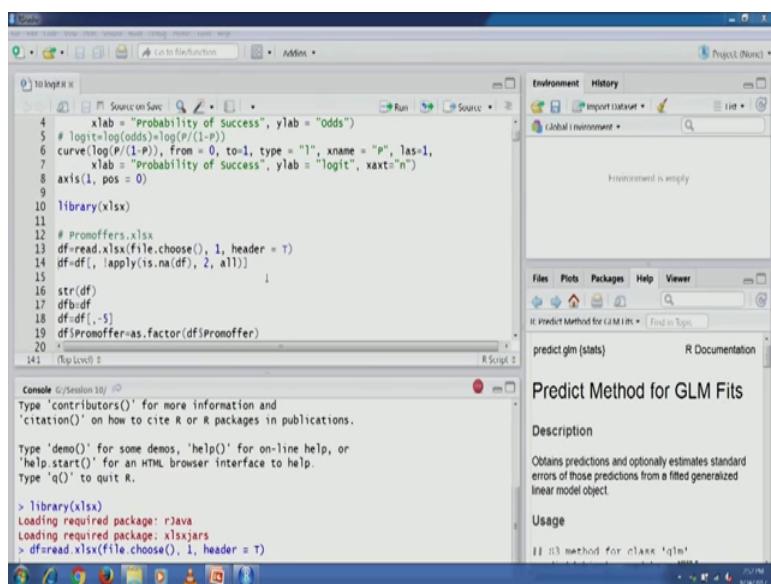


Business Analytics & Data Mining Modeling Using R
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture - 49
Logistic Regression - Part IV

Welcome to the course business analytics and data mining modeling using R. So, in previous a few lectures, we have been discussing logistic regression and in a previous lecture, specifically, we talked about how we can actually interpret a Logit model, Odds model and also probability based model, we also understood the differences in terms of interpretation . So, let us in this particular lecture let us start with our exercise in R that we have been doing. So, we have been using this promotional offers data set. So, we would like to complete this particular exercise.

(Refer Slide Time: 01:00)



So, let us load this protocol library xlsx. So, promotional offers data set that we are already familiar with 5000 observations. So, let us a load it into R environment.

So, a in a previous lecture, we have been able to build the model and we also understood the results and interpreted the results that we got in our promotional offers model. Now we will check the performance of this particular model on test partition and also will for the training partition as well we will look at some of the charts like cumulative lift curve and also design chart for this particular data set. So, as you can see now observations

have been loaded into environment section you can see this 5000 observation; let us remove NA columns let us look at the structure once again.

(Refer Slide Time: 02:03)

The screenshot shows the RStudio interface. The left pane displays the R script 'logit.R' with the following code:

```

1 library(gmodels)
2 library(gridExtra)
3 library(ggplot2)
4 library(caret)
5 library(e1071)
6 library(nnet)
7 xlab = "probability of Success", ylab = "logit", xaxt="n")
8 axis(1, pos = 0)
9
10 library(xlsx)
11
12 # Promoffers.xlsx
13 df<-read.xlsx(file.choose(), 1, header = T)
14 df<-df[, !apply(is.na(df), 2, all)]
15
16 str(df)
17 df$df<-df[, -5]
18 df$Promoffer<-as.factor(df$Promoffer)
19 df$online<-as.factor(df$online)
20 str(df)
21
22
23

```

The right pane shows the 'Environment' tab with two data frames: 'df' (5000 obs. of 8 variables) and 'dfb' (5000 obs. of 9 variables). The 'Console' tab shows the output of the 'str(df)' command, listing variables like Income, Spending, Promoffer, Age, PIN.Code, Experience, Family.size, Education, and Online.

So, all the familiar variables, right.

So, let us take a backup of this particular data frame and we are as we talked about in previous lecture as well, we are not interested in this particular variable pin code and many categories, right. So, we would like we would not like to consider this particular variable in this model. So, let us get it off get rid of this particular column. Now we are left with a promote to categorical variable promotional offers and online activities online activities whether a customer whether a particular individual is active online or not and the promotional offer is our outcome variable of interest whether the customer accepts the offer on or not.

So, let us convert them a to factor variable categorical variable now these are the variables that you would like to take forward for our modeling exercise income expanding promotional offer and then age experience family size education and then online.

(Refer Slide Time: 03:00)

The screenshot shows the RStudio interface. In the top-left pane, there is R code for reading a file and partitioning the data into training and testing sets. In the bottom-left pane, the R console shows the execution of the code. On the right side, there is a help window for the 'predict.glm' function, which is used for obtaining predictions from a fitted generalized linear model object.

```
11 # Promoffers.xlsx
12 df<-read.xlsx(file.choose(), 1, header = T)
13 df<-df[, !apply(is.na(df), 2, all)]
14 str(df)
15 df<-df[-5]
16 df$Promoffer<-as.factor(df$Promoffer)
17 df$Online<-as.factor(df$Online)
18 str(df)
19 # Partitioning: Tr:Te->60%:40%
20 partidx<-sample(1:nrow(df), 0.6*nrow(df), replace = F)
21 dfrtrain<-df[partidx,]
22 dftest<-df[-partidx,]
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
```

Console Session 10/1:

```
> df$Online<-as.factor(df$Online)
> str(df)
'data.frame': 5000 obs. of 8 variables:
 $ Income : num 49 35 10 101 45 31 71 23 80 182 ...
 $ Spending : num 1.6 2.201 0.495 2.73 1 ...
 $ Promoffer : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ Age : num 25 45 39 35 35 37 53 50 35 34 ...
 $ Experience : num 1.19 15 9.8 13 27 24 10 9 ...
 $ Family.Size: num 4 3 1 1 4 4 2 1 3 1 ...
 $ Education : Factor w/ 3 levels "Grad","HSC","PostGrad": 2 2 2 1 1 1 1 3 1 3 ...
 $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...
```

R Documentation

Predict Method for GLM Fits

Description

Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object.

Usage

```
## S3 method for class 'glm'
predict.glm(..., newdata = NULL, ...)
```

So, we followed a 60 percent, 40 percent partitioning in previous lecture as well. So, let us do the partitioning.

(Refer Slide Time: 03:20)

The screenshot shows the RStudio interface. The code has been updated to include a model fit for the 'Income' predictor. The environment section now shows 3000 observations for the training set ('dfrtrain') and 2000 observations for the test set ('partidx').

```
15
16 str(df)
17 df<-df
18 df<-df[-5]
19 df$Promoffer<-as.factor(df$Promoffer)
20 df$Online<-as.factor(df$Online)
21 str(df)
22
23 # Partitioning: Tr:Te->60%:40%
24 partidx<-sample(1:nrow(df), 0.6*nrow(df), replace = F)
25 dfrtrain<-df[partidx,]
26 dftest<-df[-partidx,]
27
28 # Model with a single predictor: Income
29 mod1glm<-glm(Promoffer ~ Income, family = binomial(link = "logit"),
30                 data = dfrtrain)
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
```

Console Session 10/1:

```
> str(df)
'data.frame': 5000 obs. of 8 variables:
 $ Income : num 49 35 10 101 45 31 71 23 80 182 ...
 $ Spending : num 1.6 2.201 0.495 2.73 1 ...
 $ Promoffer : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ Age : num 25 45 39 35 35 37 53 50 35 34 ...
 $ Experience : num 1.19 15 9.8 13 27 24 10 9 ...
 $ Family.Size: num 4 3 1 1 4 4 2 1 3 1 ...
 $ Education : Factor w/ 3 levels "Grad","HSC","PostGrad": 2 2 2 1 1 1 1 3 1 3 ...
 $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...
```

R Documentation

Predict Method for GLM Fits

Description

Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object.

