

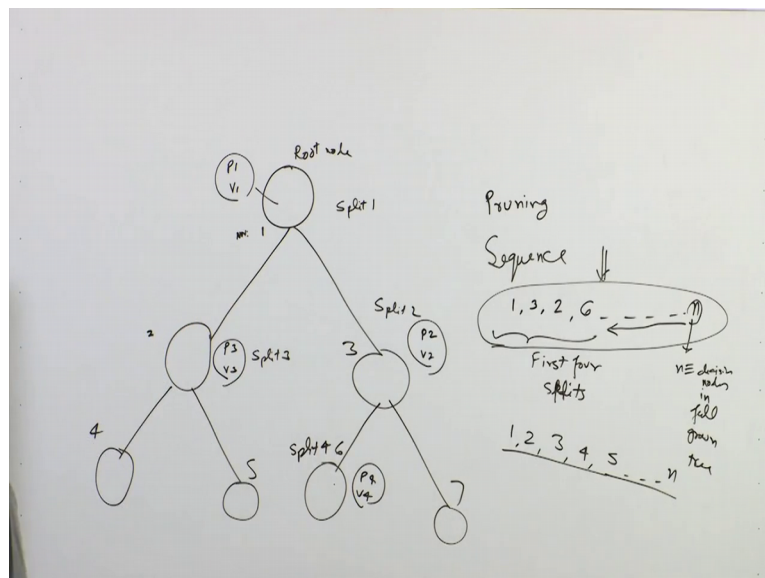
Business Analytics & Data Mining Modeling Using R
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture – 43
Pruning Process- Part II

Welcome to the course business analytics and data mining modeling using R. So, in the previous lecture, we were discussing classification trees, in particular, we were doing an exercise in R for the same. So, we did some modeling using the promotional offers data set. So, we talked about the way we did a modelling, there especially, the pruning part.

So, we were specifically focusing on the pruning part and there; when we try to prune back the full grown tree to a label where it does not over fit the data or fit the noise the way, we followed the pruning process that was you know a sequence of pruning was as per the node number ordering and it was not the nested sequence, right. So, we talked about a bit about this in previous lecture where we discussed that if this is our root node.

(Refer Slide Time: 01:16)



And in this root node we will have a predictor one and value one. So, predictor value combination based on which the split would be performed. So, some observation will fall in this part other observation will fall in this part. Similarly, for next split, we have to see that whether on this node or this node you know where the reduction performing you

know for these nodes where the optimum split (maximum reduction in impurity) is going to take place.

So, let us say; the next you know (maximum) impurity reduction happens in this particular node. So, let us say, this happens at variable P_2 and V_2 right. So, this is going to be about a split 1 this is split 2, then after the split is performed some observations will go to this side other observations will go to this part. Now, again for next split will have to check between these 3 which one you know which particular node and which particular predictor value combination will improve the impurity further, right improve the impurity the improvement (reduction) and impurity would be highest.

So, let us say now that here at this node the reduction in impurity is highest, then this is let us say the predictor value combination for the same is here. So, this is split 3 right now. So, here again we will have some observations that will go into this part some observations will go into this part right now for next split. Now, among these 4 nodes will have to check which one is giving the most reduction in impurity let us say this is the split this is the node and we have a $P_4 V_4$ and predictor value combination and it will be split 4.

So, the pruning sequence. So, from this we wanted to derive the pruning sequence. So, we look at the pruning sequence, it is going to be this node, right. So, if it is node number one. So, if we follow the unique you know node numbers that ordering that we discussed in the previous lecture this is going to be node number one this is going to be 2, this is going to be 3, then 4, 5, 6, 7.

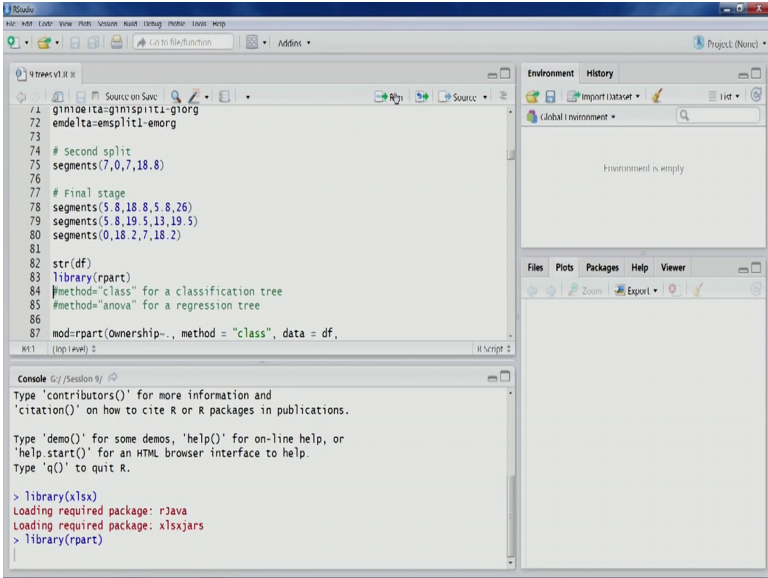
So, our pruning sequences first node number 1, then the second split happened at node number 3, then it happened at node number 2, then it happened at node number 6, right. So, 4 first 4 splits in this example, if we look at first 4 splits. So, they happen in this order. So, when we prune back the full grown tree to a certain level will have to follow this splitting pattern. right ah. So, let us say last know if there are n number of splits ah; that means, actually this is going to be n number of this is going to be equal to the decision nodes (decision number) of decision nodes in full grown tree.

So, therefore, we have to when we start pruning the full grown tree back to the desired levels will start deleting the you know least important splits; that means, splits which have done a least amount of reduction in impurity. So, probably we will start from here

and go our way back to the higher up to level. So, that we get to a point where the error on validation data is minimized. So, essentially the exercise that we had performed in the previous lecture the pruning that we had that we were following was based on this.

So, we just looked at the road node numbers and you know pruning was based on this. So, we are following the sequence in the increasing order as per the node numbers the optimal way of pruning that we want to follow is this one. So, today we will do an exercise in R, wherein, we will follow this particular pruning sequence and then let will understand few of the, you know few more points using a particular exercise in R. So, let us start. So, first let us load this particular package x plus x. So, let us go down. So, all these things we have already done.

