Cory Suzuki

STAT 574

Dr. Olga

7 February 2025

STAT 574 HW1

Problem 1.

SAS Code

```
proc import out=hospital
datafile="C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/hospital_data.csv"
dbms=csv replace;
run;

/* (a) Splitting the data into 80% training and 20% testing sets*/

proc surveyselect data=hospital rate=0.8 seed=479576
out=hospital outall method=srs;
run;

/*RSS Splitting Criterion-Full Tree*/

proc hpsplit data=hospital seed=113123;
    class ASA gender;
    model surgery_cost = gender age BMI ASA surgery_duration_min;
    grow RSS;
    partition rolevar=selected(train="1");
run;

/*RSS Splitting Criterion and Cost-Complexity Pruning*/

proc hpsplit data=hospital seed=113123;
    class ASA gender;
    model surgery_cost = gender age BMI ASA surgery_duration_min;
    grow RSS;
    prune costcomplexity(leaves=12);
    partition rolevar=selected(train="1");
    output out=predicted;
    ID selected;
run;

/*(b) Computing prediction accuracy for testing data*/
```

```
data test;
    set predicted;
    if (selected="0");
    keep _leaf_ surgery_cost P_surgery_cost;
run;

data accuracy;
    set test;
    if(abs(surgery_cost-P_surgery_cost)<0.10*surgery_cost)
    then ind10=1; else ind10=0;
    if(abs(surgery_cost-P_surgery_cost)<0.15*surgery_cost)
    then ind15=1; else ind15=0;
    if(abs(surgery_cost-P_surgery_cost)<0.20*surgery_cost)
    then ind20=1; else ind20=0;
run;

proc sql;
    select mean(ind10) as accuracy10, mean(ind15) as accuracy15, mean(ind20) as
accuracy20
    from accuracy;
quit;

/* (c) CHAID Splitting Criterion - Full Tree*/

proc hpsplit data=hospital seed=108698;
    class ASA gender;
    model surgery_cost = gender age BMI ASA surgery_duration_min;
    grow CHAID;
    partition rolevar = selected(train="1");
run;

/*CHAID Splitting Criterion -Cost Complexity Pruning*/

proc hpsplit data=hospital seed=108698;
    class ASA gender;
    model surgery_cost = gender age BMI ASA surgery_duration_min;
    grow CHAID;
    prune costcomplexity(leaves=33);
    partition rolevar=selected(train="1");
    output out=predicted;
    ID selected;
run;

/*(d) Computing prediction accuracy on testing set for CHAID Tree*/
```

```
data test;
    set predicted;
    if (selected="0");
    keep _leaf_ surgery_cost P_surgery_cost;
run;

data accuracy;
    set test;
    if(abs(surgery_cost-P_surgery_cost)<0.10*surgery_cost)
    then ind10=1; else ind10=0;
    if(abs(surgery_cost-P_surgery_cost)<0.15*surgery_cost)
    then ind15=1; else ind15=0;
    if(abs(surgery_cost-P_surgery_cost)<0.20*surgery_cost)
    then ind20=1; else ind20=0;
run;

proc sql;
    select mean(ind10) as accuracy10, mean(ind15) as accuracy15, mean(ind20) as
accuracy20
    from accuracy;
quit;
```

The SAS System

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	HOSPITAL
Random Number Seed	479576
Sampling Rate	0.8
Sample Size	3047
Selection Probability	0.800158
Sampling Weight	0
Output Data Set	HOSPITAL

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

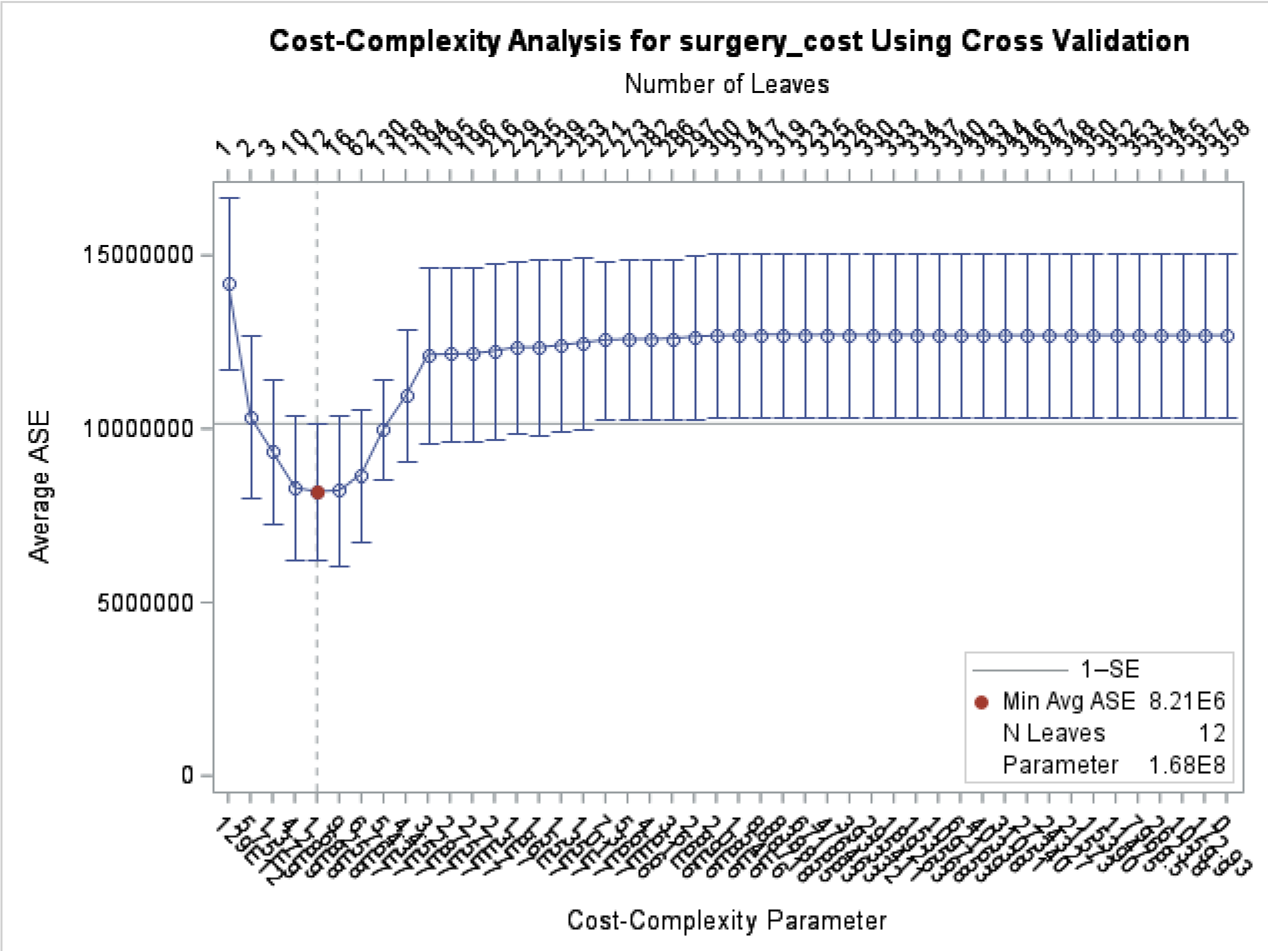
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

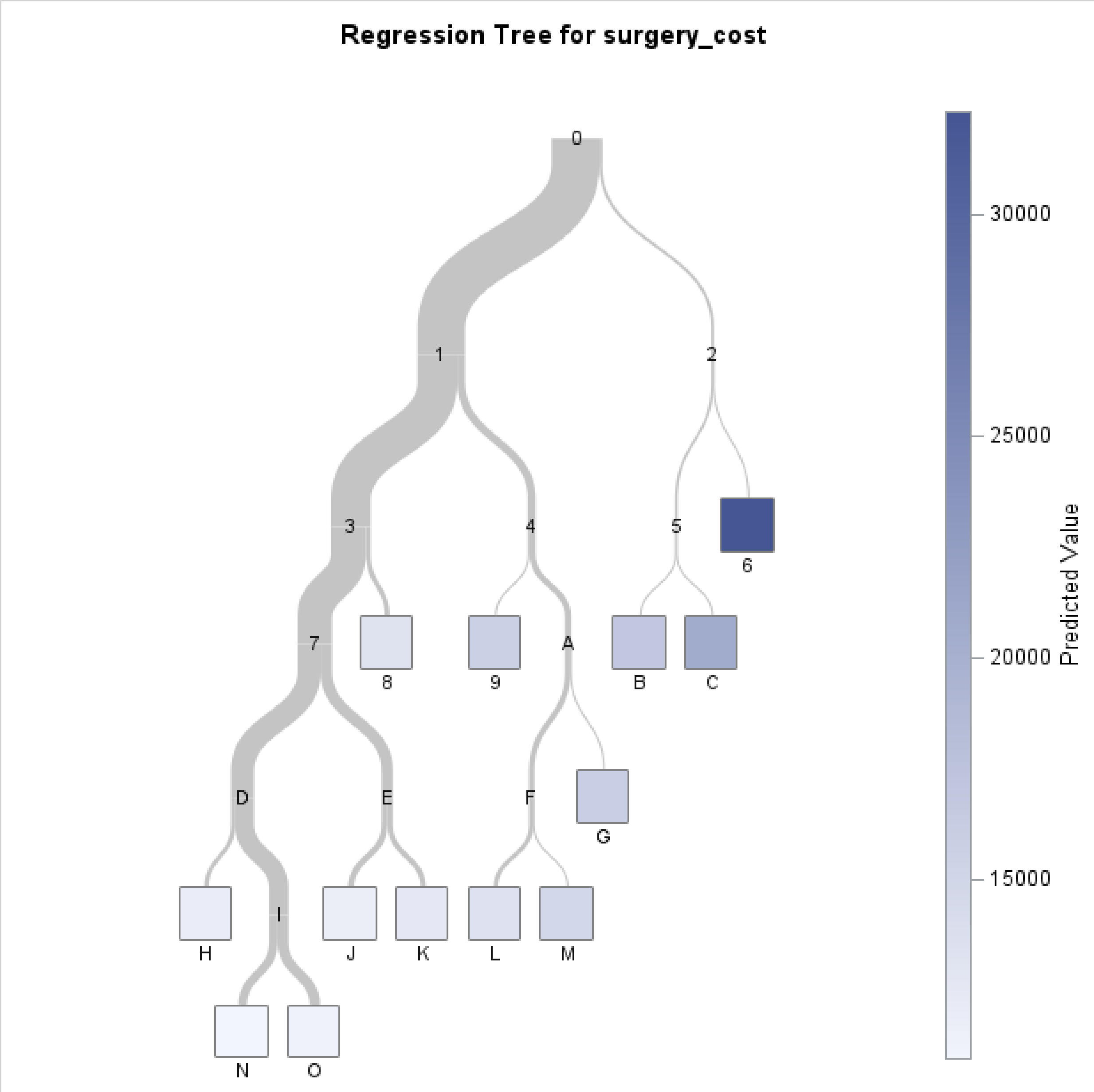
Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	13