Usage

```
## S3 method for class 'glm'
predict.glm(..., newdata = NULL, ...)
```

So, 60 percent of the observation will go into the training partition as you can see in the environment section 3000 observations for df train 8 variables and for test partition the remaining observations that is 2000 observations on 8 variable. So, that is also there.

(Refer Slide Time: 03:43)

The screenshot shows the RStudio interface with the following details:

- File menu:** File, Edit, View, Plots, Session, Tools, Debug, Project, Help.
- Toolbar:** Source on Save, Run, Source.
- Text Editor:** The code for a logistic regression model is displayed. It includes:
 - Data loading: `dfonline<-as.factor(df\$online)`
 - Data summary: `str(df)`
 - Data partitioning: `partidx<-sample(1:nrow(df), 0.6*nrow(df), replace = F)`
 - Training set creation: `dfrtrain<-df[partidx,]`
 - Test set creation: `dftest<-df[-partidx,]`
 - Model fitting: `mod<-glm(promoffer ~ Income, family = binomial(link = "logit"), data = dfrtrain)`
 - Model summary: `summary(mod)`
 - Model coefficients: `b0<-unname(mod\$coefficients[1]); b0`
`b1<-unname(mod\$coefficients[2]); b1`
 - Fitted Model: `# Fitted Model`
- Environment Tab:** Shows variables `df` (5000 obs. of 8 variables), `dfb` (5000 obs. of 9 variables), `dftest` (2000 obs. of 8 variables), `dfrtrain` (3000 obs. of 8 variables), and `partidx` (int [1:3000 2591 4737 302]).
- Values Tab:** Shows the value of `partidx` as 2591.
- Files Tab:** Shows files like `R Scripts` and `HTML`.
- PLOTS Tab:** Shows a histogram of income.
- Packages Tab:** Shows packages like `gridExtra` and `knitr`.
- Help Tab:** Shows help for `predict.glm`.
- Viewer Tab:** Shows the output of the R code.
- Console Tab:** Shows the R session history.
- Output Tab:** Shows the results of the logistic regression fit.
- Project Tab:** Shows a project named "Project (None)".
- Help Tab:** Shows help for `predict.glm`.
- Documentation Tab:** Shows documentation for `predict.glm`.

Now, as we talked about in previous lecture the GLM is the function that can be used. So, this program order we have already build model with single beta we already discussed this one.

(Refer Slide Time: 03:53)

The screenshot shows an RStudio interface with the following details:

- Top Bar:** File, Edit, View, Tools, Window, Help.
- Left Panel:** A code editor window titled "10 logit.R" containing R code for a logistic regression model. The code includes plotting the relationship between income and promotional offers, fitting a model with all predictors, and summarizing the results.
- Right Panel:** An "Environment" tab showing the global environment with objects like df, dfb, dttest, dtrain, mod1, and values. It also includes tabs for "Data", "Values", "Files", "Plots", "Packages", "Help", and "Viewer".
- Bottom Left:** A "Console" window showing the output of the R code, including factor levels and summary statistics.
- Bottom Right:** A "Script" pane showing the predict.glm function and its documentation.

So, let us move to the model with all predictors. So, as you can see in this particular model, we are we have this formula promotional offer tilde dot so; that means, we are going to build model against all predictors using all predictors right other parameters remain same. So, let us run this run this model, let us look at the summary.

(Refer Slide Time: 04:23)

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Contains R code for logistic regression, including plotting odds ratios and fitting a model with all predictors.
- Console:** Displays the output of the logistic regression model, showing coefficients, standard errors, z-values, and p-values. The output includes:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.291594	2.179821	-4.263	2.02e-05 ***
Income	0.063759	0.003717	17.153	< 2e-16 ***
Spending	0.177935	0.052768	3.372	0.000740 ***
Age	-0.024237	0.081659	-0.297	0.766616
Experience	0.033651	0.081245	0.414	0.678736
Family.Size	0.644928	0.096514	6.682	2.35e-11 ***
EducationHSC	-0.087387	0.333602	-0.252	< 2e-16 ***
EducationPostGrad	0.043388	0.234768	0.185	0.853377
online1	-0.203790	0.195281	-1.044	0.296682

- Output:** Shows the results of the predict method for GLM fits, including a description of the method and usage information.

So, this particular model also the results of this model also we have discussed, however there is a one slight change in the results ah. So, in the previous run; that we had done in the last lecture, as we can see is spending this particular variable right. So, this was this particular variable this was significant at 90 percent confidence interval level in the p in the v in the run that we did in previous lecture; however, you can see that as the sample as changed ah.

Now, this is also significant at 99.9 percent significance level right. So, the results that we have today, in today's run; we can see that income is spending and the family size education HSC; these are the significant variables and 3 of them were significant at 99.9 percent level in previous run as well and in this particular lectures, run a spending also comes out to be significant. So, this is slight; this can happen when we run a particular model multiple times. So, a larger sample size can a guarantee us a more stable results more robust results right because it the model results also depends on the observations because training partition, we randomly draw observation from the full data set and then use them for training partition.

So, the observations the every time we run the observation a that are used for model a are going to change and therefore, a slight differ slight differences in terms of a model can be seen to repeat a to repeat the model a using the same observation as we have talked about in some of the initial lectures of this course set dot seed function can be used set

dot seed function will actually allow us to use the same partitioning same observations pertaining partition for the modeling as well. So, we have already discussed the results of this particular model. Now let us move forward. Now let us check the performance .

(Refer Slide Time: 06:39)

The screenshot shows the RStudio interface. On the left, the 'R Script' tab displays R code for logistic regression, including plotting odds and logit functions, fitting a model, and using the predict function. The 'Console' tab shows the execution of the code and the resulting output, including deviance statistics and iteration counts. On the right, the 'Help' window is open, specifically showing the documentation for the 'predict.glm' function under the 'glm' package.

```

1 # Odds
2 # P(odd/(1+odd))
3 curve(odd/(1+odd), from = 0, to=100, type = "l", xname = "odd",
4       xlab = "odds", ylab = "Probability of Success")
5 # P(exp(logit)/(1+exp(logit)))
6 curve(exp(logit)/(1+exp(logit)), from = -100, to=100, type = "l",
7       xname = "logit", las=1,
8       xlab = "logit", ylab = "Probability of Success")
9
10 # Score test partition for probability values
11 modtest<-predict(mod1, dftest[, -c(3)], type="response")
12 # Score test partition for logit values
13 modtest<-predict(mod1, dftest[, -c(3)], type="link")
14 # Classify observations using a cutoff value of 0.5
15 modtest<-ifelse(modtest>0.5, 1, 0)
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
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
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
856
857
858
858
859
859
860
861
862
863
864
865
866
866
867
868
868
869
869
870
871
872
873
874
875
876
876
877
878
878
879
879
880
881
882
883
884
885
886
886
887
888
888
889
889
890
891
892
893
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
904
905
905
906
907
907
908
908
909
909
910
911
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
```