(Refer Slide Time: 06:16)



```
71 gini_delta = gini_split1 - giniorg
72 em_delta = em_split1 - emorg
73
74 # Second split
75 segments(7, 0, 7, 18.8)
76
77 # Final stage
78 segments(5, 8, 18.8, 5, 8, 26)
79 segments(5, 8, 19.5, 13, 19.5)
80 segments(0, 18.2, 7, 18.2)
81
82 str(df)
83 library(rpart)
84 #method="class" for a classification tree
85 #method="anova" for a regression tree
86
87 mod = rpart(Ownership ~ ., method = "class", data = df,
88            (top1.pval) 2
```

Environment History

Global Environment

Environment is empty

Files Plots Packages Help Viewer

Console G:/Session 9/

Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

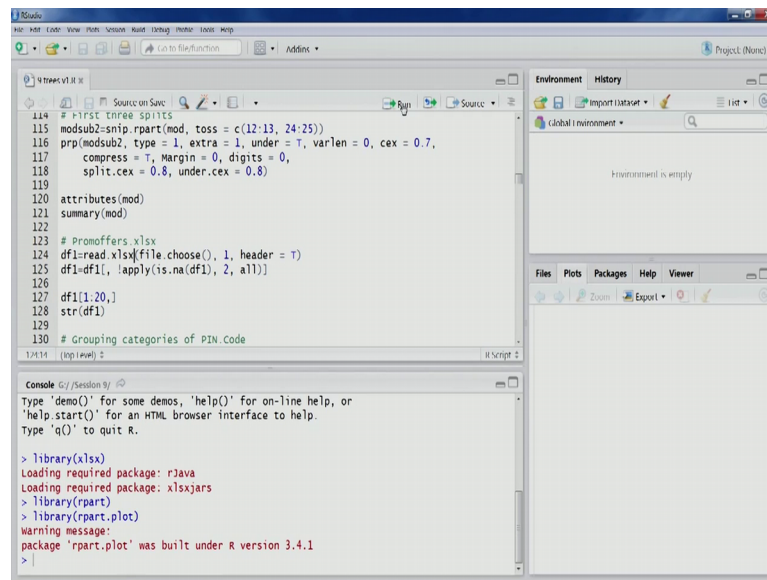
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

```
> library(xlsx)
Loading required package: rJava
Loading required package: xlsxjars
> library(rpart)
```

In previous lectures, let us load this program package as well we would be requiring this R part and one more package we would be requiring this one as well R part dot plot. Now let us move to our data set. So, promo offers dot x l s x is the file. So, we would like to import it here in R environment.

(Refer Slide Time: 06:36)



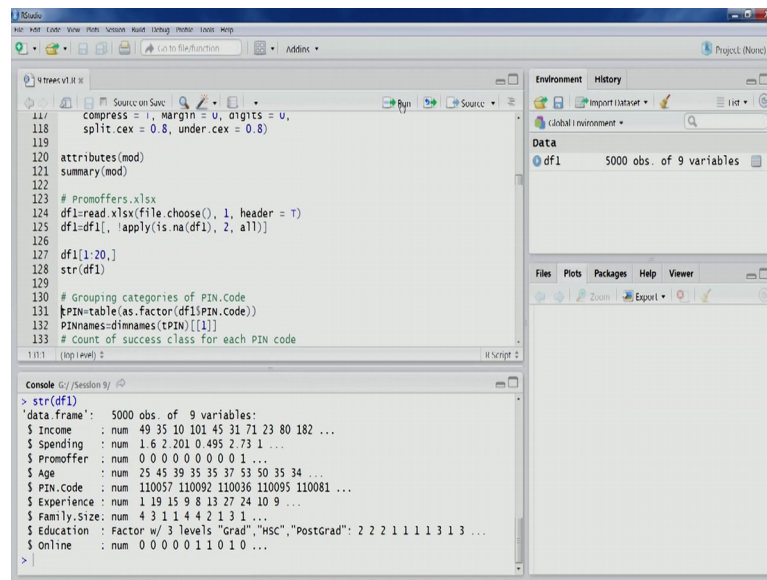
The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for data manipulation. The code includes comments and functions like `modsub2=snip.rpart(mod, toss = c(12:13, 24:25))`, `prp(modsub2, type = 1, extra = 1, under = T, varlen = 0, cex = 0.7, compress = T, Margin = 0, digits = 0, split.cex = 0.8, under.cex = 0.8)`, `attributes(mod)`, `summary(mod)`, `df1=read.xlsx(file.choose(), 1, header = T)`, `df1=df1[, !apply(is.na(df1), 2, all)]`, `df1[1:20,]`, `str(df1)`, and `# Grouping categories of PIN Code`.
- Environment:** Shows "Global Environment" with the message "Environment is empty".
- Console:** Displays the output of the code execution, including package loading messages: `> library(xlsx)`, `Loading required package: rJava`, `Loading required package: xlsxjars`, `> library(rpart)`, `> library(rpart.plot)`, and a warning message: `Warning message: package 'rpart.plot' was built under R version 3.4.1`.

So, let us perform this. So, it will take some time because it has this particular data set has 5000 observations. So, it will take slightly more time that we have been doing for other datasets smaller datasets.

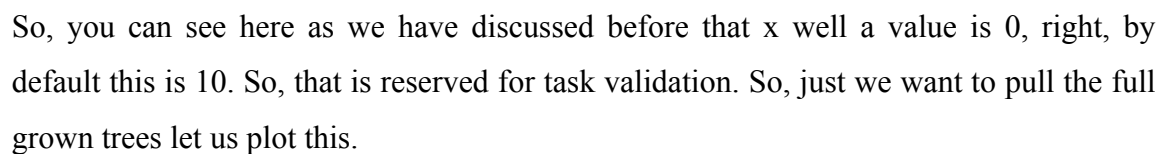
So, once this particular data set is loaded we will go through some of the steps that we had performed in the previous lecture and once that is done. So, once the pruning specific steps start, then we will discuss what we have covered here. So, you can see all the observation 5000 observations of 9 variables all of them are loaded in R environment. Now let us move any columns structure these are the variables.