Number of Observations Read	3047
Number of Observations Used	3047

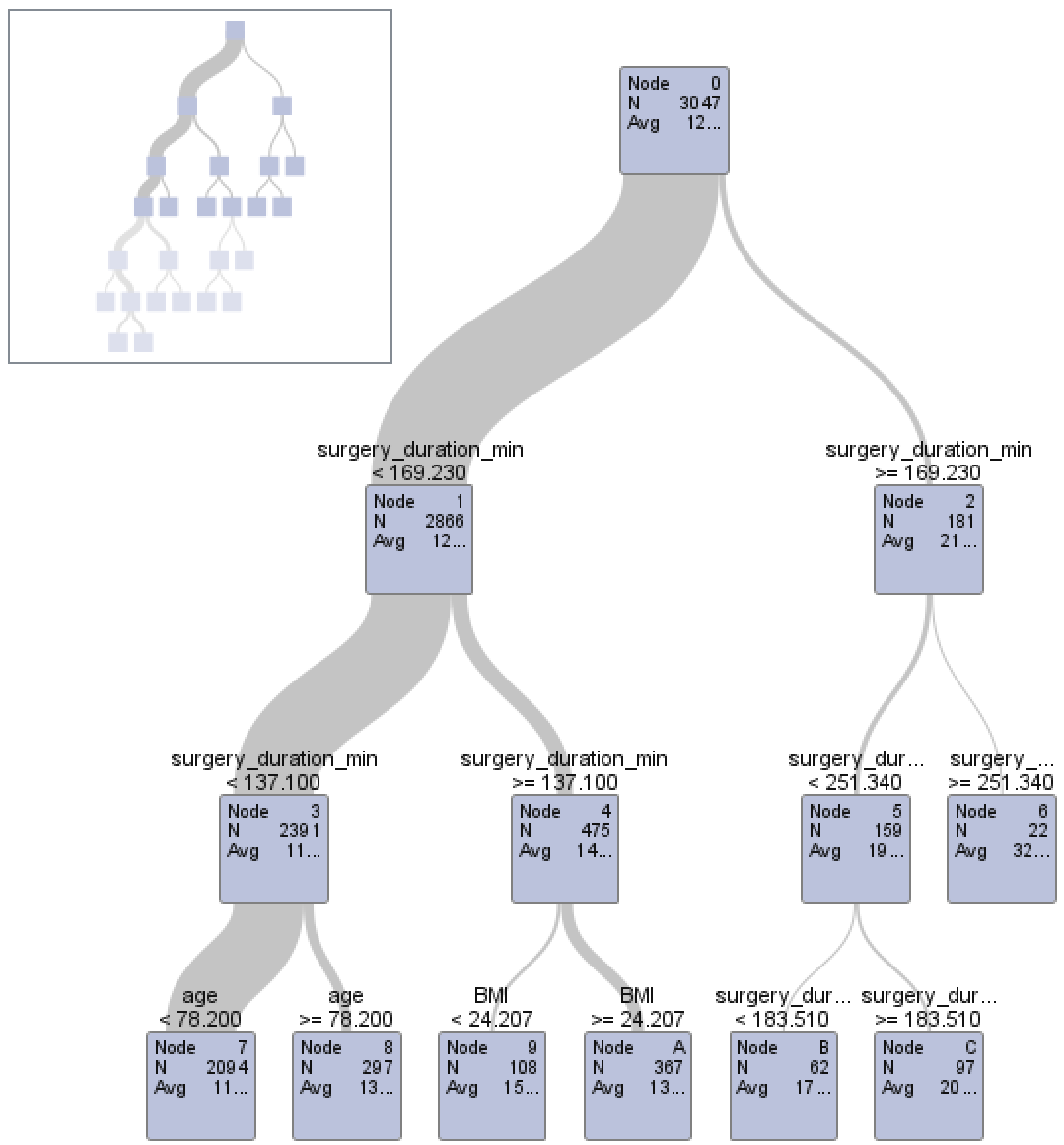
The HPSPLIT Procedure



The HPSPLIT Procedure



Subtree Starting at Node=0



The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
13	7242180	2.207E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	140554	6
age	0.2287	32141.5	4
BMI	0.1187	16676.8	1
ASA	0.0862	12112.9	1

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

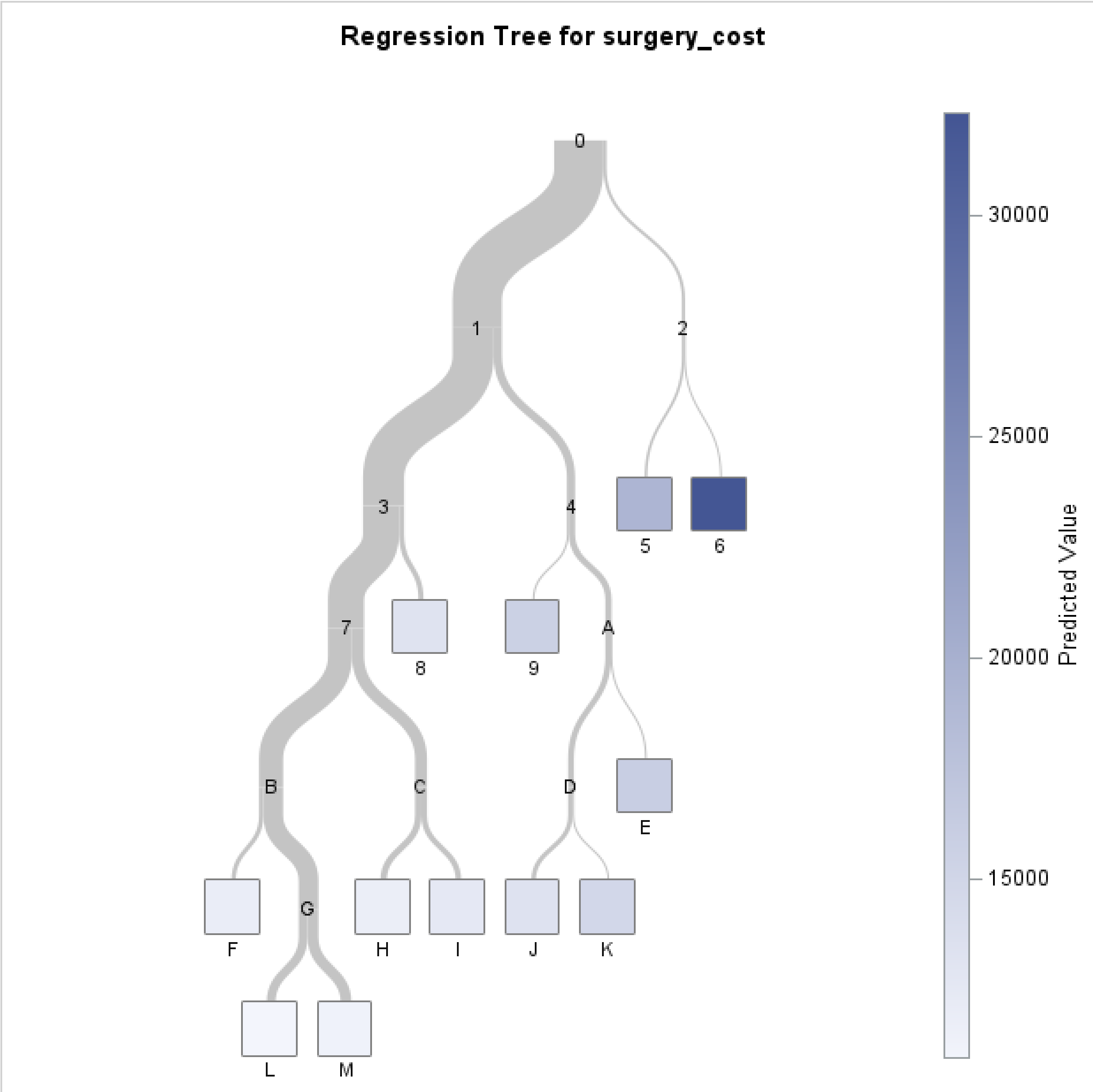
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

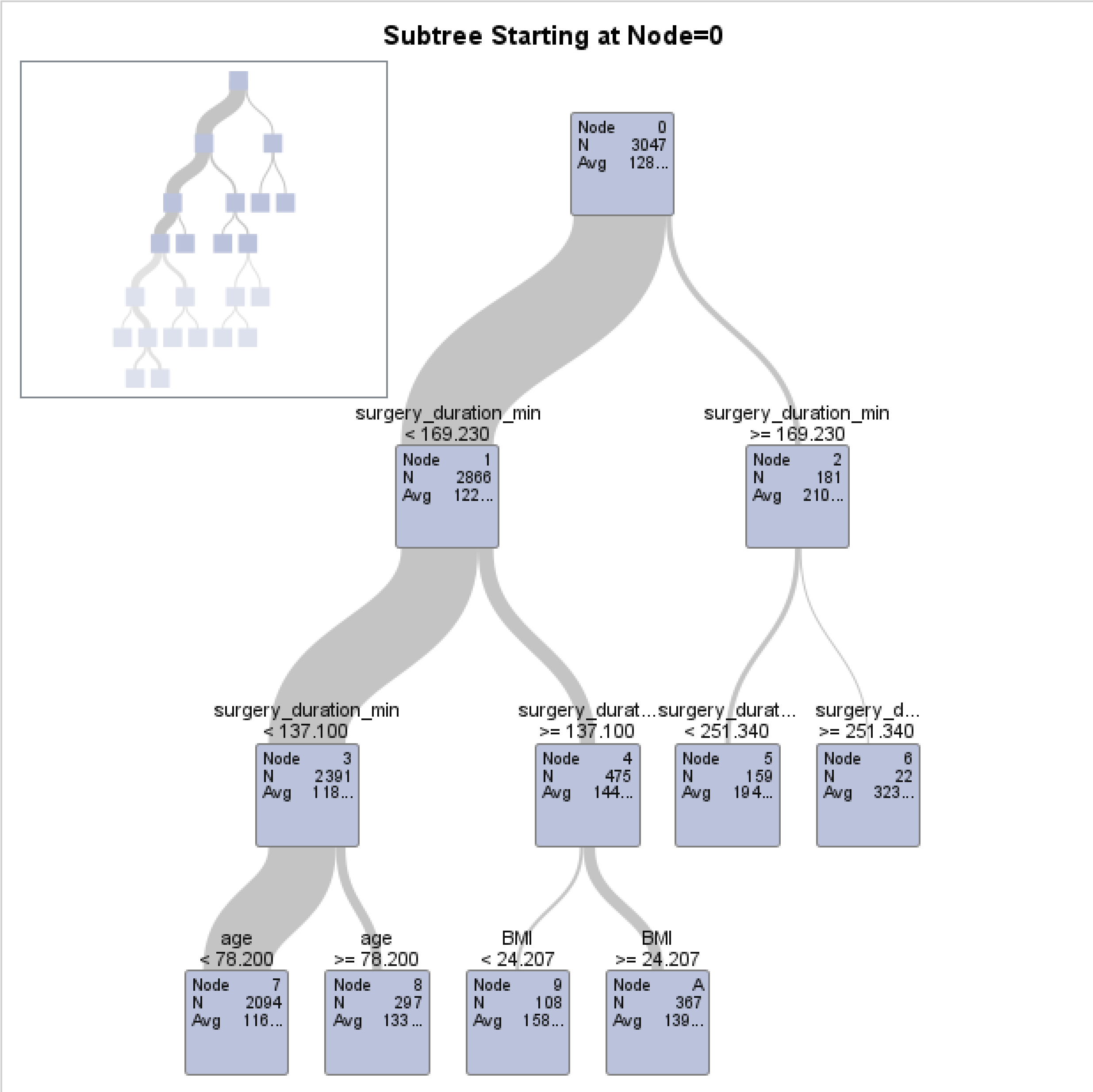
Model Information

Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	12

Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 3047
Avg 128...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 122...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 210...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 118...
surgery_durat...
>= 137.100
Node 4
N 475
Avg 144...
surgery_durat...
< 251.340
Node 5
N 159
Avg 194...
surgery_d...
>= 251.340
Node 6
N 22
Avg 323...
age
< 78.200
Node 7
N 2094
Avg 116...
age
>= 78.200
Node 8
N 297
Avg 133...
BMI
< 24.207
Node 9
N 108
Avg 158...
BMI
>= 24.207
Node A
N 367
Avg 139...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
12	7415512	2.26E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	138663	5
age	0.2318	32141.5	4
BMI	0.1203	16676.8	1
ASA	0.0874	12112.9	1