(Refer Slide Time: 07:22)

The screenshot shows the RStudio interface. The left pane displays an R script named '10 logit.R' with the following code:

```
58 xlab = "logit", ylab = "Probability of success"
59
60 # score test partition for probability values
61 modtest<-predict(mod1, dftest[,-c(3)], type="response")
62 # score test partition for logit values
63 modtest<-predict(mod1, dftest[,-c(3)], type="link")
64 # Classify observations using a cutoff value of 0.5
65 modtestc<-ifelse(modtest<=0.5, 1, 0)
66
67 table("Actual Value"<dftest$Promoffer, "Predicted Value"<modtestc)
68 #classification accuracy
69 mean(modtestc==dftest$Promoffer)
70 #misclassification error
71 mean(modtestc!=dftest$Promoffer)
72
73 head(data.frame("Predicted Class"<modtestc,
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
```

The right pane shows the 'Environment' tab with objects: dfb (5000 obs. of 9 variables), dftest (2000 obs. of 8 variables), dftrain (3000 obs. of 8 variables), mod1 (Large glm (30 elements, 2 ..), modtest (Named num [1:2000] 0.00061..), modtestc (Named num [1:2000] 0 1 1 0 ..)). A help window for the 'predict.glm' function is open, showing its description and usage.

And then we can score in this fashion the observations. So, cutoff value is 0.5. So, we have just two classes. So, this is a two class case. So, 0.5 cutoff value of 0.5 will be equivalent to most probable class method and in this particular case. So, let us use it.

So, now let us look at our classification matrix. So, with this code we would be able to generate the same. So, you can see in the classification matrix out of the 2000 observations that we have in the trend test partition as you can see in the environment section as well. So, out of 2000 observations that we have 70, 175 observations have been correctly classified as class 0 members and hundred and twenty five observations have been correctly classified as class one members the of diagonal elements that is 65 and 35. So, these are the observations which have been incorrectly classified either into class 0 or class 1. So, we can go ahead and compute our classification accuracy. So, this comes out to be 95 percent in this particular run.

If you remember in the previous run that we did in a in the last lecture there also we got the similar number. So, that was also near about 95 point something in last lecture. So, you can see the model is a on in terms of performance numbers in terms of matrix the model is quite this stable and robust right in previous run we also got similar perform. So, the remaining is the error that is 5 percent.

Now we can compute the important variables for this particular modeling exercise where you have a predicted class actual class predicted class is stored in mod test c and then we

can create a data frame of all these important key variables here actual class is stored in promotional offer we can have probability value mod test. So, using this we can also have a look at the table this particular data frame and the table and have a look how our model has performed log Odds also we can have in this fashion mod test 1 that we have already computed.

And this is our then we can also have the test partitions those variables here in this particular data frame. So, let us look at first 6 observations of this particular data frame.

(Refer Slide Time: 10:00)

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows objects like `dfb`, `dftest`, `dfrtrain`, `mod1`, `modtest`, and `modtstc`.
- Plots:** A histogram titled "Predicted Class" is visible.
- Packages:** The `glm` package is selected.
- Help:** The help page for `predict.glm` is open.
- Console:** Displays the R code for logistic regression, the resulting data frame `dftest`, and a table of coefficients.
- Data View:** Shows the data frame `dftest` with columns: Experience, Family.Size, Education, and Online.

```
library(MoM)
# Fit a logistic regression model
dftest <- glm(Promotion ~ Experience + Family.Size + Education + Online, data = dfb, family = binomial)

# Check model summary
summary(dftest)

# Create a table of predicted values
table("Actual Value"=dftest$Promoffer, "Predicted value"=modtstc)

# Classification accuracy
mean(modtstc==dftest$Promoffer)

# #Model selection error
# mean(modtstc!=dftest$Promoffer)

# Head of data frame
head(data.frame("Predicted Class"=modtstc,
                 "Actual Class"=dftest$Promoffer,
                 "Prob for {Success}"=modtest,
                 "Log odds"=modtstc,
                 dftest[,-3], check.names = F))

# Cumulative Lift Curve
dflift<-data.frame("Probability of Class 1"=modtest,
                    "Actual Class"=as.numeric(as.character(dftest$Promoffer)),
                    modtstc)

# Predictions
predict.glm(dftest)
```

Console G:\Session 10/

	Experience	Family.Size	Education	Online	
11	1	0	0.5706654028	0.2845665	106 2.5716879 65
14	0	0	0.01363628148	-4.0962457	41 3.1384614 59
15	0	0	0.0045964706	-5.3778595	113 2.2231914 67
18	0	0	0.0032369970	-5.7298670	80 2.1406497 42

So, you can see predicted class and the actual class. So, because our accuracy is 95 percent for this particular model; however, in first 6 observational itself you would see that one error for this particular observation the actual class was 0, but it has been predicted as one, if we look at the a probability value for the same we can see that 0.57 is the probabilities value. So, is the probability value? So, therefore, it has been classified as class 1 ah; however, actual class is 0.

If we look at the other numbers for example, the first row here you can see the probabilities value is quite low. So, therefore, it has been correctly classified as class 0 we look at the row number 2 the probability value is .98 quite close to 1. So, it has been correctly classified as class 1 and this one is the error right probability value is a more than 0.5 therefore, it has been classified as one even though the actual class was 0; if we look at 3 remaining a 3 remaining rows also, we can see that the all for all these 3 rows

the probability value is much less than 0.5, it is quite close to 0. So, therefore, all the all these 3 rows have been correctly classified as class 0.

So, a log Odds value a you can also see. So, you can see the values which are close to 0. So, as we had seen the plot of you know probability a probability versus log Odds logic values. So, from there, we also I can understand that the log Odds value is Logit values on the negative side so; that means, it will have the corresponding probabilities value quite close to 0. So, the same thing you can we can see in all the rows where the Logit values are negative similarly a positive Logit a values as we saw in previous lecture, in the plot that positive logic values.

They will typically mean a higher probability value a probability value close to one the same thing is reflected in row number 2 positive Logit value and higher probability corresponding value if we look at this particular value. So, we saw that that around the you know when the Logit value is around 0 mark, then we see sudden you know change in a probability value. So, all the variation in the probabilities values come around the when the Logit value is near about 0 mark.

So, you can see 0.28 when the Logit value is near about 0 mark 0.28; you can see that the probability value is also near about 0.5, right, you can see 0.57 in this particular case and these are the cases and these are the cases the cases where Logit value is close to 0; that means, the probability value will be close to a 0.5 mark you know on either direction. So, those are the cases which will which will be difficult to classify for a model in this case, as we can see also row number 3 the model was not able to classify correctly the observations.