(Refer Slide Time: 07:38)

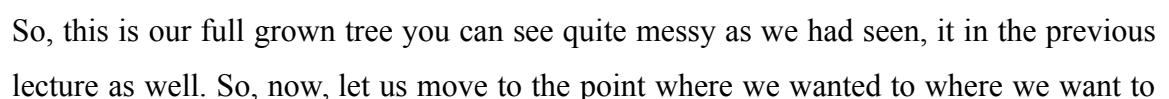


```
117 compress = 1, margin = u, digits = u,  
118 split.cex = 0.6, under.cex = 0.8)  
119  
120 attributes(mod)  
121 summary(mod)  
122  
123 # Promoters.xlsx  
124 df1=read.xlsx(file.choose(), 1, header = T)  
125 df1=df1[, !apply(is.na(df1), 2, all)]  
126  
127 df1[1:20,]  
128 str(df1)  
129  
130 # Grouping categories of PIN.Code  
131 tPIN=table(as.factor(df1$PIN.Code))  
132 PINnames=dimnames(tPIN)[[1]]  
133 # Count of success class for each PIN code  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
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  
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
```

(Refer Slide Time: 08:19)

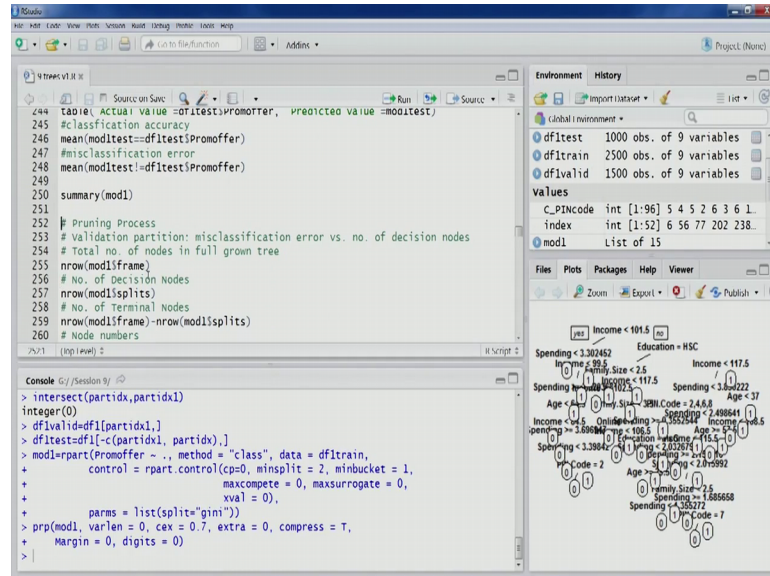


(Refer Slide Time: 08:44)



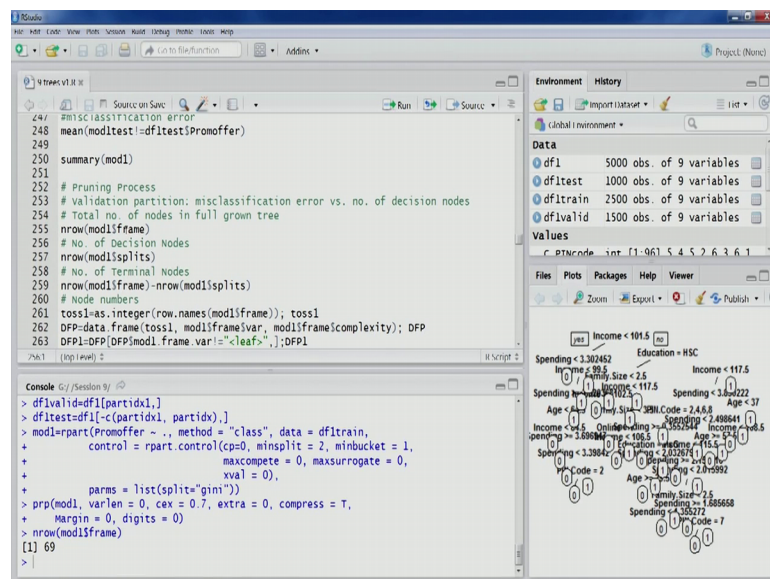
discuss further. So, the split variable and value combination this particular table we have already discussed we have gone through this.

(Refer Slide Time: 09:11)



So, we will not do this again performance of full grown tree we have gone through that. So, let us come back to the pruning process where as I discussed we followed a different pattern you know different pattern for pruning. Now we will follow the actual pattern the desired pattern for based on complexity.

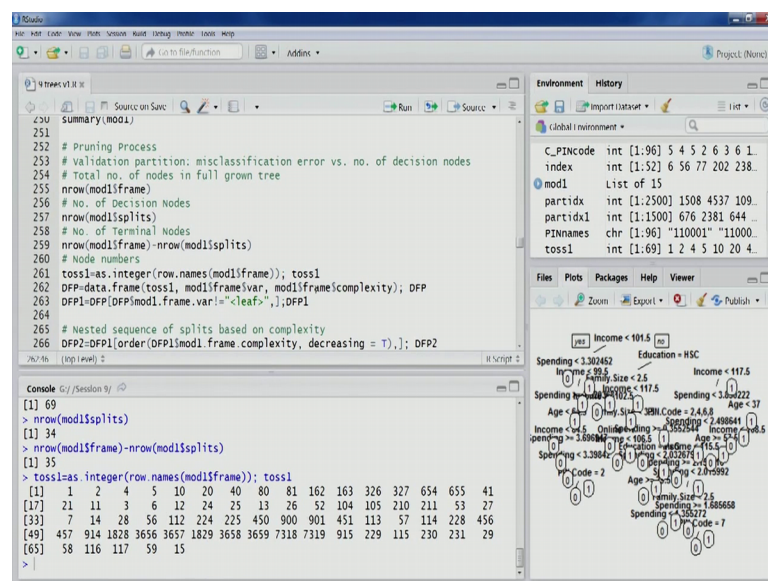
(Refer Slide Time: 08:36)



So, a pruning process let us look at the number of total nodes in this particular tree as you can see number of total node number of total nodes 69 and 34 in the 34 decision nodes and 35 terminal nodes. So, node numbering; so, you would see now certain steps that we had performed you will see differences now toss one is the argument that we want to compute at this point which we would be passing on to this snip r part function now toss one.