accuracy10	accuracy15	accuracy20
0.51117	0.660972	0.78318

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

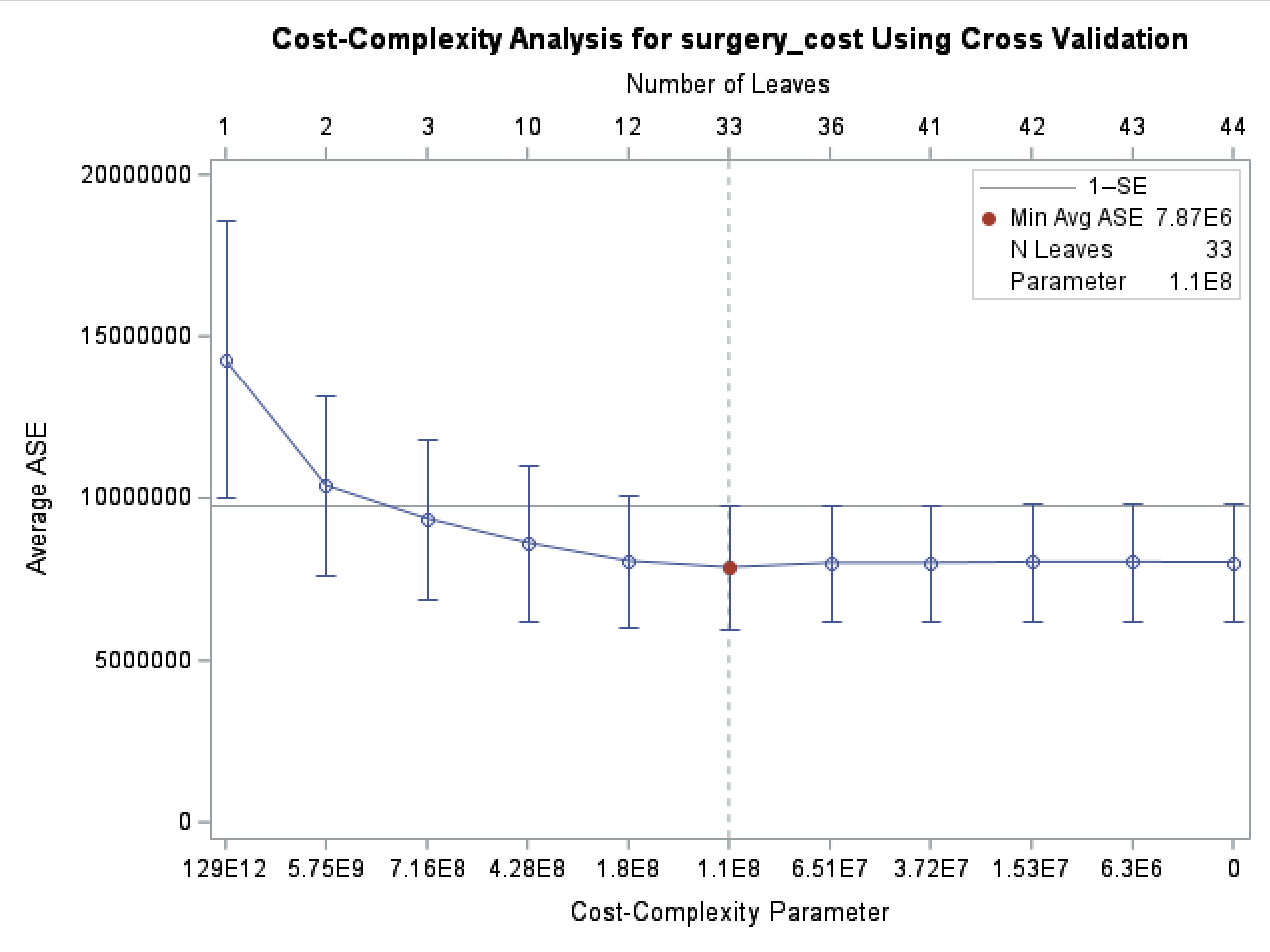
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

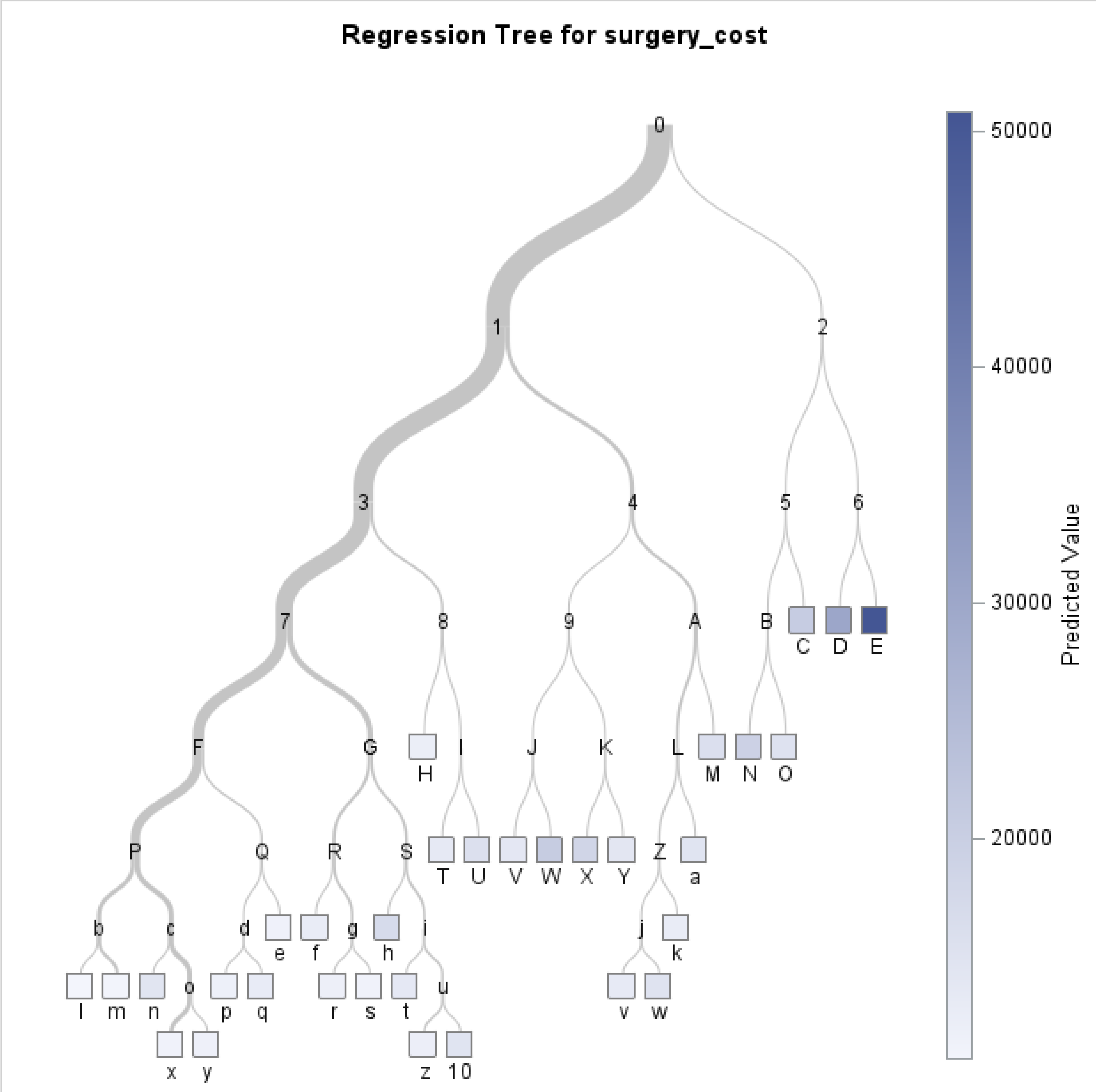
Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	32

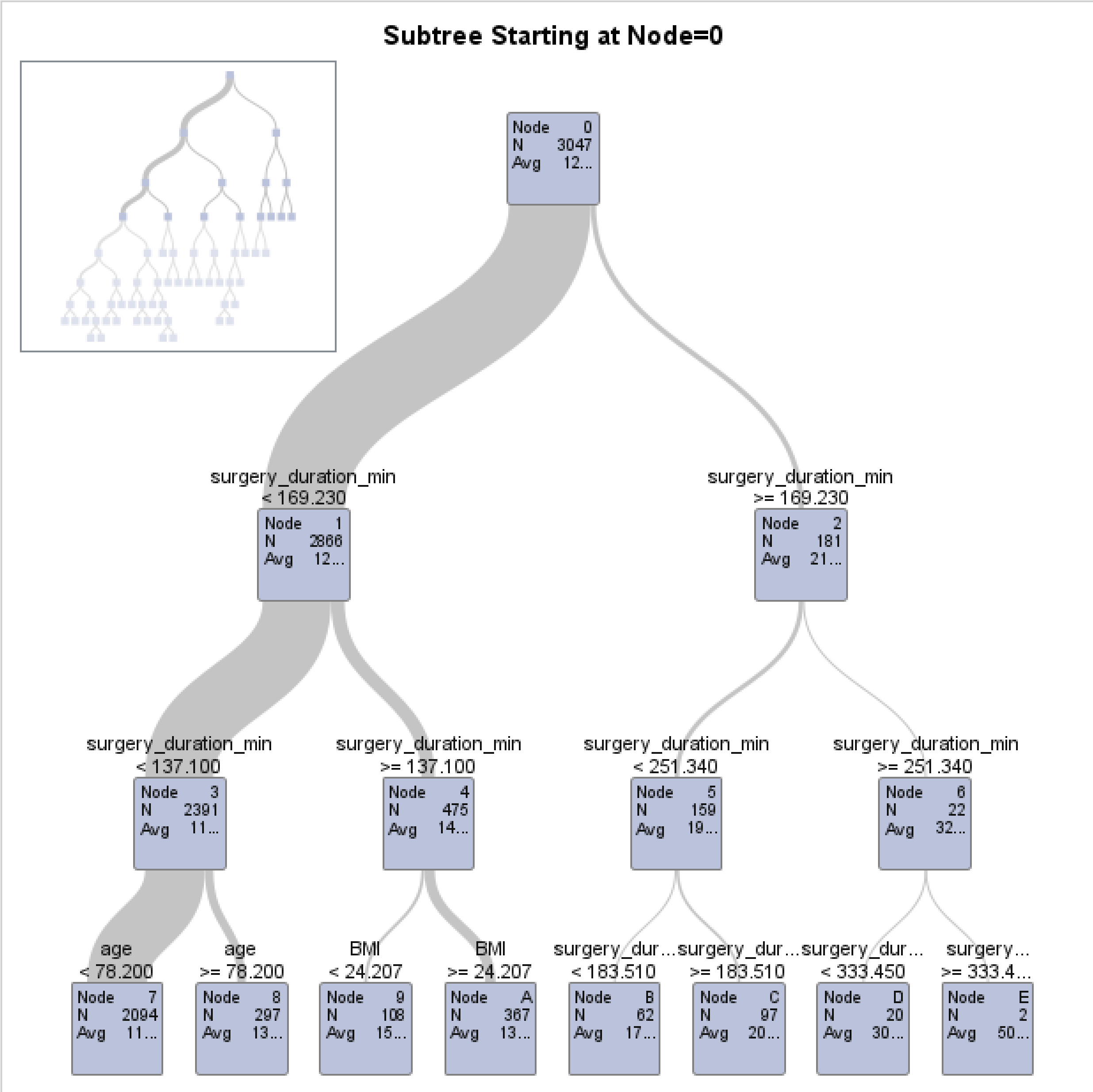
Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure



The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_duration_min
< 251.340
Node 5
N 159
Avg 19...
surgery_duration_min
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...
surgery_dur...
< 333.450
Node D
N 20
Avg 30...
surgery...
>= 333.4...
Node E
N 2
Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
32	6469173	1.971E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2588	37412.2	9
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

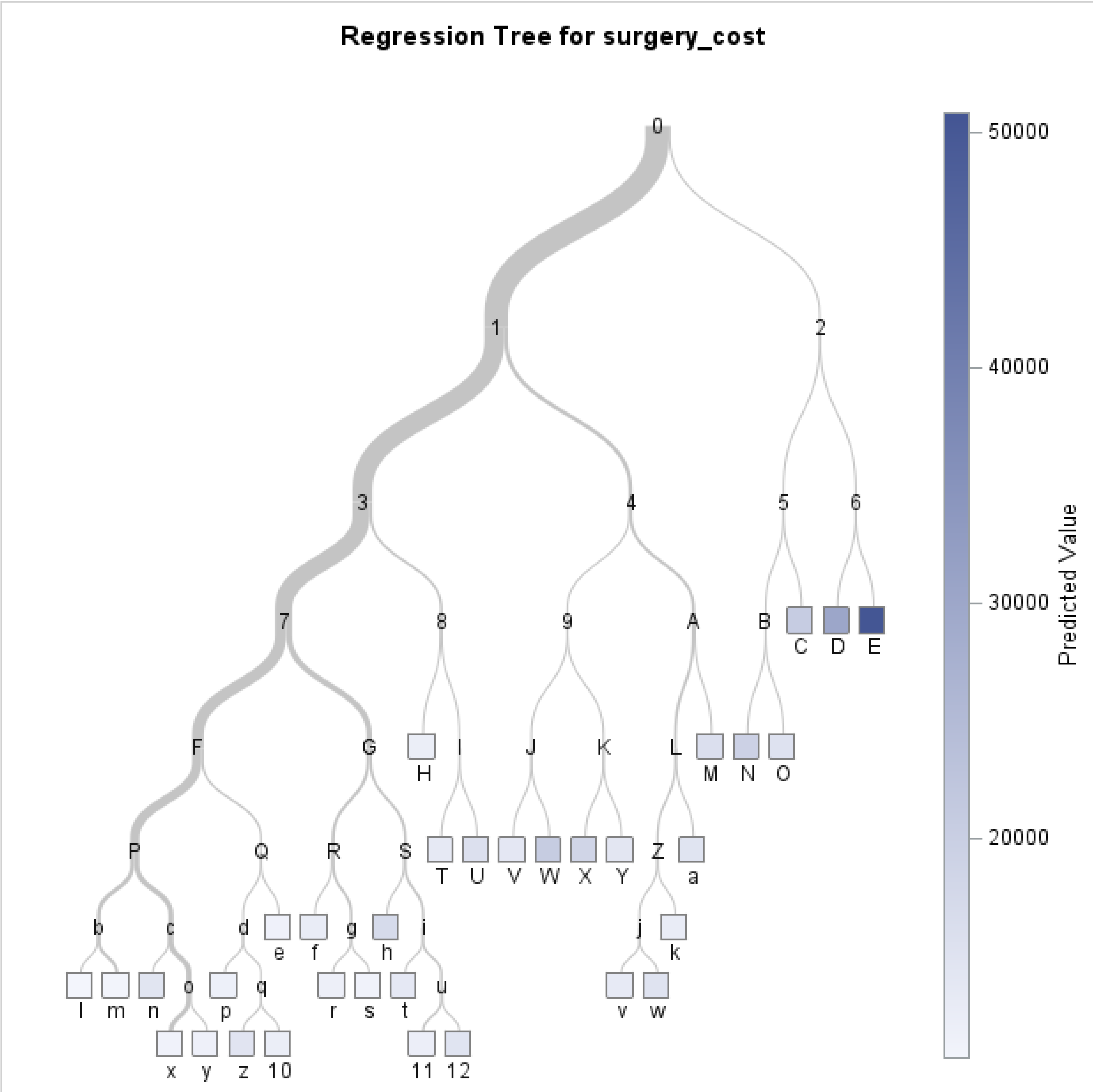
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