Then a the predictors variables have also been added to this particular table. So, that can also be analyzed accordingly income spending aged experienced family size education. So, that can also be analyzed. So, if we look at the most interesting row that is the row number 3 here you can see the income levels the spending and the age. So, on the higher side our experience and family size and education. So, we can look at different a values specific values for a particular observation and we can understand the results further another thing that is possible here is that a we can have a look at the we can have a look at the values which were which are you know which have been incorrectly you know which have been incorrectly classified by the model.

So, if you are interested in those value. So, we can previous command was this one.

(Refer Slide Time: 14:23)

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Audit, Catalog, Photo, Help, and Help. A toolbar with icons for Source On Save, Run, Source, and Addins is visible. The left pane contains a script editor with R code for logistic regression, including dftest, dftrain, and predict.glm. The right pane shows the Environment and History panes, and a search bar. The bottom pane is the Console, displaying the results of the R code execution.

```
67 table("Actual value"=dftest$Promoffer, "Predicted Value"=modtest)
68 #classification accuracy
69 mean(modtest==dftest$Promoffer)
70 #misclassification error
71 mean(modtest!=dftest$Promoffer)
72
73 head(data.frame("Predicted Class"=modtest,
74                 "Actual Class"=dftest$Promoffer,
75                 "Prob for l(success)"=modtest,
76                 "Log odds"=modtest,
77                 dftest[,3], check.names = F))
78
79 # cumulative Lift Curve
80 dflift=data.frame("Probability of Class 1"=modtest,
81                    "Actual Class"=
82                     as.numeric(as.character(dftest$Promoffer)),
83                    .by="Actual Class")
83
781 (Top Level) R Script
```

Console G:/Session 10/ ↴

```
8      24      1 PostGrad    0
10     9       1 PostGrad    0
11    39      4 PostGrad    0
14    32      4   Grad     1
15    41      1    HSC     0
18    18      4    HSC     0
> DF=data.frame("Predicted Class"=modtest,
+                 "Actual Class"=dftest$Promoffer,
+                 "Prob for l(success)"=modtest,
+                 "Log odds"=modtest,
+                 dftest[,3], check.names = F)
> DF[which(DF[,3]<0.6&DF[,3]>0.4),]
```

So, a in this particular data frame itself we can look for the values which have been incorrectly classified right. So, first we will have to store this particular variable in a data frame. So, let us say df. So, we; so, we store this particular variable in this a data frame df and now within this df, if we are interested in finding the rows where the predicted class was not equal to predicted class was not equal to the actual class, right or rather more interesting rows would be where the probability value a the probability value that is a the third third that is the third column right probability value is close to 0.5 right.

So, that would be more interesting the, those would be more interesting observation. So, let us compute the same. So, third row and we would like it to be let us say less than 0.6 and the same observations we would like it to be greater than let us say 0.4; so, all the observations which all the rows which follow this.

(Refer Slide Time: 16:00)

The screenshot shows the RStudio interface. The left pane displays an R script named '10 logit.R' with the following code:

```

67 table("Actual value"=df$Promoffer, "Predicted Value"=modtestc)
68 #classification accuracy
69 mean(modtestc==df$Promoffer)
70 #misclassification error
71 mean(modtestc!=df$Promoffer)
72
73 head(data.frame("Predicted Class"=modtestc,
74                 "Actual Class"=df$Promoffer,
75                 "Prob for 1(success)"=modtestc,
76                 "Log odds"=modtestc,
77                 df$test[,3], check.names = F))
78
79 # cumulative Lift curve
80 dflift=data.frame("Probability of class 1"=modtestc,
81                   "Actual Class"=
82                   as.numeric(as.character(df$Promoffer)),
83                   )
84
85 > df[which(df[,3]<0.6&df[,3]>0.4),]

```

The right pane shows the 'predict.glm()' function documentation under 'R Documentation'. It includes sections for 'Description', 'Usage', and 'Details'.

So, you we can see here and the results. So, that there could be too many observations in this case. So, there can be too many observations. So, let us take a first few observations let us take a let us twenty observations here again. So, in this fashion we can do it. So, let us scroll.

(Refer Slide Time: 16:30)

The screenshot shows the RStudio interface. The left pane displays the same R script '10 logit.R' as before, but the command at the bottom is now:

```

4985 > df[which(df[,3]<0.6&df[,3]>0.4),][1:20,]

```

The right pane shows the 'predict.glm()' function documentation. The 'Usage' section includes the command:

```

## S3 method for class 'glm'
predict(fit, newdata = NULL, ...

```

So, now, we can see so, these are the observations for which we probability values range from 0.4 to 0.5 as you can see from our criteria as well, right. So, probability is value range from 0.4 to 0.6. So, that was the range where Logit values a you know are close to

0 and we see a change in you know sudden change, you know a in a probabilities values near about this range.

So, now let us look at the some of these observations we can see the probabilities values are co close to 0.5 and Logit values are close to 0, right and all if we look at the weather these observations have been correctly classified you can see first one first row incorrectly classified second third fourth incorrectly classified with the fourth row where we see the correct classification and if we look at a look further then this one is incorrectly classified.

(Refer Slide Time: 17:30)

The screenshot shows the RStudio interface. On the left, the 'R Script' pane displays R code for a logistic regression model. The code includes loading data, fitting a model, and calculating various metrics like classification accuracy and a cumulative lift curve. On the right, the 'Environment' pane shows data frames like 'df', 'dfb', 'dftest', and 'dfrain'. A 'Help' window is open for the 'predict.glm' function, showing its description and usage. The 'Description' section states: 'Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object.' The 'Usage' section shows the command: '## S3 method for class 'glm''.

```

67 table("Actual value"=dftest$Promoffer, "Predicted Value"=modtestc)
68 #classification accuracy
69 mean(modtest==dftest$Promoffer)
70 #misclassification error
71 mean(modtest!=dftest$Promoffer)
72
73 head(data.frame("Predicted Class"=modtestc,
74                 "Actual Class"=dftest$Promoffer,
75                 "Prob for l1(success)"=modtest,
76                 "Log odds"=modtest$,
77                 dftest[,3], check.names = F))
78
79 # Cumulative Lift Curve
80 dflift<-data.frame("Probability of class 1"=modtest,
81                      "Actual Class"=
82                      as.numeric(as.character(dftest$Promoffer)),
83
84
85
86      1   0    0.5685043  0.27575153  111 2.5518666
87      1   0    0.5380182  0.15236709  112 1.7950494
88      1   0    0.5947861  0.38378692  114 2.2731626
89      1   1    0.5234209  0.09375212  170 2.7904483
90      1   0    0.5271490  0.10870272  112 2.4000250
91      1   0    0.5132092  0.05284899  156 5.9669782
92      0   1    0.4653474  -0.13883295  116 5.9669782
93      1   0    0.5132092  0.05284899  192 7.2664183
94      0   1    0.4617525  -0.15328934  123 5.6560853
95      0   1    0.4575891  -0.17005230  116 4.7952220
96      0   0    0.4490289  -0.20459519  110 1.524640
97      0   1    0.4181346  -0.33043557  180 0.7838342
98      0   0    0.4876569  -0.04938226  194 6.0810362
99      n   n    n.4176004  .n.3211866n  101 1.047600
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
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
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1408
1409
1410
1411
1412
1413
1414
1415
1416
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1508
1509
1510
1511
1512
1513
1514
1515
1516
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1608
1609
1610
1611
1612
1613
1614
1615
1616
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1708
1709
1710
1711
1712
1713
1714
1715
1716
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1808
1809
1810
1811
1812
1813
1814
1815
1816
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1908
1909
1910
1911
1912
1913
1914
1915
1916
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
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
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096