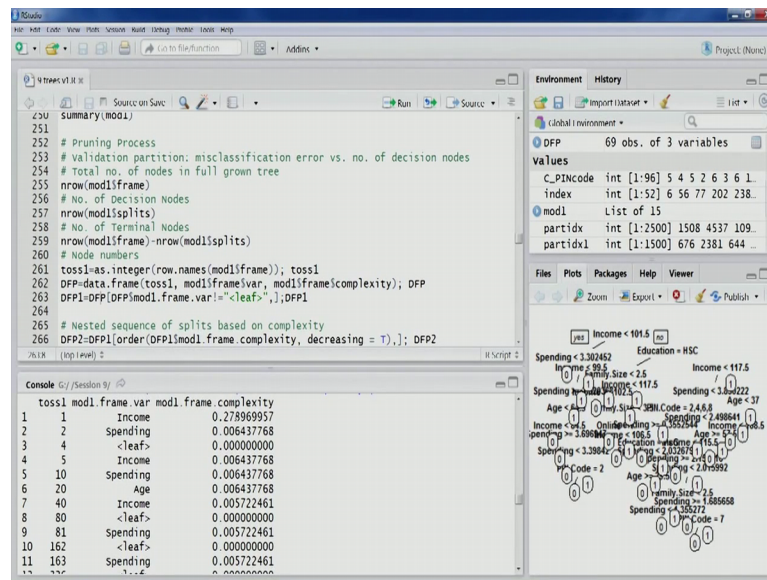
So, as we have discussed that r part object it has a frame attribute and within that frame attribute it has the row numbers. So, this we have discussed in previous lecture. So, we will get the row numbers will convert it into integer vector. So, that we will have the these numbers unique node numbers ah, but the ordering is not at for the desired order. So, now, we will constrain now we will create this data frame where we have the these node numbers in toss 1 and we will also have the variables write the variables involved at different nodes. So, whether the decision nodes are leaf nodes for leaf node it would just mention leaf as we have seen in tables in the previous lecture.

(Refer Slide Time: 11:54)



Now, for each node, we will also have complexity value which is also still stored in the frame attribute and within the frame we have this complexity variable. So, it would be stored there. So, let us create this data frame you can see here let us scroll through this particular data frame, as you can see first column is toss one which is nothing, but unique you know node numbering with respect to rows.

(Refer Slide Time: 12:54)



In decreasing order of complexity values right. So, the starting nodes from where the first split and then say onwards other splits happen. So, the starting nodes will have higher complexity values, right here the complexity value would be much higher that is why this was the split 1 number 1 here the complexity value would be after this.

So, that is why it was split number 2 followed by you know split number three and spirit number four. So, we would like to order this particular data frame by complexity values and once we order this particular data frame that we just saw by complexity values will also get the this sequence, right because this was the first split and the complexity value will be higher for this, right. So, this would be first then this was the second split complexity value for this particular node is going to be the second one after this. So, it will come here.

So, once we order this particular data frame which is having complexity values for each node we will get this. So, let us do this execute this code. So, you would see that before ordering first we are trying to remove the leaf node. So, we do not want to have a leaf node at this point you would see there are many leaf nodes here. So, we would like to remove the leaf node because the pruning is basically driven by the decision nodes, right. So, once we remove the leaf nodes from this data frame, we will get the new one this is the one were.

So, this is the one this is the new data frame that we can see DFP 1, we have 34 observation which is equal to the number of decision nodes that are there this is this these particular numbers, we have already seen in the previous output as you can see 34 decision nodes and 35 terminal nodes. So, once we remove the leaf nodes the number of you know observation that are there in the new data frame the 34 equal to the number of as you can see in the environment section 34 equal to the number of decision nodes.

Now once this is done we can order as we talked about we want to obtain the nested sequence of splits based on complex tree. So, we will order this data frame based on the complexity values and once we order this will get the desired nested sequence, right. So, let us execute this code now you would see that ordering has been done and if we scroll back to see this table now the first you know you know first is entry is income variable. So, this is the split number one and the complexity value is there the second split is also having the same complexity values, right we will discuss this further what happens if we the same complexity values are there, then why you know income was the first split and education was the second split ah. So, considering that what happens when this is the scenario?

So, family size and then third mode the third spirit is based on this having the third highest complexity value. So, in this fashion you can see that complexity values are decreasing. So, this is this is how our trees when we develop the full go grown tree. So, this is how this sequence determines how the splits are going to take place and how the tree is going to be built. So, once we start deciding about pruning you know pruning this full grown tree this is the process that we have to take and therefore, the earlier one that we did in previous lecture is not the desired process now you would see because we have sorted this particular data frame the row numbers have changed you can see these were the original row numbers we present in the original data frame DFP 1.

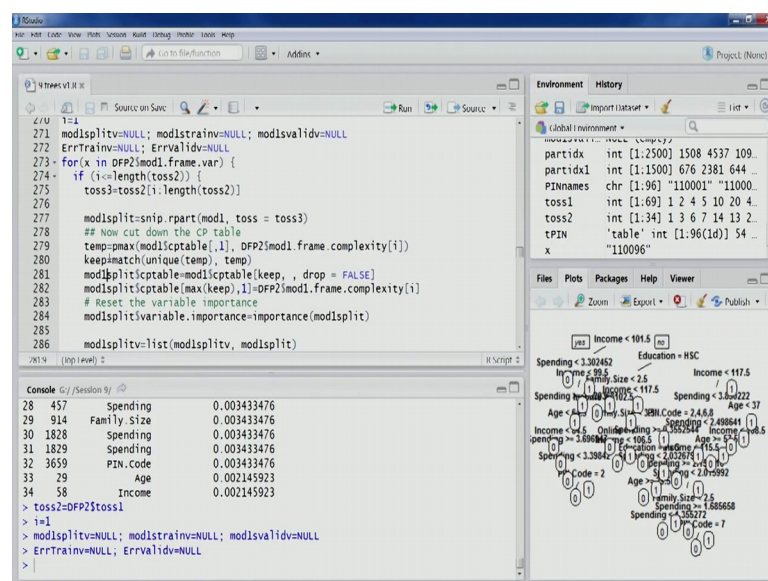
Now one sorting of that has been performed the row numbers are still same. So, we would like to change these row numbers to reflect the now DFP 2, let us look at the table like again you can see now the row numbers row numbers are also sorted. So, 1 to 34; 34 decision nodes, right. So, once this is done, now we can start calculating our toss argument that we have to pass on to the snip dot r part function. So, toss the 2 argument can is simple nothing, but in the data frame that we have just you know created the P 2; the first you know variable toss one that is going to be this argument.