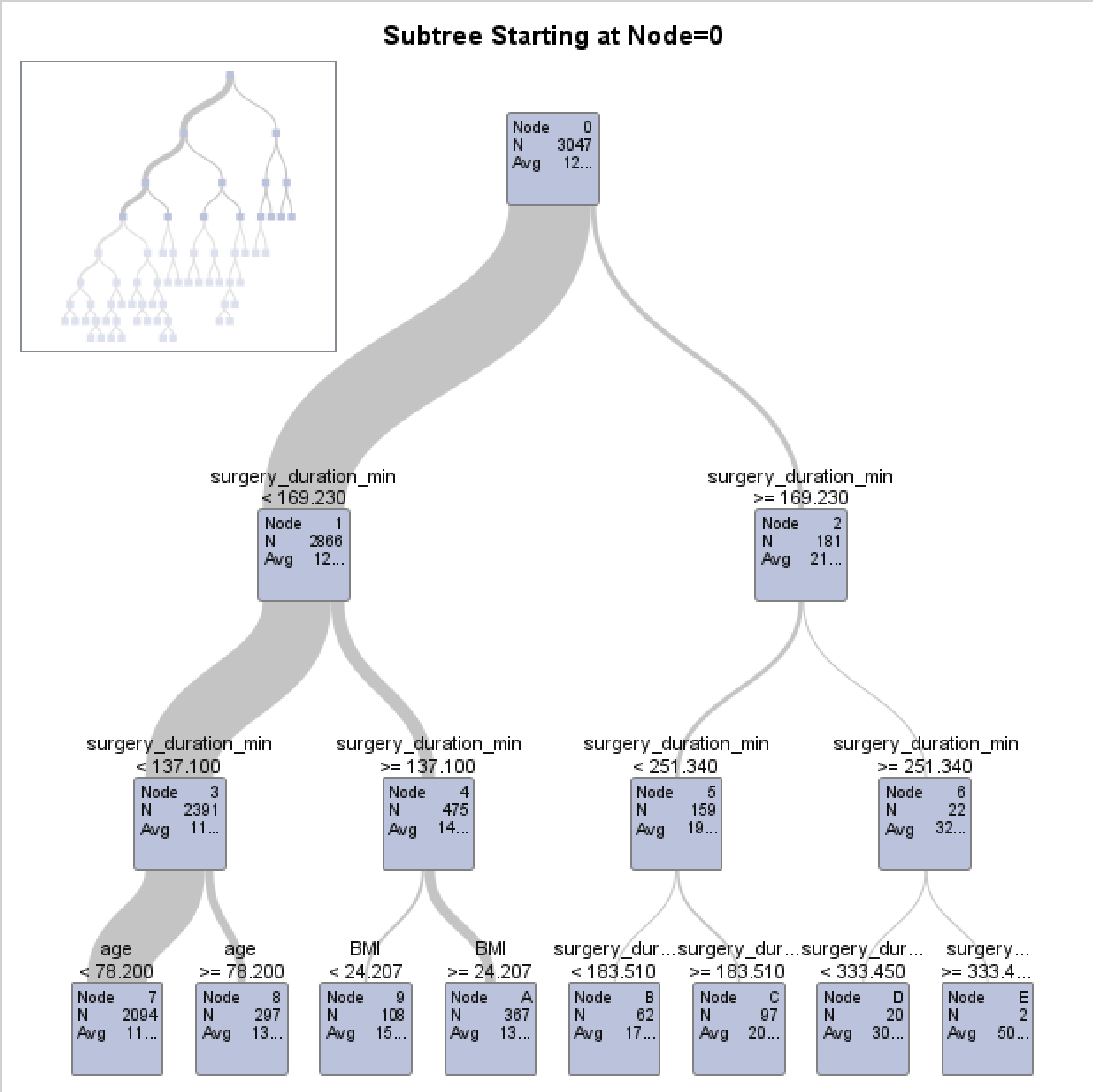
Model Information

Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	33

Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0 N 3047 Avg 12...
surgery_duration_min < 169.230
Node 1 N 2866 Avg 12...
surgery_duration_min >= 169.230
Node 2 N 181 Avg 21...
surgery_duration_min < 137.100
Node 3 N 2391 Avg 11...
surgery_duration_min >= 137.100
Node 4 N 475 Avg 14...
surgery_duration_min < 251.340
Node 5 N 159 Avg 19...
surgery_duration_min >= 251.340
Node 6 N 22 Avg 32...
age < 78.200
Node 7 N 2094 Avg 11...
age >= 78.200
Node 8 N 297 Avg 13...
BMI < 24.207
Node 9 N 108 Avg 15...
BMI >= 24.207
Node A N 367 Avg 13...
surgery_dur... < 183.510
Node B N 62 Avg 17...
surgery_dur... >= 183.510
Node C N 97 Avg 20...
surgery_dur... < 333.450
Node D N 20 Avg 30...
surgery... >= 333.4...
Node E N 2 Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
33	6449015	1.965E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2645	38224.2	10
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

accuracy10	accuracy15	accuracy20
0.507227	0.670171	0.805519

R Code

```
library(readr)
library(rpart)
library(rpart.plot)
library(dplyr)
library(partykit)
library(CHAID)

hospital_data =
read.csv(file="C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/hospital_data
.csv",
header=T, sep=",")

# (a) Splitting data into 80% training and 20% testing sets and building
# a regression tree on the training set using the RSS Splitting Criterion
# to model surgery cost.

set.seed(257496)
sample = sample(c(T,F), nrow(hospital_data),
replace=T, prob=c(0.8, 0.2))
train = hospital_data[sample,]
test = hospital_data[!sample,]

reg_tree_full = rpart(surgery_cost~gender+age+BMI+ASA
+ surgery_duration_min, data=train, method="anova", xval=10, cp=0)

printcp(reg_tree_full)

# Fitting regression tree with RSS Splitting and cost-complexity pruning

reg_tree_RSS = rpart(surgery_cost~gender+age+BMI+ASA
+ surgery_duration_min, data=train, method="anova",
cp=0.0041801)

rpart.plot(reg_tree_RSS, type=3)
```

```r
# Computing prediction accuracy for testing data within 10%, 15%, and
# 20%.

P_surgery_cost = predict(reg_tree_RSS, newdata=test)

# Accuracy within 10%

accuracy10 = ifelse(abs(test$surgery_cost-P_surgery_cost)<0.10*test$surgery_cost,
1, 0)
print(mean(accuracy10))

# Accuracy within 15%

accuracy15 = ifelse(abs(test$surgery_cost-P_surgery_cost)<0.15*test$surgery_cost,
1, 0)
print(mean(accuracy15))

# Accuracy within 20%

accuracy20 = ifelse(abs(test$surgery_cost-P_surgery_cost)<0.20*test$surgery_cost,
1, 0)
print(mean(accuracy20))

# Fitting regression tree with CHAID Splitting Criterion and cost-
# complexity pruning.

# Binning continuous predictor variables.

hospital_data = mutate(hospital_data, gender_cat = ntile(gender, 10),
age_cat = ntile(age, 10), BMI_cat = ntile(BMI, 10), ASA_cat = ntile(ASA, 10),
surgery_duration_min_cat = ntile(surgery_duration_min, 10),
surgery_cost_cat = ntile(surgery_cost, 10))

set.seed(233364)
sample = sample(c(T,F), nrow(hospital_data), replace=T,
prob=c(0.8, 0.2))
train = hospital_data[sample,]
test = hospital_data[!sample,]

reg_tree_CHAID = chaid(as.factor(surgery_cost_cat)~as.factor(gender_cat)+
as.factor(age_cat)+as.factor(BMI_cat)+as.factor(ASA_cat)+
as.factor(surgery_duration_min_cat), data=train,
control = chaid_control(maxheight=4))

plot(reg_tree_CHAID, type="simple")
```

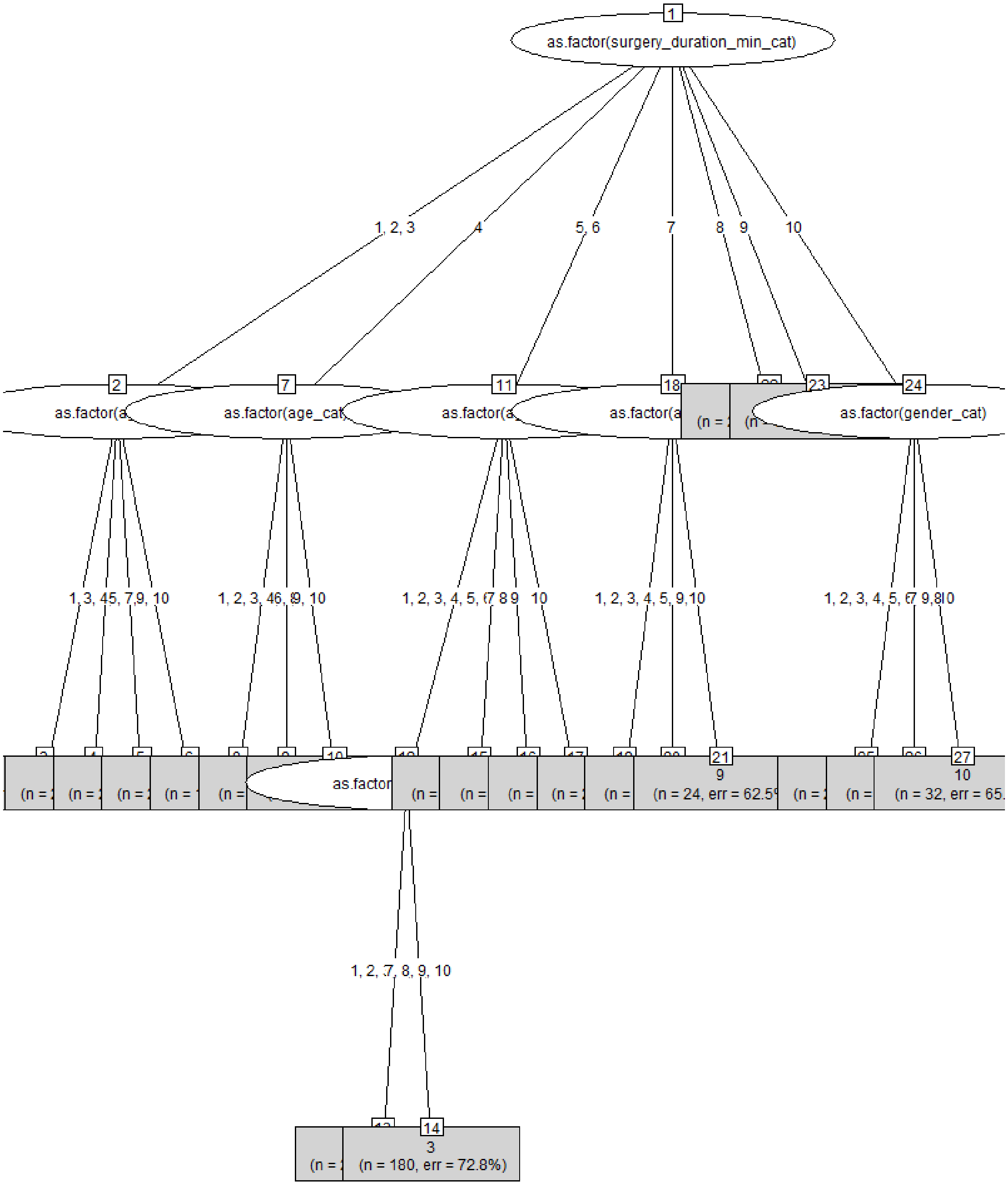
```r
# Computing prediction accuracy for testing data for CHAID regression
# tree

predclass = as.numeric(predict(reg_tree_CHAID, newdata=test))
test = cbind(test, predclass)

aggr_data = aggregate(train$surgery_cost, by=list(train$surgery_cost_cat),
FUN=mean)
aggr_data$predclass = aggr_data$Group.1
aggr_data$P_surgery_cost = aggr_data$x
test = left_join(test, aggr_data, by='predclass')

# Accuracy within 10%

accuracy10 = ifelse(abs(test$surgery_cost-
test$P_surgery_cost)<0.10*test$surgery_cost, 1, 0)
print(mean(accuracy10))

# Accuracy within 15%

accuracy15 = ifelse(abs(test$surgery_cost-
test$P_surgery_cost)<0.15*test$surgery_cost, 1, 0)
print(mean(accuracy15))

# Accuracy within 20%

accuracy20 = ifelse(abs(test$surgery_cost-
test$P_surgery_cost)<0.20*test$surgery_cost, 1, 0)
print(mean(accuracy20))
```

```
Variables actually used in tree construction:
[1] age                  ASA                  BMI
[4] gender               surgery_duration_min

Root node error: 4.9394e+10/3070 = 16089359

n= 3070

           CP nsplit rel error  xerror     xstd
1  3.4460e-01      0   1.00000 1.00028 0.084402
2  8.1891e-02      1   0.65540 0.67242 0.049341
3  5.2408e-02      2   0.57351 0.61736 0.043263
4  2.0349e-02      3   0.52111 0.55893 0.040430
5  1.5282e-02      4   0.50076 0.54003 0.038197
6  1.2063e-02      5   0.48548 0.50796 0.035432
7  8.5942e-03      6   0.47341 0.51202 0.035776
8  7.1489e-03      7   0.46482 0.51456 0.037009
9  7.0144e-03      8   0.45767 0.51470 0.036670
10 5.6976e-03      9   0.45065 0.50843 0.036883
```