```

(Refer Slide Time: 17:46)

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Displays R code for a logit model. The code includes:
 - Table creation with columns "Actual value", "Predicted Value", and "modtestc".
 - Classification accuracy calculation.
 - Mean of modtestc.
 - Mean of modtest.
 - Head of data frame with columns "Predicted Class", "Actual Class", "Prob for 1(success)", "Log odds", and "dftest[,3]".
 - A comment about a cumulative lift curve.
 - Data frame creation with "Probability of class 1" from modtest, "Actual Class" as numeric, and "Log odds" from dftest.
 - Final line: 781 (Top Level) 2
- Console:** Shows the output of the R code, including a table of 20 observations with columns: Age, Experience, Family Size, Education, Online, Predicted Class, Prob for 1(success), Log odds, and modtestc.
- Environment:** Shows objects like DF, df, dfb, dftest, and dftrain.
- Help:** A tooltip for the "predict" method of a GLM fit is shown, describing it as a function to obtain predictions and standard errors for a fitted GLM object.

So, very few observations seem to be out of the twenty observation within this range 0.4 to 0.6 that we have seen.

So, in a sense from this kind of analysis we can see that our model is you know our model is able to correctly classify the clear case records and when the situation comes a bit close where the probabilities values are quite close to 0.5 our logit values are close to 0 in those situations the performance of the model performance of the model goes down most of the values are being incorrectly classified; however, if we look at the overall picture the model is giving us 95 percent accuracy. So, that is mainly because of some of the easy some of the direct maybe more observation which are easier to predict.

So, in some situations in this kind of situation we would require expert knowledge. So, the observations which have probability value close to 0.5. So, a in these cases can be identified and you know closure is scrutiny with the help of experts can be done to classify these observations.

(Refer Slide Time: 19:05)

The screenshot shows the RStudio interface with the following details:

- Code Editor:** The script pane contains R code for a logistic regression model. The code includes:
 - Importing datasets (df, dfb, dftest, dftrain).
 - Performing a Chi-squared test (modtest) on the dftest dataset.
 - Creating a data frame (dflift) with columns for Predicted Class, Actual Class, Probability of class 1, Log odds, and dftest.
 - Computing a cumulative lift curve.
 - Displaying the head of the dflift data frame.
 - Computing the cumulative actual class.
 - Displaying the head of the hflift data frame.
- Console:** The console pane shows the output of the R code, including the data frames dftest, df, dfb, dftest, and dftrain, and the head of the dflift and hflift data frames.
- Environment:** The environment pane shows the global environment with objects like df, dfb, dftest, and dftrain.
- Help:** A help window for the "predict.glm" function is open, showing its description and usage.

Now let us look at the cumulative lift curve for this particular exercise for this particular model. So, for this as we have done in some of the techniques in previous lectures as well. So, I will have will clear this particular data frame first column would be the probability of class one in this case, mod test is storing that information a actual class in this fashion because you can see the code is just slightly you know adjusted. So, that we get the values in numeric form because later on we would be computing the cumulative actual class number. So, you can see promotional offer it was converted into a factor variable; however, the labels where is to 0 and 1.

So, we would we would first required to change it to a character variable now the levels would be now the ones. So, labels would be gone and the values would be in correct format 0 and one and then from that we can convert into a numeric format 0 and one right. So, directly the direct conversion factor to numeric might lead to some errors and the values might not be in the desired format. So, if we directly convert from factor variable to numeric variable.

So, the classes would be classes a number of the numeric code for class 0 could can become one and numeric code for class one can become two; however, you would like to have numeric code for class 0 at 0 and numeric code for class one as one because we require certain computation based on that those values. So, this code will give us the

desired value. So, factor labels for 0 and one and when we converted it into a numeric vector then the values will also be 0 and one using this particular code.

So, let us create this data frame, let us look at the first 6 observations.

(Refer Slide Time: 21:05)

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Displays R code for generating a cumulative lift curve. The code includes `head(data.frame` for 'Predicted Class' and 'Actual Class', `dflift` for 'Probability of Class 1', and `cumactual` for cumulative actual class.
- Console:** Shows the output of the R code, including the generated data frames and their head contents.
- Environment:** Shows the global environment with objects like `df`, `dfb`, `dflift`, `dftest`, and `dfrtrain`.
- Help:** A help page for the `predict.glm` function is open, detailing its purpose as "Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object".

So, we can see in the first column we have the probabilities values and we also have the corresponding actual class. So, please note this that these are the estimated probabilities and the actual class. So, this particular cumulated class in this exercise we have gone through before as well. So, now, what the next thing that we would do we would sort this particular data frame in the decreasing order of probabilities values, right. So, order is order function can be used and the decreasing argument has to be set as true. So, that we get the values in the decreasing order ah. So, let us run this code let us look at the observations.

(Refer Slide Time: 21:44)

The screenshot shows the RStudio interface. In the top-left pane, there is an R script titled '10 logit.R' with the following code:

```
77 dftest[,3], check.names = F))
78 # Cumulative Lift Curve
79 dflift<-data.frame("Probability of class 1":modtest,
80 "Actual class": as.numeric(as.character(dftest$Promoffer)),
81 check.names = F)
82 head(dflift)
83 dflift=dflift[order(dflift[,1], decreasing = T),]
84 head(dflift)
85 cumActualClass=cumsum(dflift[,2])
86 dflift=cbind(dflift, cumActualClass)
87 head(dflift)
88 dflift
89 range(1:nrow(dflift))
90 range(dflift$cumActualClass)
91
92
93
94
```

In the bottom-left pane, the console shows the execution of the code:

```
> dftest[,3], check.names = F))
> # Cumulative Lift Curve
> dflift<-data.frame("Probability of class 1":modtest,
> "Actual class": as.numeric(as.character(dftest$Promoffer)),
> check.names = F)
> head(dflift)
> dflift=dflift[order(dflift[,1], decreasing = T),]
> head(dflift)
> Probability of Class 1 Actual Class
2957 0.9982728 1
1085 0.9978341 1
2383 0.9971026 1
783 0.9969172 1
4283 0.9964823 1
1653 0.9956191 1
> cumActualClass=cumsum(dflift[,2])
> dflift=cbind(dflift, cumActualClass)
> |
```

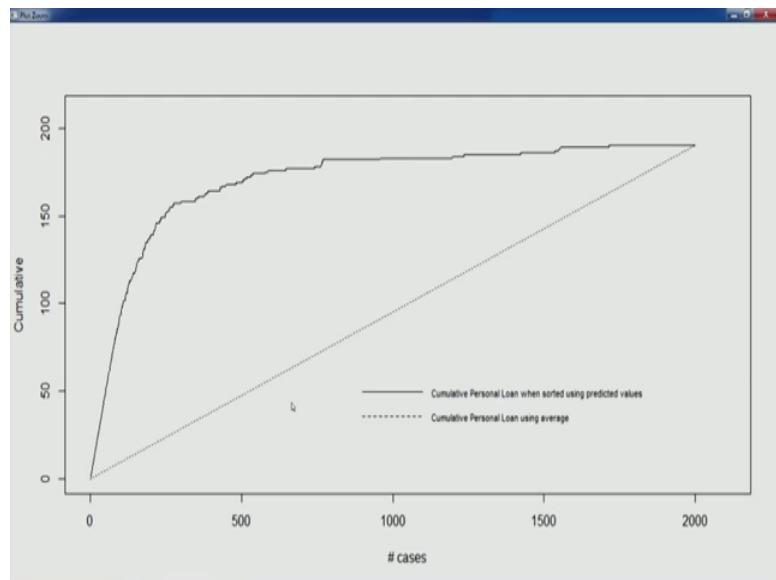
In the top-right pane, the 'Environment' tab is selected, showing objects like 'df', 'dfb', 'dflift', 'dftest', and 'dfrtrain'. In the bottom-right pane, a help window for the 'predict.glm()' function is open, showing its description and usage.