So, let us create this toss two now what we are going to do is we are going to start our pruning process and as we did in the last lecture after every pruning, we used to record the model and we used to apply that model to a score on training, you know partition and other partition validation another part validation partition training and validation partition. So, that later on we can compare the error rates right.

So, the same thing will follow here what we did in the last lecture, but now this time with the actual pruning sequence the nested training sequence. So, counter for nodes to be sniffed off I and once, then we have this mod one split v the same wherever that we use in the last lecture this is going to you know store all the mod variables then you know mod 1 train v, these variables are going to store the other things that we will see this course right mod one mod one train. So, it will its score it will have the that return value of predictor.

So, it is scored variable right list ah. So, let us initialize them, then we will have these two vector 0 train v and other valid v. So, these two actors are the important for our plotting and to identify where the error on validation partition is minimizing. So, let us initialize these two variables now you can see as we discussed in the previous lecture the loop is running for all the variables. So, the in the this in this particular case you can see we are running this loop for all the decision nodes that are there right DFP 2 in this particular column.

(Refer Slide Time: 19:34)

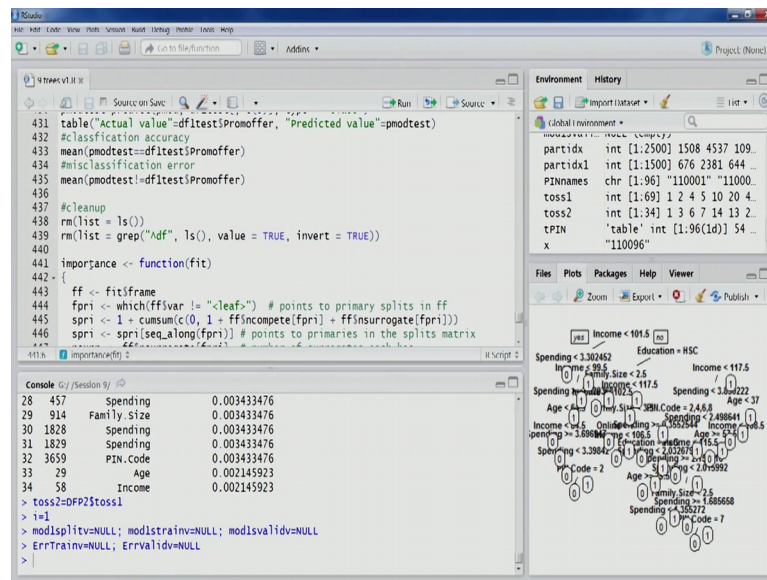


Now we have only the decision node you can see again that in environment section DFP 2 has just 34 observation that are the number of decision nodes in this particular tree. So, we will run this loop for the number of decision nodes. So, some of the checks that we had done in the previous lecture; that code that we were eliminating the leaf now we do not need to perform because we are dealing with only decision nodes. So, the if section, you would see that I; we are comparing with the length of the toss two that is the total number of nodes and then because we would be pruning a node by node. So, we are starting this protocol process from i that is one to the full to the final node number that is the last one.

So, first we start by you know pruning all nodes, then you know from node number 2 do the last one node number 2, in the sequence not node number 2 actually node number 2 in sequence as a stored in toss 2. So, to show you the toss 2 values you can see toss 2 1 to 34. So, 136. So, node number are actually unique node numbers are 136. So, first we start by you know first we start by sniffing all the nodes, then we start by sniffing from this particular node to the remaining nodes then we start from this particular node that is 6 to the remaining nodes in this fashion we will start and then a snip dot for part function is being called for the for every time the loop is run and you are recording a few more we are correcting few more things.

For example, CP table; once we create and you know once we do this sniffing, we will get the new model new sub tree model. So, therefore, we need to correct the CP table and there. So, the code for the same is there then one CP table is corrected will also have to correct the variable importance code for that is also there right. So, for this we are using an importance function which is nothing, but taken from the source code of you know prune dot r part function. So, there they have written this importance function.

(Refer Slide Time: 21:53)



And we are directly using the same source code here in this our exercise because this particular function is not available for us to you know call you know is not part of the r part library, once we load they are they do not have access to this function this is called internally within r part. So, that is why we have to get that source code here and to be able and then we are using this particular here.

So, we will have to now create this function here. So, that will do here. So, the you and you would see that in the environment section function this importance function has been created. So, we will not go into the detail of this particular function this function is actually being called to once we create the sub tree model, we would like to we would like to change the variable importance accordingly right in the sub tree model. So, the same thing is being done by calling this function. So, then we are storing we keep on storing these you know; all these all these sub tree models then we score them off score the training partition and the validation partition as we did in the last lecture and then we are storing the error rates for training partition and validation partition and this is the entry look counter.