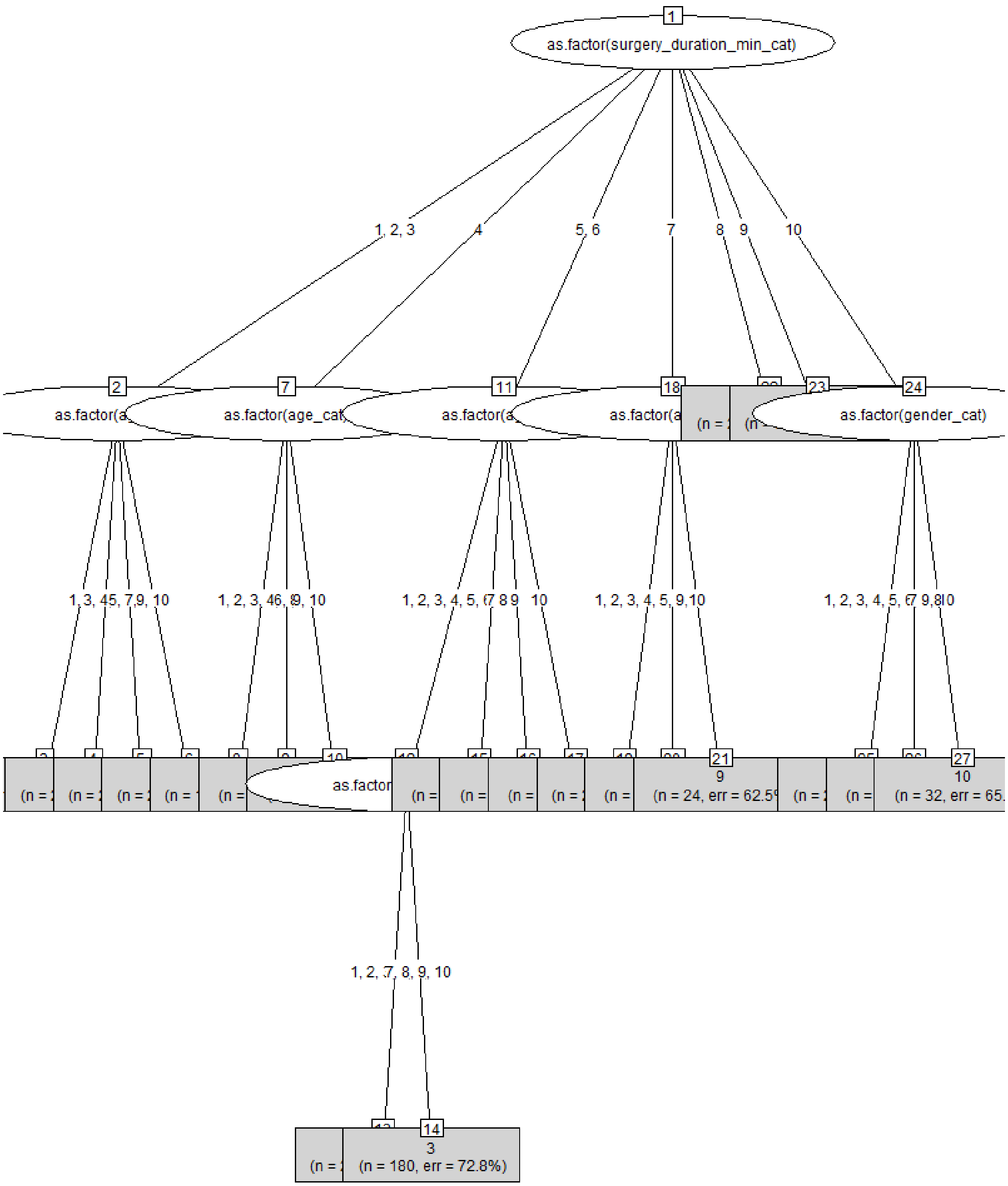
11	5.3183e-03	10	0.44496	0.50693	0.037080
12	4.1801e-03	11	0.43964	0.50424	0.036486
13	4.0237e-03	12	0.43546	0.50419	0.036505
14	3.7016e-03	14	0.42741	0.50204	0.036165
15	3.0901e-03	15	0.42371	0.50002	0.036307
16	2.8110e-03	17	0.41753	0.49109	0.034900
17	2.7748e-03	18	0.41472	0.49023	0.034534
18	2.6997e-03	19	0.41194	0.49046	0.034562
19	2.4311e-03	20	0.40924	0.49585	0.034867
20	2.1778e-03	21	0.40681	0.49819	0.035122
21	2.1687e-03	23	0.40246	0.49932	0.035284
22	1.6737e-03	24	0.40029	0.50054	0.035672
23	1.6214e-03	26	0.39694	0.50400	0.035692
24	1.6064e-03	27	0.39532	0.50363	0.035701
25	1.5628e-03	28	0.39371	0.50386	0.035654
26	1.4039e-03	29	0.39215	0.50664	0.035705
27	1.3224e-03	30	0.39075	0.51116	0.035919
28	1.2317e-03	33	0.38678	0.51370	0.036652
29	1.2241e-03	35	0.38432	0.51214	0.036613
30	1.1468e-03	36	0.38309	0.51142	0.036604
31	1.1372e-03	37	0.38195	0.51209	0.036606
32	1.0455e-03	38	0.38081	0.51204	0.036609
33	1.0338e-03	39	0.37976	0.51344	0.036605
34	1.0222e-03	40	0.37873	0.51365	0.036615
35	1.0172e-03	41	0.37771	0.51536	0.036670
36	9.7626e-04	42	0.37669	0.51475	0.036649
37	9.3689e-04	43	0.37571	0.51681	0.036667
38	8.9855e-04	44	0.37478	0.51686	0.036571
39	8.5317e-04	46	0.37298	0.51697	0.036515
40	7.9216e-04	47	0.37213	0.51972	0.036526
41	7.8172e-04	49	0.37054	0.52180	0.036535
42	7.6915e-04	50	0.36976	0.52082	0.036386
43	7.0989e-04	52	0.36822	0.52105	0.036381
44	6.3868e-04	54	0.36680	0.52216	0.036360
45	6.2630e-04	56	0.36552	0.52491	0.036391
46	6.0939e-04	58	0.36427	0.52578	0.036393
47	5.9214e-04	59	0.36366	0.52537	0.036392
48	5.7795e-04	60	0.36307	0.52614	0.036416
49	5.4176e-04	62	0.36191	0.52586	0.036390
50	5.0977e-04	63	0.36137	0.52790	0.036434
51	5.0718e-04	65	0.36035	0.52740	0.036443
52	4.9234e-04	68	0.35883	0.52806	0.036442
53	4.5825e-04	70	0.35785	0.52943	0.036445
54	4.4222e-04	71	0.35739	0.52854	0.036429
55	4.3833e-04	72	0.35695	0.52865	0.036430
56	4.3735e-04	74	0.35607	0.52786	0.036424
57	4.0912e-04	75	0.35563	0.52780	0.036423
58	3.8688e-04	78	0.35441	0.52854	0.036418
59	3.7557e-04	79	0.35402	0.52960	0.036377
60	3.6938e-04	81	0.35327	0.52945	0.036377
61	3.6450e-04	82	0.35290	0.52916	0.036377
62	3.5133e-04	83	0.35253	0.52897	0.036373
63	3.4822e-04	85	0.35183	0.52909	0.036319
64	3.4156e-04	86	0.35148	0.52941	0.036321
65	3.2528e-04	87	0.35114	0.53032	0.036381
66	3.1955e-04	91	0.34981	0.53174	0.036444
67	3.1694e-04	92	0.34949	0.53216	0.036438
68	3.1544e-04	93	0.34918	0.53216	0.036477
69	3.0432e-04	94	0.34886	0.53220	0.036476
70	2.9713e-04	100	0.34703	0.53208	0.036477
71	2.8644e-04	101	0.34674	0.53213	0.036453
72	2.8293e-04	102	0.34645	0.53187	0.036450
73	2.8217e-04	103	0.34617	0.53225	0.036449
74	2.8005e-04	106	0.34532	0.53240	0.036450

75	2.7520e-04	107	0.34504	0.53259	0.036468
76	2.7475e-04	108	0.34477	0.53263	0.036468
77	2.7131e-04	109	0.34449	0.53279	0.036468
78	2.5489e-04	110	0.34422	0.53396	0.036622
79	2.5435e-04	111	0.34396	0.53443	0.036612
80	2.5139e-04	112	0.34371	0.53439	0.036612
81	2.5069e-04	113	0.34346	0.53442	0.036612
82	2.3533e-04	114	0.34321	0.53413	0.036527
83	2.3365e-04	119	0.34203	0.53487	0.036542
84	2.3138e-04	120	0.34180	0.53514	0.036541
85	2.3097e-04	121	0.34156	0.53525	0.036541
86	2.2061e-04	122	0.34133	0.53543	0.036541
87	2.1534e-04	125	0.34067	0.53591	0.036543
88	2.1051e-04	128	0.34003	0.53627	0.036543
89	2.1007e-04	129	0.33982	0.53642	0.036543
90	1.9909e-04	130	0.33961	0.53667	0.036544
91	1.9722e-04	132	0.33921	0.53649	0.036494
92	1.9677e-04	133	0.33901	0.53647	0.036494
93	1.8445e-04	134	0.33881	0.53675	0.036493
94	1.8436e-04	138	0.33808	0.53662	0.036496
95	1.8164e-04	139	0.33789	0.53725	0.036578
96	1.7606e-04	142	0.33735	0.53743	0.036575
97	1.7187e-04	143	0.33717	0.53792	0.036570
98	1.7009e-04	145	0.33683	0.53822	0.036580
99	1.7003e-04	146	0.33666	0.53822	0.036580
100	1.6118e-04	147	0.33649	0.53838	0.036590
101	1.6030e-04	148	0.33632	0.53833	0.036590
102	1.5866e-04	149	0.33616	0.53856	0.036590
103	1.5838e-04	150	0.33601	0.53862	0.036590
104	1.4973e-04	151	0.33585	0.53878	0.036587
105	1.4099e-04	152	0.33570	0.53885	0.036602
106	1.3625e-04	154	0.33542	0.53874	0.036588
107	1.3178e-04	155	0.33528	0.53847	0.036586
108	1.2466e-04	156	0.33515	0.53773	0.036435
109	1.2205e-04	157	0.33502	0.53818	0.036435
110	1.2177e-04	158	0.33490	0.53798	0.036434
111	1.2082e-04	159	0.33478	0.53815	0.036434
112	1.1849e-04	160	0.33466	0.53836	0.036425
113	1.1277e-04	161	0.33454	0.53871	0.036424
114	1.1185e-04	162	0.33443	0.53871	0.036423
115	1.1183e-04	163	0.33432	0.53875	0.036423
116	1.0934e-04	165	0.33409	0.53871	0.036423
117	1.0249e-04	166	0.33398	0.53892	0.036417
118	1.0125e-04	169	0.33366	0.53907	0.036416
119	9.9164e-05	171	0.33346	0.53905	0.036416
120	9.8439e-05	173	0.33326	0.53906	0.036416
121	9.4021e-05	175	0.33306	0.53889	0.036417
122	9.3674e-05	178	0.33278	0.53924	0.036423
123	8.9571e-05	179	0.33269	0.53927	0.036423
124	8.4962e-05	181	0.33251	0.53941	0.036423
125	8.2773e-05	182	0.33242	0.53986	0.036425
126	8.1656e-05	184	0.33226	0.54005	0.036428
127	8.1593e-05	185	0.33218	0.54008	0.036427
128	8.0806e-05	186	0.33209	0.54034	0.036427
129	7.8094e-05	191	0.33164	0.54019	0.036375
130	7.7798e-05	192	0.33156	0.54049	0.036374
131	7.6532e-05	193	0.33148	0.54065	0.036376
132	7.6109e-05	194	0.33141	0.54065	0.036376
133	7.5312e-05	195	0.33133	0.54064	0.036364
134	7.3788e-05	196	0.33126	0.54080	0.036363
135	7.3477e-05	197	0.33118	0.54083	0.036363
136	7.1498e-05	198	0.33111	0.54088	0.036363
137	7.1196e-05	200	0.33097	0.54097	0.036363
138	7.0012e-05	202	0.33082	0.54105	0.036363

