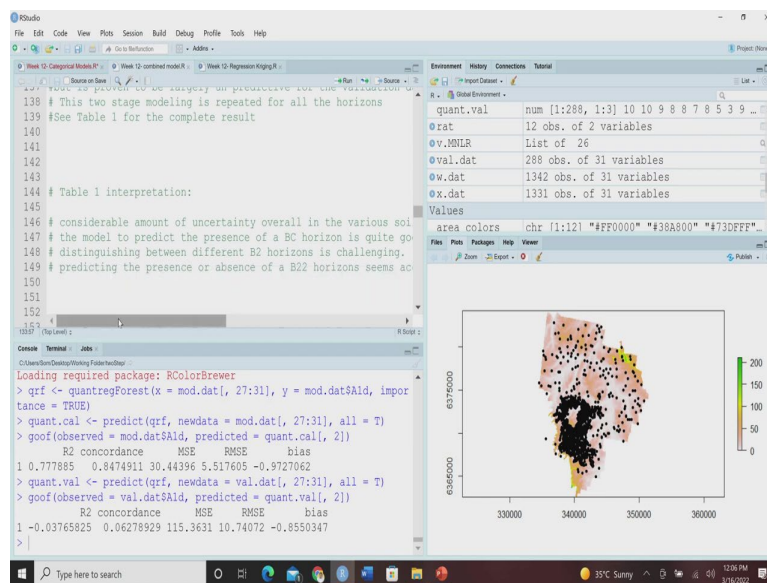
Horizon	Presence/Absence of Horizon			Depth of Horizon		
	Overall Accuracy	User's Accuracy	Kappa Statistic	Concordance	RMSE	PICP
A1	87%	Pres = 89% Abs = 54%	0.19	0.05	10	46%
A2	87%	Pres = 100% Abs = 87%	0.04	0.10	12	42%
AP	86%	Pres = 50% Abs = 88%	0.15	0.00	12	53%
B1	91%	Pres = 0% Abs = 91%	0	0.16	12	45%
B21	97%	Pres = 97% Abs = 0%	0	0.05	17	41%
B22	73%	Pres = 73% Abs = 34%	0	0.10	14	41%
B23	78%	Pres = 0% Abs = 78%	0	0.04	12	45%
B24	97%	Pres = 0% Abs = 97%	0	0.00	22	46%
BC	74%	Pres = 68% Abs = 75%	0.20	0.06	18	29%
C	95%	Pres = 0% Abs = 95%	0	0	NA	68%

So, if we go back to this table 1 which I have produced here, so, you can see here that this is the table 1 and these are the horizons, they are the presence and absence of the horizon you can see here and then there depth of the horizon is predicted here. So, from this table 1 we can infer we can see that some important observations that here you can see the presence is 0

percent absence is 91 percent, for B21 it is 97 0, B22 73 34, B23 0 to 78 and then B24 0 to 97.

So, you can see clearly that prediction of the presence of an absence of Horizon specifically in these horizons B2 horizons are not very accurate and it is very challenging not accurate, I would not say I would say it is very challenging at the same time the presence and absence of B22 Horizon prediction looks okay. But at the and also, if you see the Kappa statics, from this kappa statistics, we can see that presence and absence of Horizon BC horizon also looks okay. These are the concordance correlation coefficient and RMSE values.

(Refer Slide Time: 28:33)



The screenshot shows the RStudio interface. The console on the left contains the following R code:

```

144 # Table 1 interpretation:
145
146 # considerable amount of uncertainty overall in the various soil
147 # the model to predict the presence of a BC horizon is quite good
148 # distinguishing between different B2 horizons is challenging.
149 # predicting the presence or absence of a B22 horizons seems acceptable.
150
151
152
153 #Another way to assess the quality of the two-stage modeling is
154 #the number of soil profile that have matching sequences of soil
155 #we can do this using
156
157 vv.dat <- read.table(file = "validation_out.txt",
158                      sep = ",", header = T)
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
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 environment pane on the right shows the following objects:

- quant.val: num [1:288, 1:3] 10 10 9 8 8 7 8 5 3 9 ...
- o.rat: 12 obs. of 2 variables
- o.v.MNLR: List of 26
- o.val.dat: 288 obs. of 31 variables
- o.w.dat: 1342 obs. of 31 variables
- o.x.dat: 1331 obs. of 31 variables

The map on the right shows a spatial distribution of data points in a region, with a color scale ranging from 0 to 200. A person's face is visible in the bottom right corner of the map area.

The screenshot shows the RStudio interface. The console on the left contains the following R code:

```

144 etation:
145
146 amount of uncertainty overall in the various soil horizon models.
147 predict the presence of a BC horizon is quite good.
148 between different B2 horizons is challenging.
149 presence or absence of a B22 horizons seems acceptable.
150
151
152
153 assess the quality of the two-stage modeling is to assess first
154 all profile that have matching sequences of soil horizon types.
155 using
156
157 file = read.table(file = "validation_out.txt",
158                  sep = ",", header = T)
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
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 environment pane on the right shows the following objects:

- quant.val: num [1:288, 1:3] 10 10 9 8 8 7 8 5 3 9 ...
- o.rat: 12 obs. of 2 variables
- o.v.MNLR: List of 26
- o.val.dat: 288 obs. of 31 variables
- o.w.dat: 1342 obs. of 31 variables
- o.x.dat: 1331 obs. of 31 variables

The map on the right shows a spatial distribution of data points in a region, with a color scale ranging from 0 to 200. A person's face is visible in the bottom right corner of the map area.

So, let us go back to our script and see that this table 1 interpretation I have given here that considerable amount of uncertainty overall in the various soil horizon model we have already seen, the model to predict the presence of BC horizon is quite good because of the Kappa statistics distinguishing between the different B2 horizon is challenging we know that and predicting the presence or absence of a B22 horizon seems okay acceptable.

Now, another way to access or assess the quality of the two-stage modelling is basically to assess first the number of soil profile that have matching sequence of soil horizons types and we can do this by using different methods. So, let us wrap up our lecture here, and we will discuss this thing, this part in our upcoming lecture.

And we will see how to use the other methods to see the fidelity between the predicted profile as well as the observed profile whether how to how to get the matches how to get the match

between the observed profile as well as the predicted profile, what are the different ways we are going to see in our upcoming lecture. So, guys I hope that you have learned something new in this lecture and we will meet in our next lecture to discuss from here and we will discuss how to use this for producing the maps, thank you very much.