So, let us execute this code. So, it is done. Now you would see that one thing I would like to point out here that those we have been storing the models, but we cannot access all of them you can see this is quite large list and given 3 MB and just having two elements. So, these this R part object that we were trying to store in list there you know

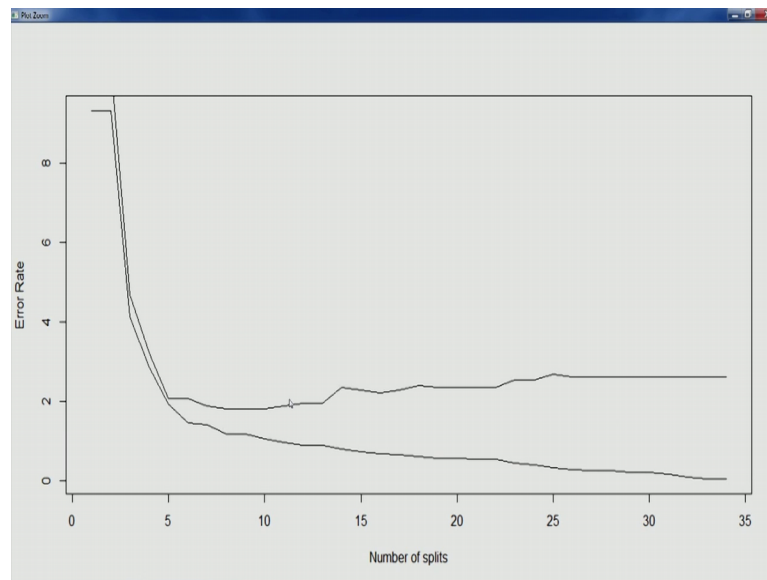
the size is quite big. So, therefore, it has not stored all the all the you know all the R part or model sub tree models and therefore, only two are there. So, ah, but; however, we are interested in only the error rates.

So, let us create this data frame like we did in the previous lecture. So, now, let us look at these values 4 for decision nodes in this ordering sequence and you can see either training and validation. So, now, when the for the first decision node this is the training error and validation error you can see that training error is slightly lower than the validations you know error when we start and as we perform second split then again the both are same.

So, there is not no not much decrease in error after second split then third you would see that further the error has significantly decreased for the training as well as for the validation. So, in this fashion if you as we did in the last lecture if you scroll down this these are rates the second column that is error rate for the training part it will keep on decreasing till it becomes 0, right till it become 0 or close to 0 right and in the in the validation partition you would see that error will keep on decreasing till one point and after that it will start in you know increasing.

So, you can see that this is this is the point where the error is minimum, right. So, this is the point where the error is minimum and then after this particular point it will it will hold up to for some more nodes and then it will start increasing that it keeps on increasing. So, with this now we can go to you know we can also we can create this plot to visualize the same information then information that we saw in table.

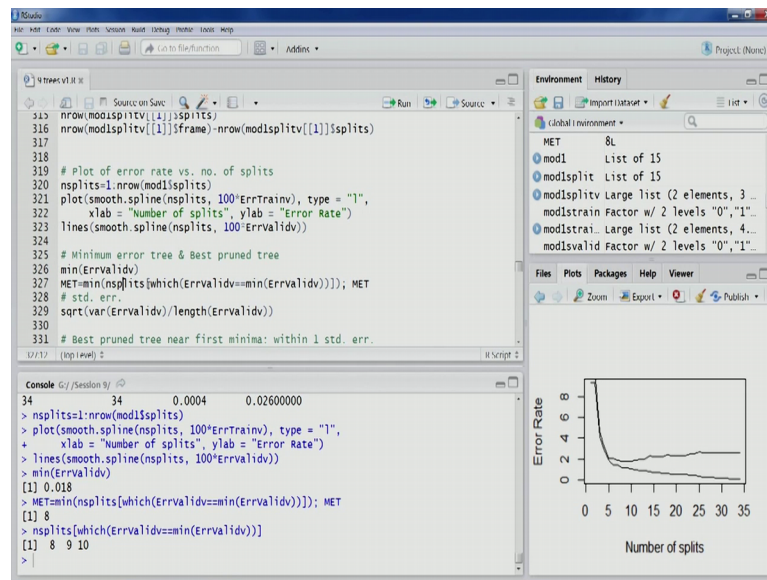
(Refer Slide Time: 25:45)



So, this is the plot that we had seen in previous lecture as well now with the correct pruning sequence you can see that the plot which is this plot this is you know the this particular is for the validation data the lower plot the upper plot is for the validation data and the lower a plot is for the training partition. So, for the training partition you would see that the error you know keeps on decreasing till it becomes 0 for the validation part you would see the error keeps on decreasing up to some point and after that it will start you know it will start increasing right.

So, probably here we need to in this particular zone we need to find out the point with minimum a territory like we did in the last lecture and then within one standard deviation we will have to find out the best tree. So, let us look at this value minimum error tree is this, this is the value which we already saw in the table then let us look at the particular number of decision nodes corresponding to this error value error 8 decision nodes minimum tree is can we obtain at 8 decision nodes if we look at the graph again. So, 8 decision node would be somewhere around here. So, probably this is this particular straight link straightening line you see. So, all these you know nodes they are nothing, but representing the, you know minimum error on validation partition. So, whether they are 8, 9 or 10.

(Refer Slide Time: 27:16)



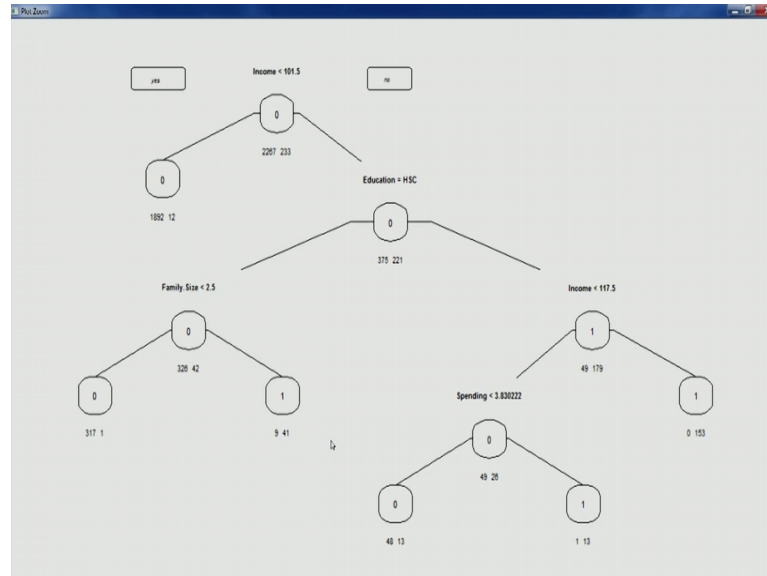
So, we can look at these values, if you are interested how many how many of these decision nodes are having the same or having the same number of same error minimum error 8, 9 and 10. So, the sub tree models with decision nodes 8 decision nodes and 9 decision nodes and 10 decision nodes; all three of them having the same number of same amount same amount of error on validation partition, but; however, we will just use the smallest tree here and then we look at the standard error of you know off error rate.