139	6.7273e-05	203	0.33075	0.54113	0.036365
140	6.6158e-05	207	0.33048	0.54119	0.036364
141	6.4212e-05	208	0.33042	0.54076	0.036349
142	6.2878e-05	209	0.33035	0.54095	0.036348
143	6.1327e-05	211	0.33023	0.54151	0.036383
144	5.9292e-05	212	0.33017	0.54166	0.036384
145	5.7370e-05	213	0.33011	0.54194	0.036383
146	5.7260e-05	214	0.33005	0.54207	0.036385
147	5.5878e-05	215	0.32999	0.54210	0.036385
148	5.4963e-05	218	0.32982	0.54216	0.036385
149	5.2299e-05	220	0.32971	0.54199	0.036385
150	5.2226e-05	221	0.32966	0.54208	0.036385
151	5.0799e-05	223	0.32956	0.54214	0.036385
152	4.9376e-05	224	0.32951	0.54201	0.036385
153	4.6195e-05	225	0.32946	0.54225	0.036385
154	4.4463e-05	226	0.32941	0.54239	0.036383
155	4.3040e-05	228	0.32932	0.54244	0.036383
156	4.2629e-05	229	0.32928	0.54244	0.036383
157	4.1265e-05	230	0.32924	0.54243	0.036383
158	3.6874e-05	231	0.32920	0.54246	0.036383
159	3.5397e-05	232	0.32916	0.54252	0.036383
160	3.4466e-05	233	0.32912	0.54255	0.036383
161	3.4013e-05	234	0.32909	0.54257	0.036382
162	3.3655e-05	235	0.32905	0.54257	0.036382
163	3.3305e-05	236	0.32902	0.54258	0.036382
164	3.2307e-05	237	0.32899	0.54260	0.036382
165	3.1775e-05	238	0.32896	0.54259	0.036383
166	2.9990e-05	239	0.32892	0.54255	0.036383
167	2.6159e-05	240	0.32889	0.54282	0.036390
168	2.2858e-05	241	0.32887	0.54286	0.036389
169	1.9384e-05	242	0.32884	0.54290	0.036388
170	1.7823e-05	243	0.32883	0.54294	0.036389
171	1.6577e-05	244	0.32881	0.54302	0.036389
172	1.5195e-05	245	0.32879	0.54306	0.036389
173	1.4478e-05	246	0.32878	0.54316	0.036390
174	1.3950e-05	249	0.32873	0.54316	0.036390
175	5.9236e-06	250	0.32872	0.54317	0.036390
176	0.0000e+00	251	0.32871	0.54318	0.036390


1
as.factor(surgery_duration_min_cat)
1, 2, 3
4
5, 6
7
8
9
10
2
7
11
18
23
24
as.factor(age_cat)
as.factor(gender_cat)
21
9
(n = 24, err = 62.5
27
10
(n = 32, err = 65.
14
3
(n = 180, err = 72.8%)

0.4850949
0.7059621
0.8346883



1
as.factor(surgery_duration_min_cat)
1, 2, 3
4
5, 6
7
8
9
10
2
7
11
18
23
24
as.factor(age_cat)
as.factor(gender_cat)
21
9
(n = 24, err = 62.5
27
10
(n = 32, err = 65.
14
3
(n = 180, err = 72.8%)

```
0.4018568
0.5596817
0.6923077
```

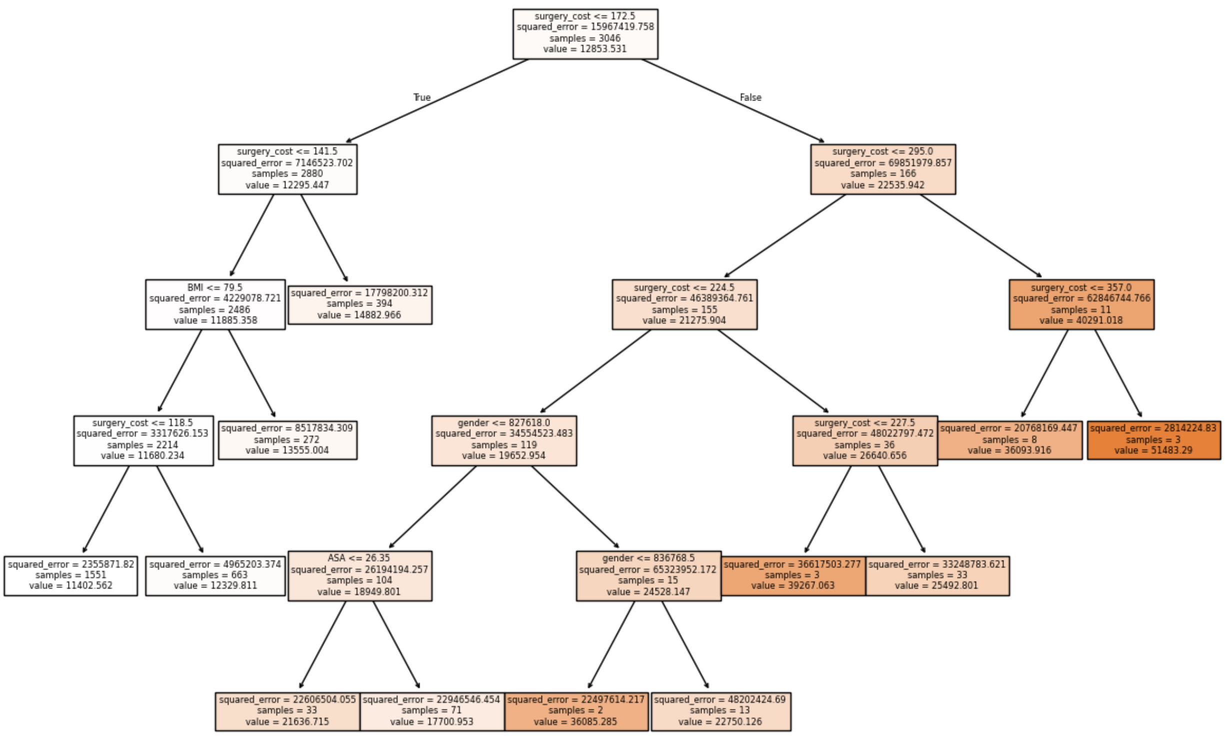
Python Code

```
# STAT 574 HW 1 Code problem 1 (Python Version)

# Import all necessary libraries.

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from chefboost import Chefboost

# Problem 1: Hospital Data

# Importing and preprocessing dataset

path_directory = "C:/Users/coryg/OneDrive/Desktop/\
STAT_574_Data_Mining/hospital_data.csv"
hospital_data = pd.read_csv(path_directory)
gender_code = {'M':1, 'F':0}
hospital_data['gender'] = hospital_data['gender'].map(gender_code)

X = hospital_data.iloc[:,0:6].values
y = hospital_data.iloc[:,6].values

# (a) Splitting data into 80% training and 20% testing sets
# and building Regression Tree with RSS Splitting Critierion
# to model surgery cost. Applying cost complexity pruning.

X_train, X_test, y_train, y_test = train_test_split(X,y,
                                    test_size=0.20,
                                    random_state=257496)

hospital_reg_tree = DecisionTreeRegressor(random_state=820101,
                                criterion="squared_error",
                                max_leaf_nodes=12)

hospital_reg_fit = hospital_reg_tree.fit(X_train, y_train)

fig = plt.figure(figsize=(15,10))
```

```
fn = ['gender', 'age', 'BMI', 'ASA', 'surgery_duration_min',
      'surgery_cost']
tree.plot_tree(hospital_reg_fit, feature_names=fn, filled=True)

# (b) Using fitted RSS Regression Tree to predict surgery cost
# on the testing data. Computing proportions of predicted values
# within 10%, 15%, and 20% of observed values.

y_pred = hospital_reg_fit.predict(X_test)
ind10 = []
ind15 = []
ind20 = []

for sub1, sub2 in zip(y_pred, y_test):
    ind10.append(1) if abs(sub1-sub2)<0.10*sub2 else ind10.append(0)
    ind15.append(1) if abs(sub1-sub2)<0.15*sub2 else ind15.append(0)
    ind20.append(1) if abs(sub1-sub2)<0.20*sub2 else ind20.append(0)

# Accuracies within 10%, 15%, and 20% respectively

accuracy10 = sum(ind10)/len(ind10)
print(accuracy10)
accuracy15 = sum(ind15)/len(ind15)
print(accuracy15)
accuracy20 = sum(ind20)/len(ind20)
print(accuracy20)

# (c) Building a regression tree on the training data using
# the CHAID Splitting Criterion and cost-complexity pruning.

#Splitting response variable into deciles and making them nominal.

hospital_data['deciles'] = pd.qcut(hospital_data['surgery_cost'], 10,
                                   labels=False)
deciles_coding = {0:'0th', 1:'1st', 2:'2nd', 3:'3rd', 4:'4th',
                  5:'5th', 6:'6th', 7:'7th', 8:'8th', 9:'9th'}
hospital_data['deciles'] = hospital_data['deciles'].map(deciles_coding)

X = hospital_data.iloc[:, 0:6].values
y = hospital_data.iloc[:, 6:9].values

#Splitting data into 80% training and 20% testing sets.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,
                                                    random_state=233364)
```

```python
X_train = pd.DataFrame(X_train, columns=['MedID', 'gender', 'age',
                                         'BMI', 'ASA', 'surgery_duration_min'])
y_train = pd.DataFrame(y_train[:,1], columns=['deciles'])
train_data = pd.concat([X_train, y_train], axis=1)

#Fitting tree

config = {'algorithm': 'CHAID', 'max_depth':4}
tree_chaid = Chefboost.fit(train_data, config, target_label='deciles')

# (d) using the CHAID regression tree to predict surgery cost
# on the testing set. Computing the proportion of predicted
# values within 10%, 15%, and 20% of observed values.

X_test = pd.DataFrame(X_test, columns=['MedID', 'gender', 'age', 'BMI',
                                       'ASA', 'surgery_duration_min'])
y_pred = []
for i in range(len(y_test)):
    y_pred.append(Chefboost.predict(tree_chaid, X_test.iloc[i,:]))

#Computing prediction accuracy for testing data

y_test = pd.DataFrame(y_test[:,0], columns=['surgery_cost'])
y_pred = pd.DataFrame(y_pred, columns=['predclass'])
pred_data = pd.concat([y_test, y_pred], axis=1)

df_new = pred_data.groupby('predclass')['surgery_cost'].mean()
inner_join = pd.merge(pred_data, df_new, on='predclass', how='inner')

ind10 = []
ind15 = []
ind20 = []

for sub1, sub2 in zip(inner_join['surgery_cost_x'],
                      inner_join['surgery_cost_y']):
    ind10.append(1) if abs(sub1-sub2)<0.10*sub1 else ind10.append(0)
    ind15.append(1) if abs(sub1-sub2)<0.15*sub1 else ind15.append(0)
    ind20.append(1) if abs(sub1-sub2)<0.20*sub1 else ind20.append(0)

#accuracy within 10%

accuracy10 = sum(ind10)/len(ind10)
print(accuracy10)

#accuracy within 15%
```

```
accuracy15 = sum(ind15)/len(ind15)
print(accuracy15)

#accuracy within 20%

accuracy20 = sum(ind20)/len(ind20)
print(accuracy20)
```



RSS splitting criterion regression tree

Accuracy within 10%: 0.442257217847769

Accuracy within 15%: 0.6548556430446194

Accuracy within 20%: 0.7860892388451444

CHAID splitting criterion regression tree

Accuracy within 10%: 0.4225721784776903

Accuracy within 15%: 0.5971128608923885

Accuracy within 20%: 0.7099737532808399

According to all three codes, the RSS Splitting Criterion regression tree yielded the highest accuracies within 10%, 15%, and 20% of the actual values in the testing set. Therefore we conclude that this regression tree was overall the best in performance.