Now if you see that very first row is having the highest probability value followed by the observation with second highest probability value and you would see that first if your observation are all are close to 1.99 something numbers and the actual class is also 1. Now with this with this transformation of this data frame we can go ahead and compute the cumulative actual class ah. So, a this come sum is the function that can be used to ah perform, this computation in our environment ah. So, you can see that in the second column we are applying function. So, we will get the cumulative number in this and stored in this worker variable come cumulative come actual class. So, let us compute this let us add this particular variable in to data frame; let us look at the first 6 observation.

So, now you can see probability and the actual class and the cumulative actual class you can see the numbers also one two 3 four five six. So, now, let us plot our cumulative lift curve. So, first let us look at the range for x axis. So, one to 2000 that is the number of observation in test partition and let us look at the range for a y axis that is range for cumulative actual class. So, 1 to 19 so, that is the range. So, in that sense we can also understand that we have hundred and nineteen in our data set of 2000, we have 190 observations belonging to class 1. So, that is also clear from that.

So, now let us plot you can see that limits x limit y limits are appropriately specified. So, that we focus mainly on the data points the plot region let us generate this plot. So, this is the plot; let us also create the reference line and a legend for the same.

(Refer Slide Time: 23:47)



Let us look at the plot. So, this is our cumulative lift curve. So, as you can see that as we have talked about in when we generated cumulative lift curve for some other techniques, alright so, when we look to identify a first few observations most probable ones right. So, from this from this particular plot we can understand the ability of the model in identifying the most probable ones. So, this lift from the reference line this indicates. So, this particular line solid line is representing the model and the dotted line is representing the reference case different scenario baseline scenario.

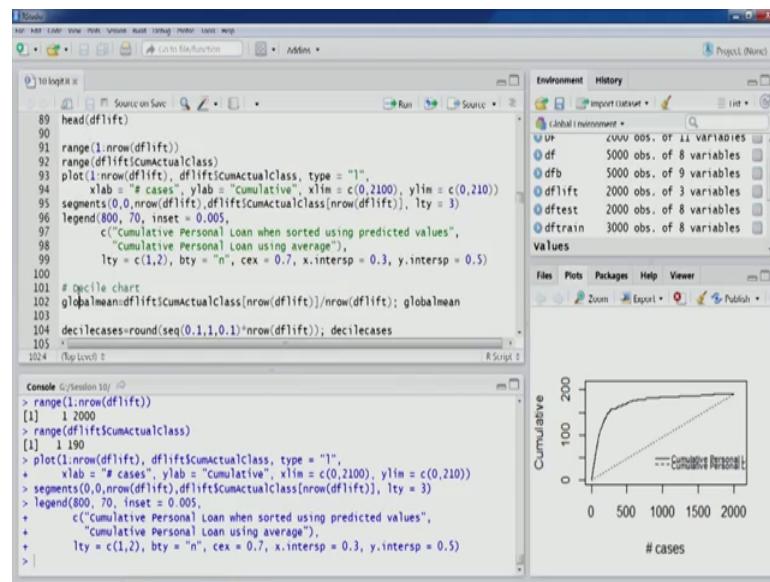
New model new rule and from this we can say our in terms of identifying the most probable ones in terms of identifying the customers who are more likely to accept the promotional offer this model does a good job and provides a good very good lift in comparison to reference in comparison to the benchmark case. So, we can see that lift is quite high in the initial part of the curve and as we look to identify more such cases and the lift keeps you know a lift starts decreasing, right that is because there are just 190 total observations which fall which total observation which actually fall in that category the individuals who have a customer who have accepted the offer.

So, as we go about reaching that number you can see here it is 2000. So, this particular mark is one ninety. So, as we go about a reaching this number the performance of the model start a merging with the performance of new rule; however, in terms of identifying the most probable ones, right ah. So, what we are looking here is the top left corner; so,

this particular corner. So, if we are looking to if we are you know if we are with identifying these many observations. So, the model gives us a quite good performance and comparison to the new rule you can see even at this point we will be able to identify about 150; 100 you know 55 more of more than 150 a individuals you know who who are more likely to accept the offer which is quite close to 190.

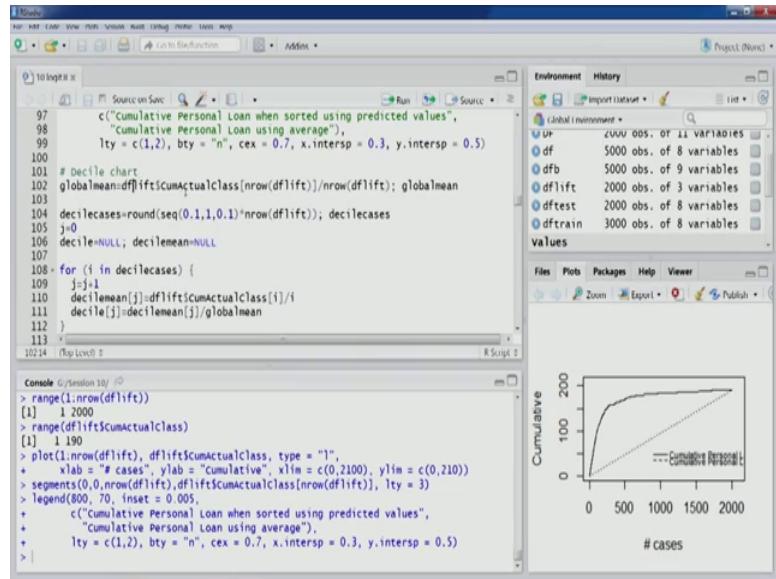
So, in terms of that in terms of identifying the most probable once the model does quite a good job. .

(Refer Slide Time: 26:30)



Now the same information can be further understood using the Decile chart. So, as we have done in previous techniques also.