So, this is the value; now we will look at the range where we need to find the now best prune tree. So, the best from tree should be having value less than this particular value error like we did in the last lecture and should be greater than the error that we saw for the this one minimum error tree, all right. So, this is the code for the same. So, this part we have already discussed. So, you can see best prune tree is now coming at 5. So, if you want to confirm this, we can go back to the level table and we can see that node 8, this is the point where the minimum error tree is there, now within that range, we can see this particular tree is giving us the best prune trees this is within one standard deviation of minimum error tree. So, this part we have already discussed.

Now, once this is done once we have identified then we can go ahead and create our min best prune tree model. So, this is how again BPT we would like to contain we would like to contain these many number of design nodes. So, we can generate about toss three and

then call this sniff R part and we will have the best prune tree. Now let us plot it. So, this is our best prune tree let us look at this particular plot.

(Refer Slide Time: 29:21)



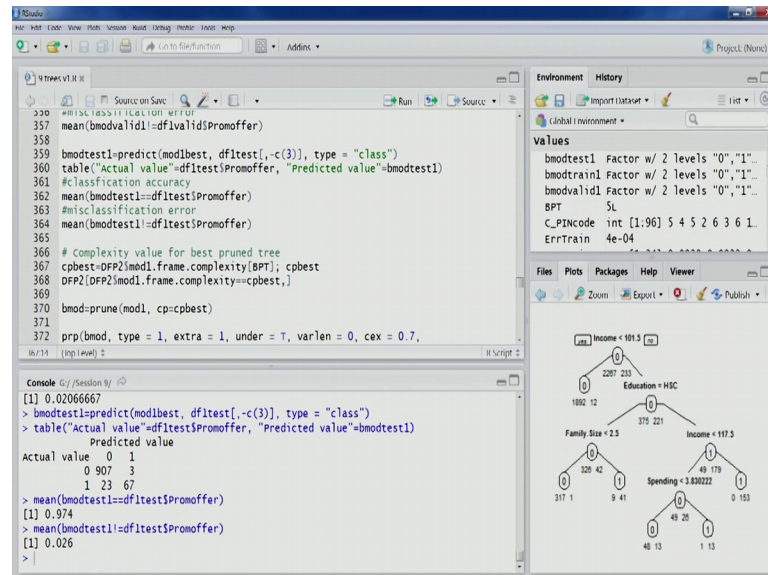
Now this particular best prune tree now the earlier one which we did in the previous lecture because we are following the order of you know shorter order of a node numbering. So, we were getting the balance tree. Now, we get the right tree would see that this is not balanced first income then education and then that sequence the it is it split sequence the optimized split sequence is being followed in this particular example, right. So, this is the best prune tree that we can have you can see 1, 2, 3, 4, 5 decision nodes are there you can see important variables of course, it is income education. So, income education families are spending.

So, all of them figuring here; now we can check the performance of this particular tree on different partitions; so, you can see the performance 98.56, then on validation 90.9 close number, then on test 97.4, this is also close. So, performance is quite good. So, there could be another approach to follow this process that we discuss in the previous lecture as well based on complex tree value.

So, we use the complex tree value for example, we have identified the best prune tree now following the you know actual order that is the split order. So, in that we can find the appropriate you know complexity value because we have this pruned function this which we use in the previous lecture which takes the CP value and cuts the prunes the

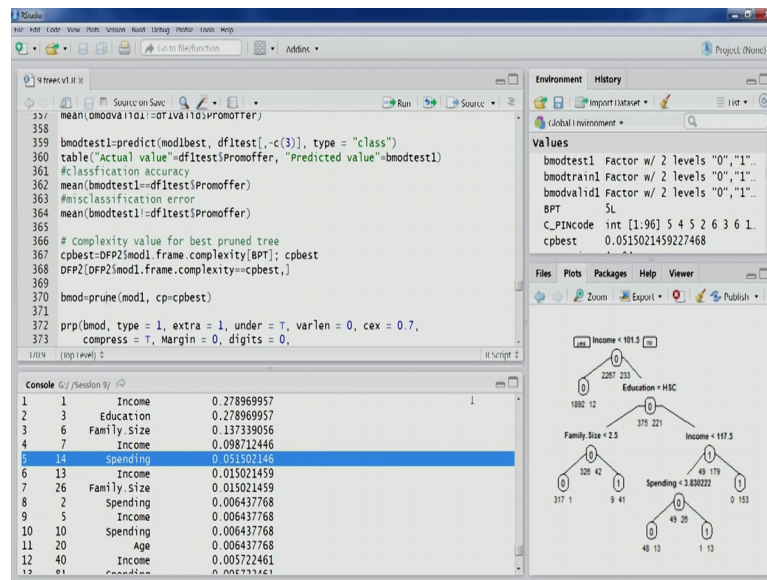
tree inter based on that CP value; however, we will understand some of the problems with this particular function.

(Refer Slide Time: 31:08)



For example, let us find out the complexity value for the best prune tree that we have just identified which was the tree with 5 decision nodes. So, CP best is this is the corresponding complexity value and this you can see, we you can see toss 1 is 15, right and so, this will using this particular value. So, we can go back to the table and find out how many number of nodes are there here ah. So, let us look at let us look at that table. So, if we look at the value that we just saw their 0.0515. So, you can see this is the value 0.0515 and we can see that toss 1.