Problem 2.

SAS Code

```
proc import out=card_data
datafile="C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/card_transdata.csv
"
dbms=csv replace;

/* (a) Splitting the data into 80% training and 20% testing sets*/

proc surveyselect data=card_data rate=0.8 seed=122470
out=card_data outall method=srs;
run;

/*Gini-splitting and cost-complexity pruning*/

proc hpsplit data=card_data maxdepth=7;
    class repeat_retailer used_chip used_pin_number online_order fraud;
    model fraud(event="1")=distance_from_home distance_from_last_transaction
ratio_to_median_purchase_price repeat_retailer used_chip used_pin_number
online_order;
    grow gini;
    prune costcomplexity;
    partition rolevar=selected(train="1");
    output out=predicted;
    ID selected;
run;

/* (b)Computing prediction accuracy for testing set for Gini Tree*/

data test;
    set predicted;
    if(selected="0");
    keep fraud P_fraud;
run;

data cutoffs;
    set test;
```

```
    do i=1 to 99;
    tp = (P_fraudyes > 0.01*i and fraud="1");
    tn = (P_fraudyes < 0.01*i and fraud="0");
    output;
    end;
run;

proc sql;
    create table rates as
    select i, sum(tp+tn)/count(*) as trueclassrate
    from cutoffs
    group by i;
    select 0.01*i as cutoff, trueclassrate
    from rates
    having trueclassrate=max(trueclassrate);
quit;

/* (c) Fitting binary classification tree using entropy splitting */
/* and cost-complexity pruning algorithm*/

proc hpsplit data=card_data maxdepth=7;
    class repeat_retailer used_chip used_pin_number online_order fraud;
    model fraud(event="1") = distance_from_home distance_from_last_transaction
ratio_to_median_purchase_price repeat_retailer used_chip used_pin_number
online_order;
    grow entropy;
    prune costcomplexity;
    partition rolevar=selected(train="1");
    output out=predicted2;
    ID selected;
run;

/* (d) Computing prediction accuracy for testing set for entropy */
/*splitting tree*/

data test2;
    set predicted2;
    if(selected="0");
    keep fraud P_fraud;
run;

data cutoffs2;
    set test2;
    do i=1 to 99;
    tp = (P_fraudyes > 0.01*i and fraud="1");
```

```
    tn = (P_fraudyes < 0.01*i and fraud="0");
    output;
    end;
run;

proc sql;
    create table rates2 as
    select i, sum(tp+tn)/count(*) as trueclassrate
    from cutoffs2
    group by i;
    select 0.01*i as cutoff, trueclassrate
    from rates2
    having trueclassrate=max(trueclassrate);
quit;

/* (e) CHAID splitting and cost-complexity pruning*/

proc hpsplit data=card_data maxdepth=7;
    class repeat_retailer used_chip used_pin_number online_order fraud;
    model fraud(event="1") = distance_from_home distance_from_last_transaction
ratio_to_median_purchase_price repeat_retailer used_chip used_pin_number
online_order;
    grow CHAID;
    prune costcomplexity;
    partition rolevar=selected(train="1");
    output out=predicted3;
    ID selected;
run;

/* (f) Computing prediction accuracy for testing set for CHAID */
/*splitting tree*/

data test3;
    set predicted3;
    if(selected="0");
    keep fraud P_fraud;
run;

data cutoffs3;
    set test3;
    do i=1 to 99;
    tp = (P_fraudyes > 0.01*i and fraud="1");
    tn = (P_fraudyes < 0.01*i and fraud="0");
    output;
    end;
```

```
run;

proc sql;
    create table rates3 as
    select i, sum(tp+tn)/count(*) as trueclassrate
    from cutoffs3
    group by i;
    select 0.01*i as cutoff, trueclassrate
    from rates3
    having trueclassrate=max(trueclassrate);
quit;
```

The SAS System

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	HOSPITAL
Random Number Seed	479576
Sampling Rate	0.8
Sample Size	3047
Selection Probability	0.800158
Sampling Weight	0
Output Data Set	HOSPITAL

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

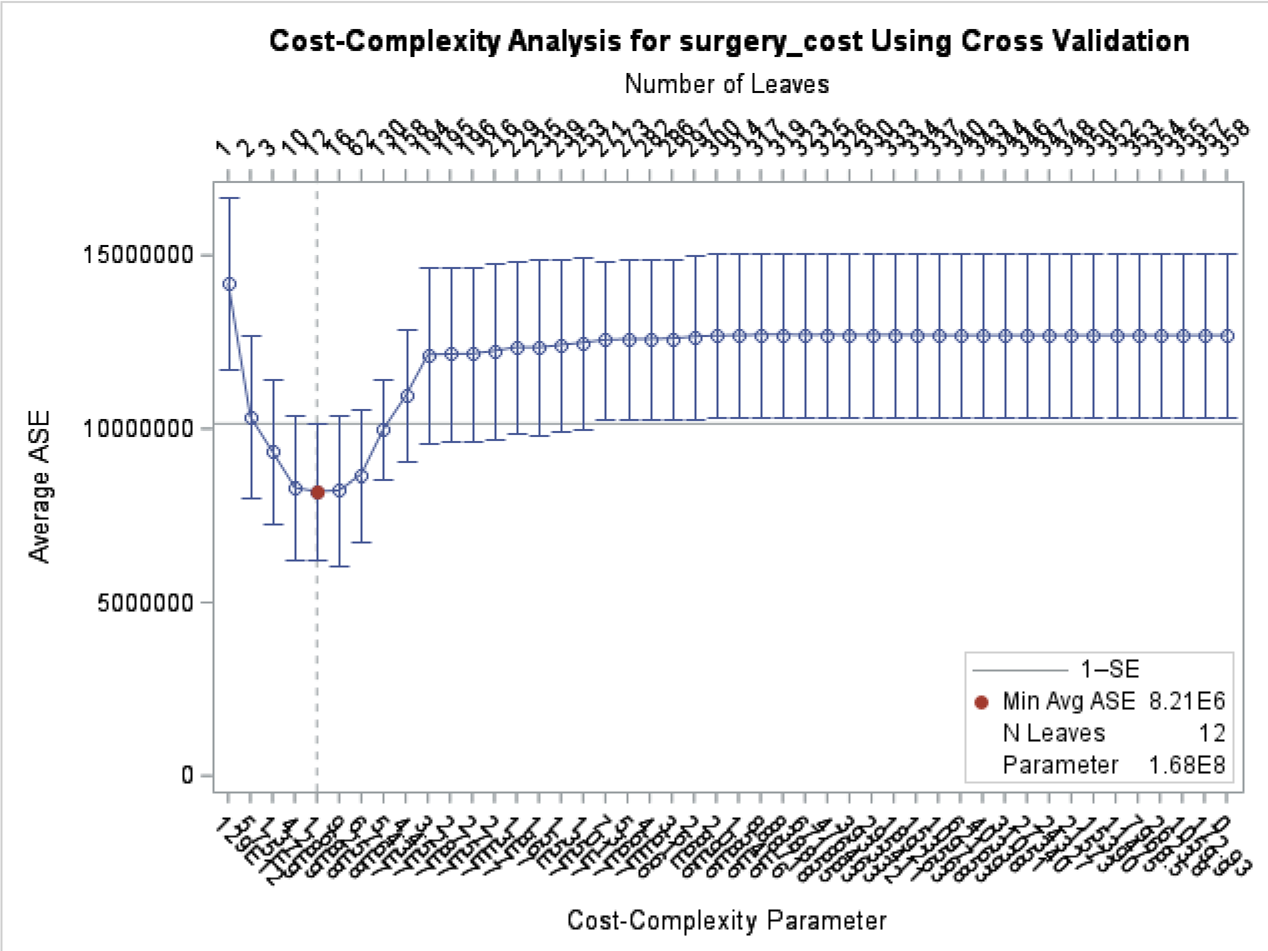
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

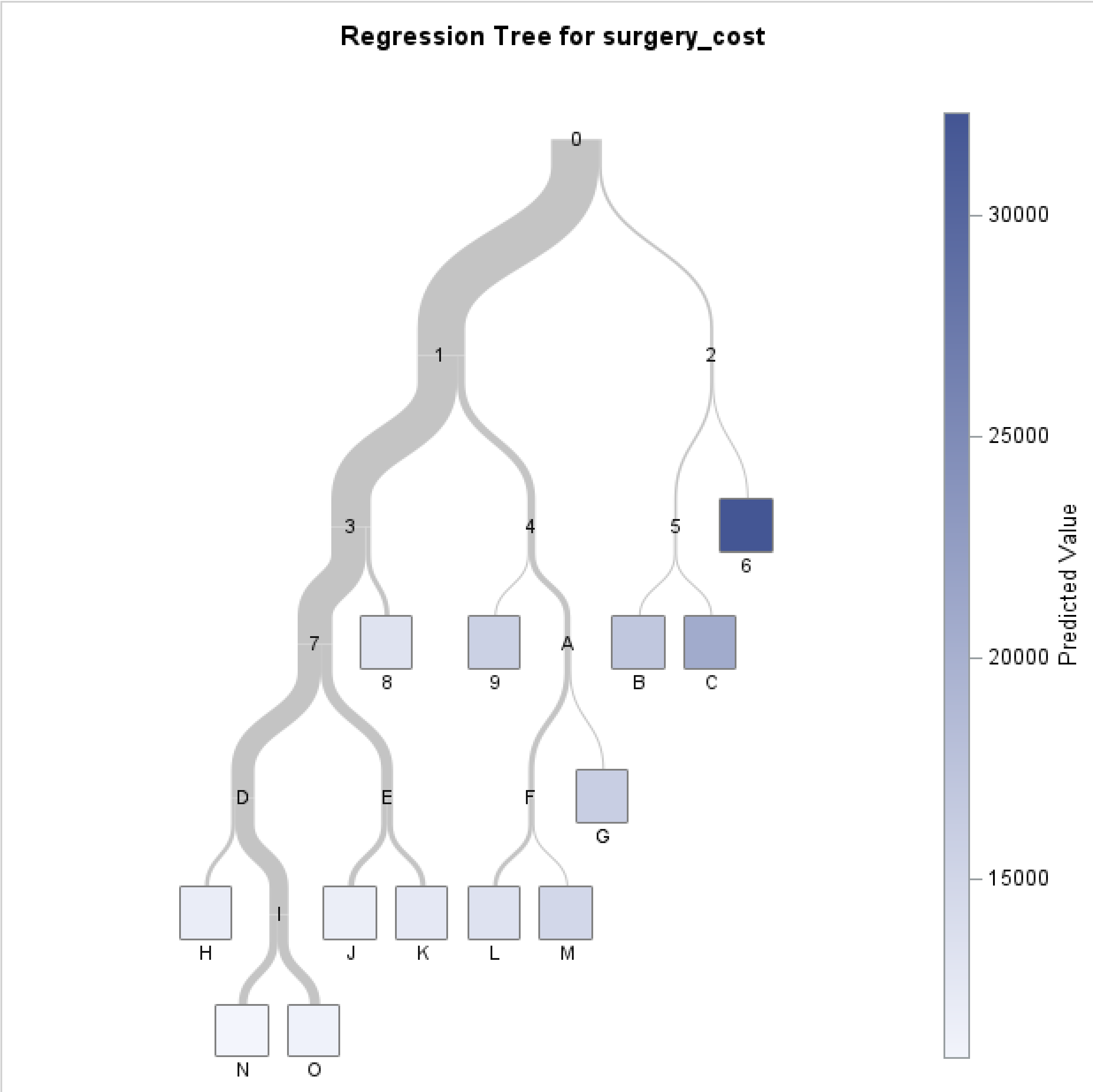
Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	13