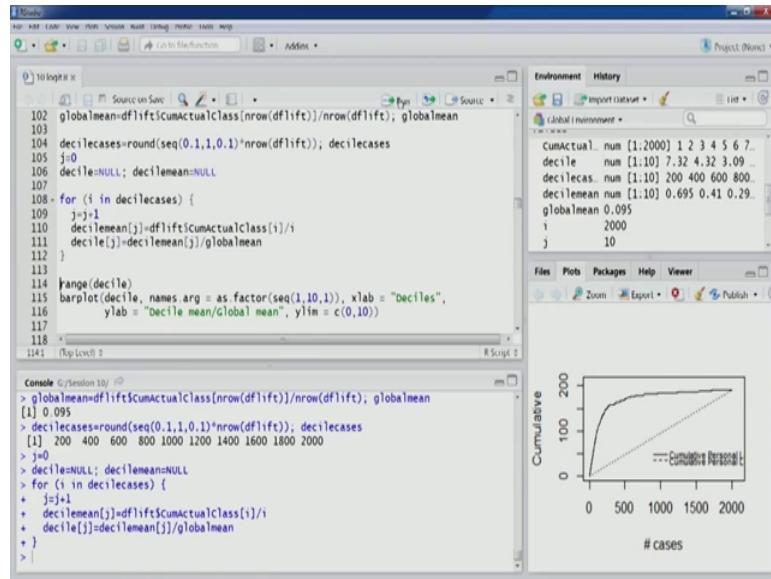
(Refer Slide Time: 26:35)



So, in Decile chart, we will have to compute this a global mean. So, you can see that cumulative actual class variable and we are trying to compute the global mean the corresponding value for the last observation and then total number of observation. So, that will give us the, a global mean. So, this is the number point 0 nine five then a Decile cases we would like to have 10 Decile. So, each Decile which represent traditional 10 percent of cases; so, first Decile would represent 10 percent second Decile 20 percent cases third Decile 20 percent cases.

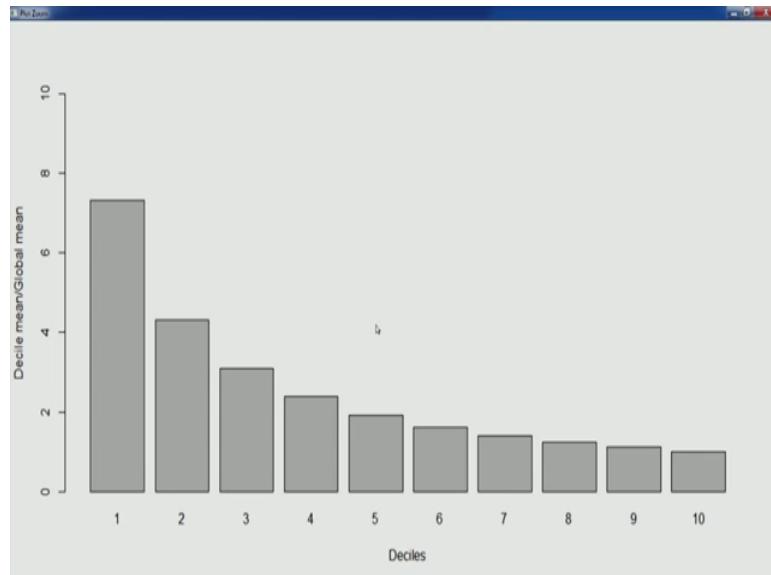
So, in this fashion, we can compute you can see this particular sequence is multiplied with number of observations. So, this will give us the appropriate number of observations for each Decile. So, once this is done we need a counter this counter is basically for the Decile. So, this is actual Decile counter for Decile ah. So, let us initialize this then we have Decile you know a this variable to store the ratio of Decile mean to global mean and the Decile meaning actually mean for each of the Decile. So, let us initialize these variables and in the for loop as you can see this is running from all the values that are in Decile case cases right. So, 10 Deciles and the number of cases and those respective Deciles and once we run this, we will have the numbers let us look at the range of Decile.

(Refer Slide Time: 28:13)



So, this is one to 7.3 something and so, the you can see the limits on y axis have been appropriately specified add 0 to 10, you can also see that other arguments are also for example, on x axis labels that is 1 to 10 Decile; 1 to 10 and other things are appropriately specified. So, let us create the Decile chart.

(Refer Slide Time: 28:42)



So, this is the Decile chart which can be created using the function like we did using a function bar plot. So, we can see. So, a then formation that we saw in the cumulative lift curve the same information is being defective a being depicted in a different format the

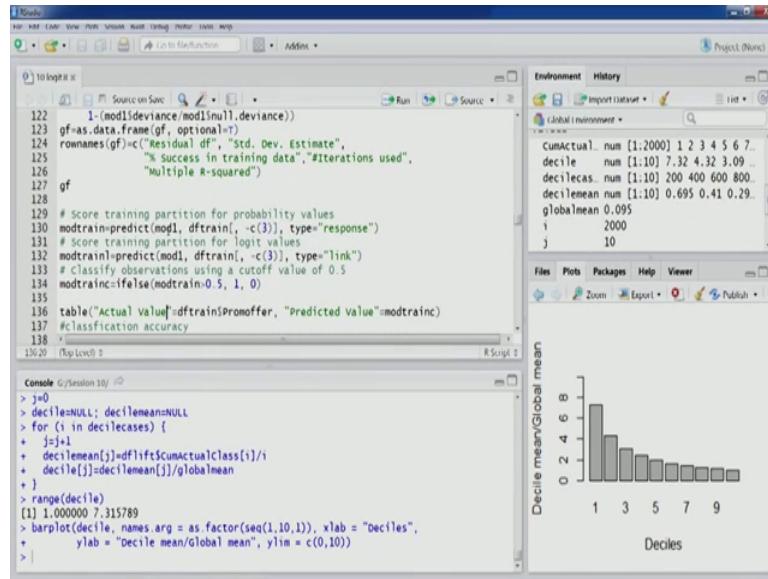
bar chart format in the design chart. So, you can see Deciles. So, each Decile as I talked about first Decile is representing the first 10 percent values.

So, second and the twenty percent thirty percent fashion. So, in a way first Decile is giving us telling us in terms of Decile means y axis a Decile mean divided by global mean. So, this positive cell is giving us the idea about; how well the model will perform in comparison to average case in identifying the most probable ones most likely customers the customer which are most likely to respond most likely to accept the promotional offer. So, you can see that for first 10 percentage a 10 percent of cases the lift is quite high its more than 7, if we look for 20 percent first twenty percent cases and the model still gives us good lift more than a 4 and if we look at the first 30 percent cases the model still gives us good lift more than 2 near about 3 and in this fashion as we can see just like the cumulative lift curve.

As we look to identify a more number of customer which are likely to accept the offer our lift value goes down the same is reflected in Decile chart if we look for this Decile 4 5 6 7; that means, we are looking to identify most probable 40 percent, 50 percent, 60 percent you know cases. So, our lift will go down. So, typically the, you know; we can we can go for the up to or Decile where the lift value is still greater than one. So, we look at near about you know a eighth Decile; that means, a 80 percent of the cases, this is about near about seems to be near about one. So, from this also we can understand that out of 190 observe of 190 observation which have you know 190 customers which have accepted the promotional offer about 80 percent of them can be easily identified by the model.