(Refer Slide Time: 31:55)



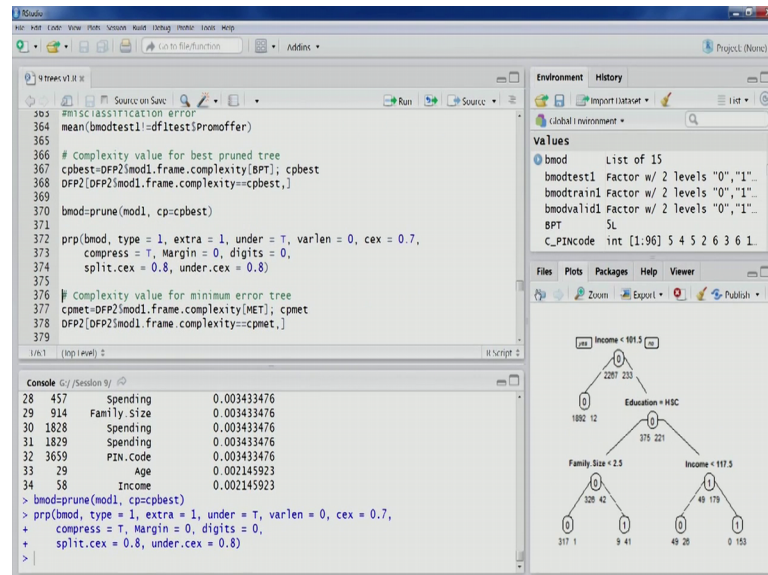
So, 1, 2, 3, 4, 5; so, this is also 5. So, the same you know corresponding tree is there, but; however, it might. So, happen that. So, now, we are discussing the problems that would be there with the prune function now the previous few values sometimes if we run the same model previous few values might also have the same complexity value in that case the tree with the smaller size would be selected by in this fashion. So, if we do the you know pruning using the complexity values, even though we have identified you know followed that passes minimum error tree and within one standard deviation best prune tree.

And now instead of the number of decision nodes we use the complexity value to prune this tree you know the previous you know nodes they also had the not in you know they also had the same complexity value. So, the pruning will happen will happen at that level. So, it is might with the tree size might reduce from 5 to 3 or 2 something in some scenarios in some runs and even in this data itself we do again the same thing, we do it again, then probably because of the sampling and the different observation that are going to be selected in the training partition and therefore, the different model that could be there because of the limitation on the sample size that we have even though this is larger data set.

So, we can get different prune tree using this particular prune function. So, in this particular case it comes out to be the same. So, we can use prune function, we pass on

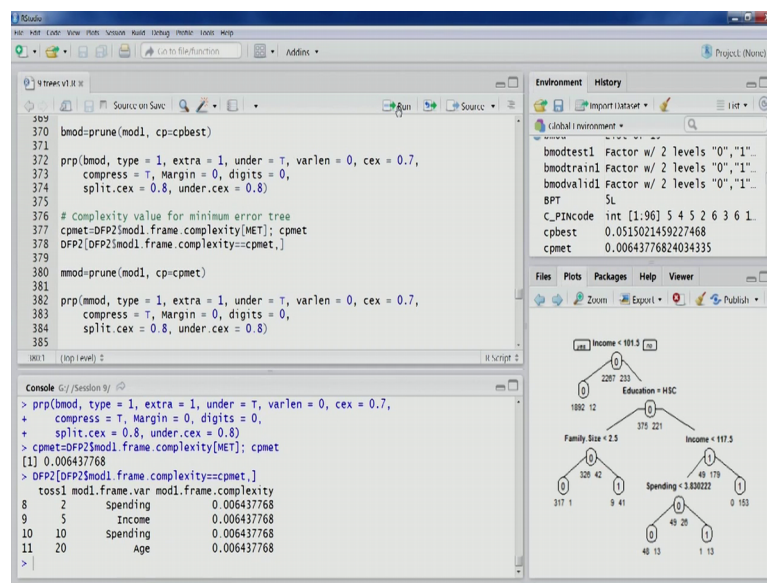
the full grown tree model mod one and then the pruning value till the point where we would like to prune it. So, we can see this.

(Refer Slide Time: 33:43)



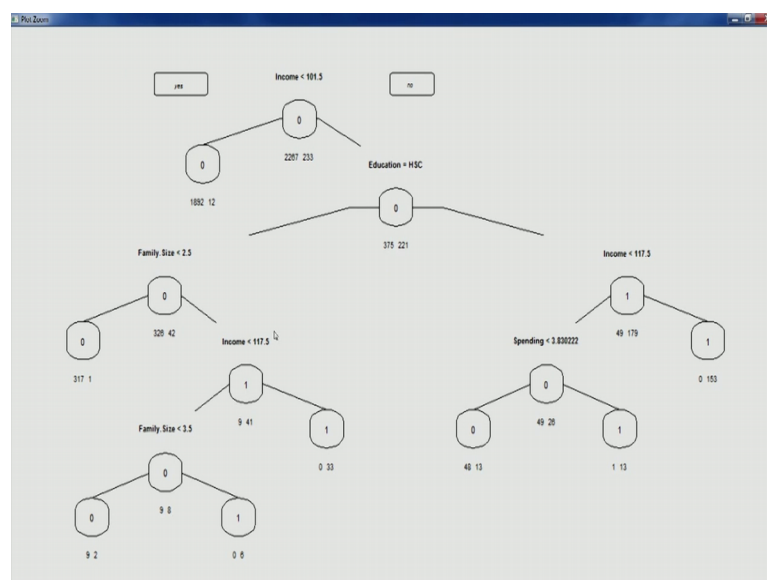
So, this particular tree; 1, 2, 3; you can see 4 nodes are there and here we had 5 nodes. Now in this particular internal processing that happens in prune function one more node they spend they spending one it has been removed off. So, that is the tree that we will have if we follow that complexity value right. So, the tree will collapse at that value collapse at this value right and only 4 particular decision nodes would be there. Now further we can we can compare this particular case with the minimum error tree. So, we can plot the minimum error tree as well.

(Refer Slide Time: 34:30)



So, this is going to be the corresponding complexity value. So, this is up following the prune process prune function process. So, you would see just. So, these are the nodes you can see this is the value. So, we look at the, we prune it. So, this is the model that we get. So, you can see minimum error tree model is much bigger, even if we follow the prune function right.

(Refer Slide Time: 34:56)



You can see 1, 2, 3, 4, 5, 6, 7 nodes; there right. So, we saw that size when the prune sequence the last one is also removed off. So, we get the 8 size, if you are interested in

looking at other things; for example, CP table and other things. So, this per got aspect, we have already discussed, right. So, with this we stop here and in the next lecture, we will start our discussion on regression trees.

Thank you.