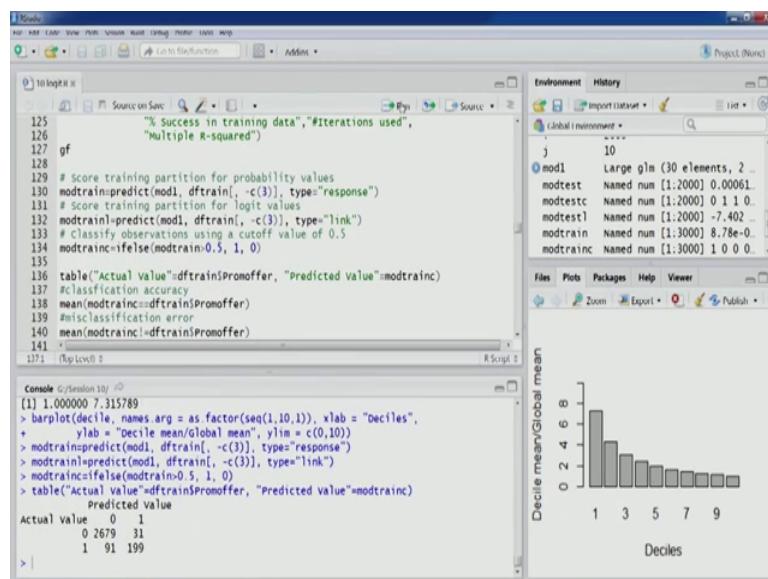
We can further look at some of the measures of goodness of it. So, these are some of the values.

(Refer Slide Time: 31:20)



So, we will discuss some of these values and later lecture now, right. Now what we will do will look for our performance in the training partition. So, the performance that we have seen till now is watch for the test partition now do the let us do the same exercise on training partition itself. So, let us have a look on the same. So, let us compute the probabilities values followed by Logit values and followed by classification just like we did for test partition. So, let us look at the classification matrix.

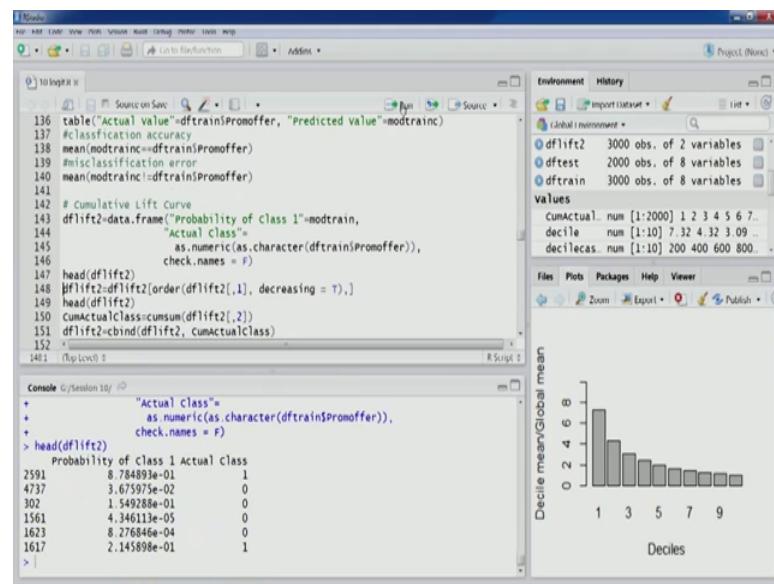
(Refer Slide Time: 31:51)



Here we can see out of 3000 observation good number of observation majority of the observation have been correctly classified as we can see in the diagonal elements, right.

So, now let us look at the classification accuracy you can see 0.959. So, this is more than the performance on tests partition which is expected because these are the observation on which the model has been built. So, the error is this much about 4, we can also create cumulative lift curve and Decile chart for this particular partition as well.

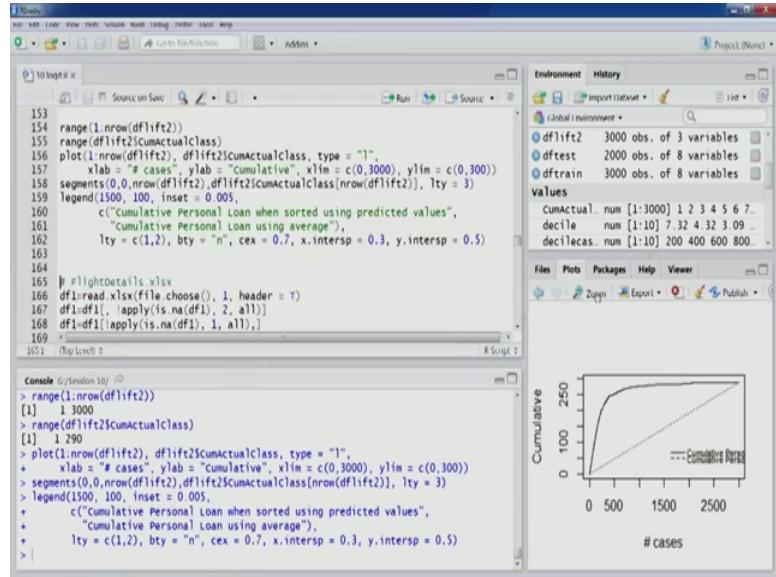
(Refer Slide Time: 32:34)



So, this data frame is created and let us look at these first 6 observation. Now, we let us sort it out let us order it in the decreasing value this is.

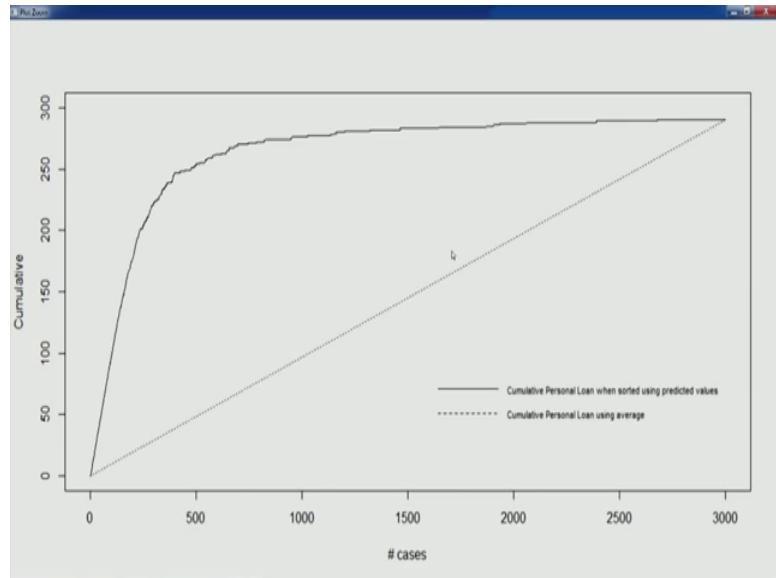
So, most of the values we can see now first 6 observation close to one let us compute the cumulative values let us look at this. So, once this is done we can go ahead and create our lift curve.

(Refer Slide Time: 33:04)



So, we can see here. So, this is the curve for the training partition.

(Refer Slide Time: 33:09)



So, we can see because the model is doing good on test partition also. So, both training and test lift curve look a quite similar. So, with this will a stop here and we will do another exercise to understand further non logistic model in next lecture.

Thank you.