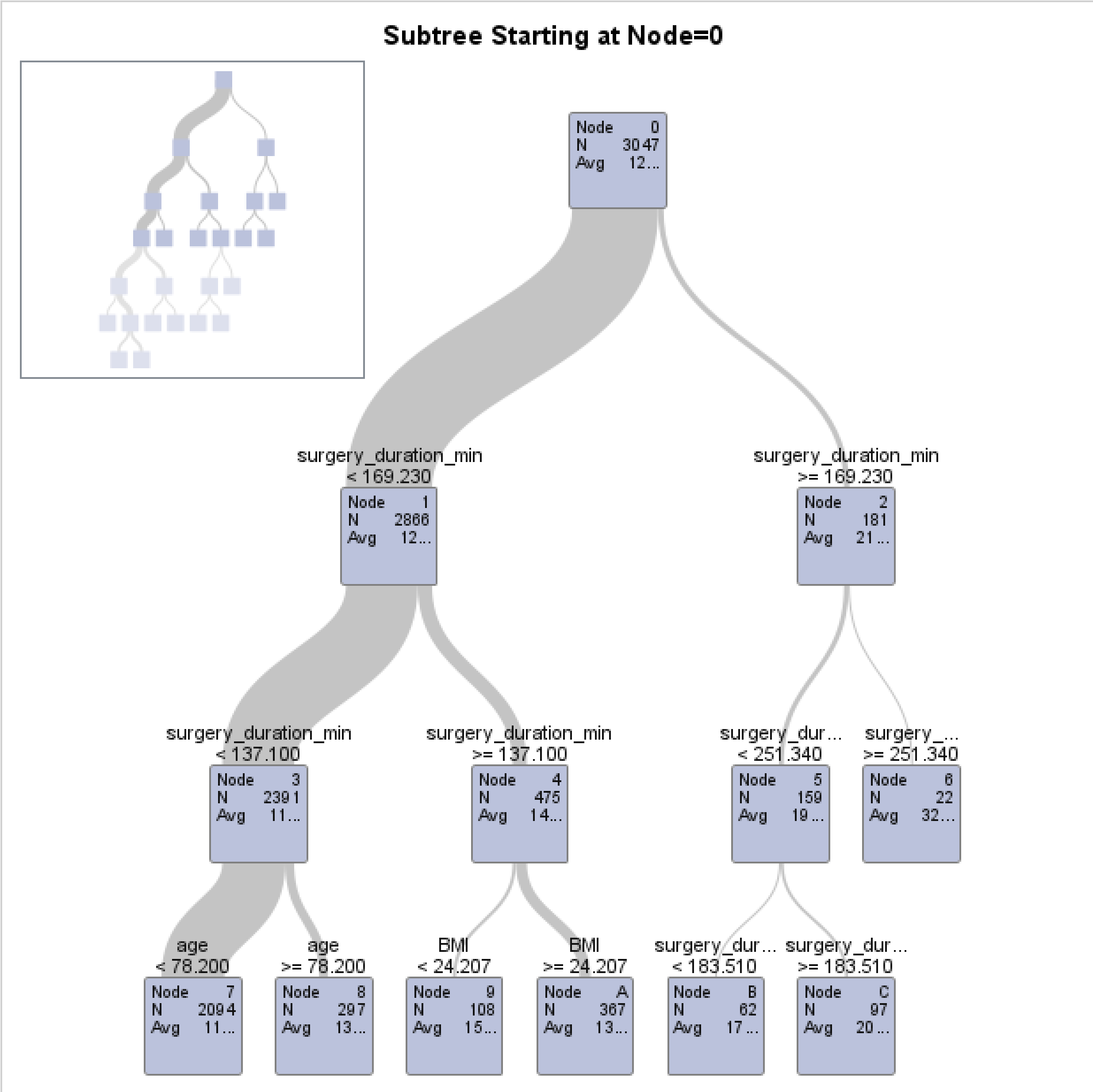
Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure



The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_dur...
< 251.340
Node 5
N 159
Avg 19...
surgery_...
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
13	7242180	2.207E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	140554	6
age	0.2287	32141.5	4
BMI	0.1187	16676.8	1
ASA	0.0862	12112.9	1

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

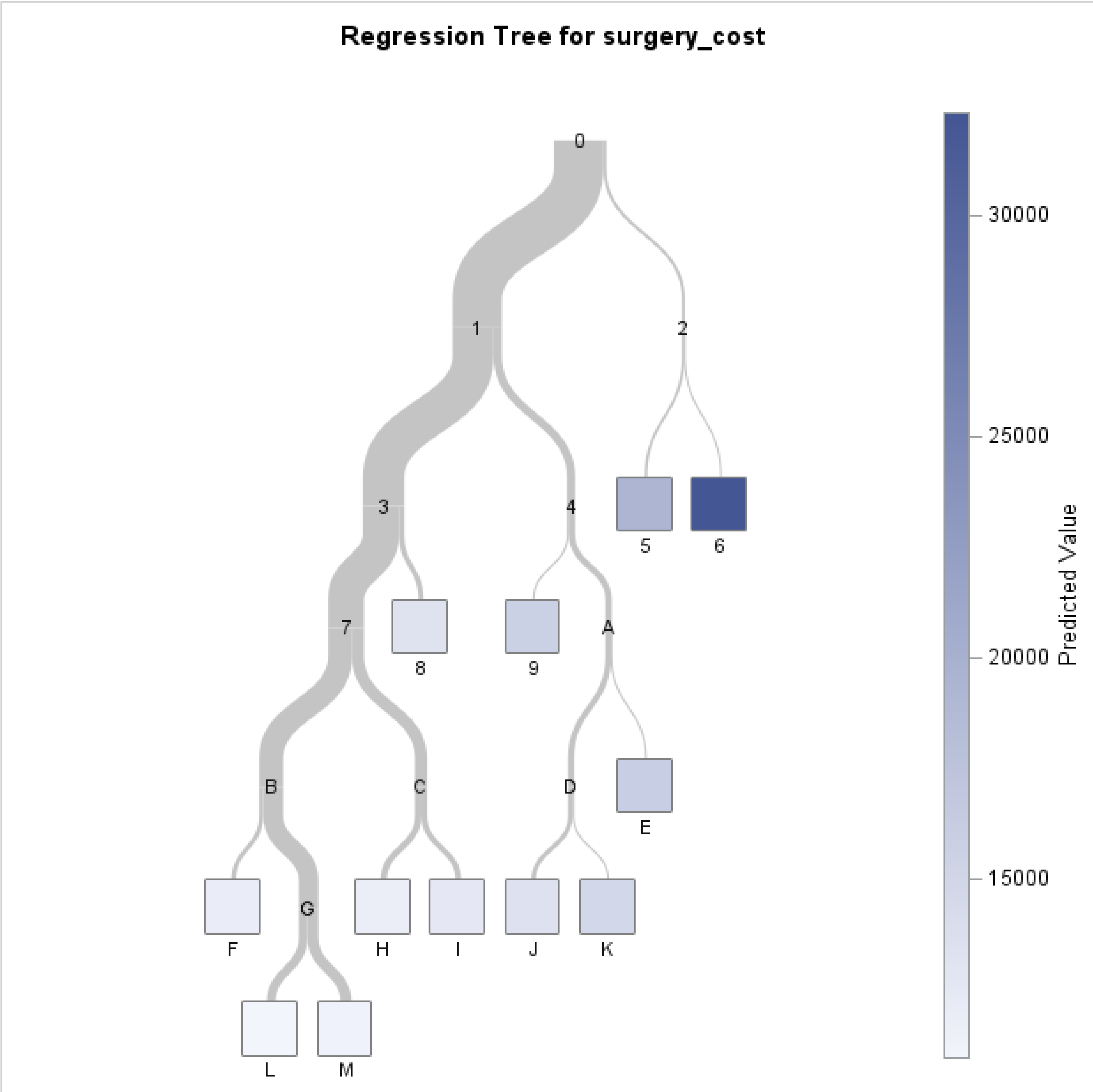
Model Information

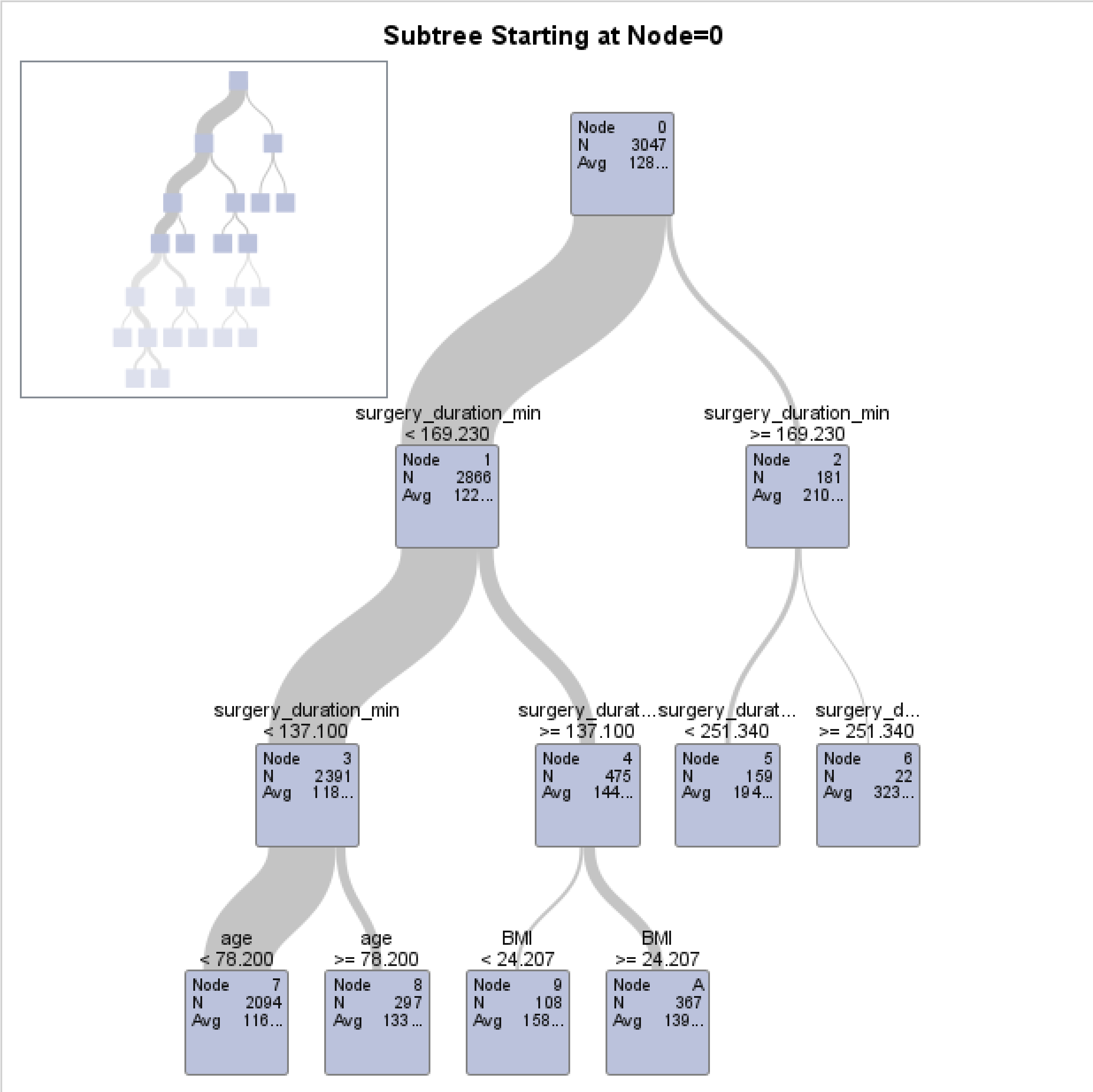
Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	12

Number of Observations Read	3047
Number of Observations Used	3047

The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 3047
Avg 128...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 122...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 210...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 118...
surgery_durat...
>= 137.100
Node 4
N 475
Avg 144...
surgery_durat...
< 251.340
Node 5
N 159
Avg 194...
surgery_d...
>= 251.340
Node 6
N 22
Avg 323...
age
< 78.200
Node 7
N 2094
Avg 116...
age
>= 78.200
Node 8
N 297
Avg 133...
BMI
< 24.207
Node 9
N 108
Avg 158...
BMI
>= 24.207
Node A
N 367
Avg 139...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
12	7415512	2.26E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	138663	5
age	0.2318	32141.5	4
BMI	0.1203	16676.8	1
ASA	0.0874	12112.9	1

accuracy10	accuracy15	accuracy20
0.51117	0.660972	0.78318

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

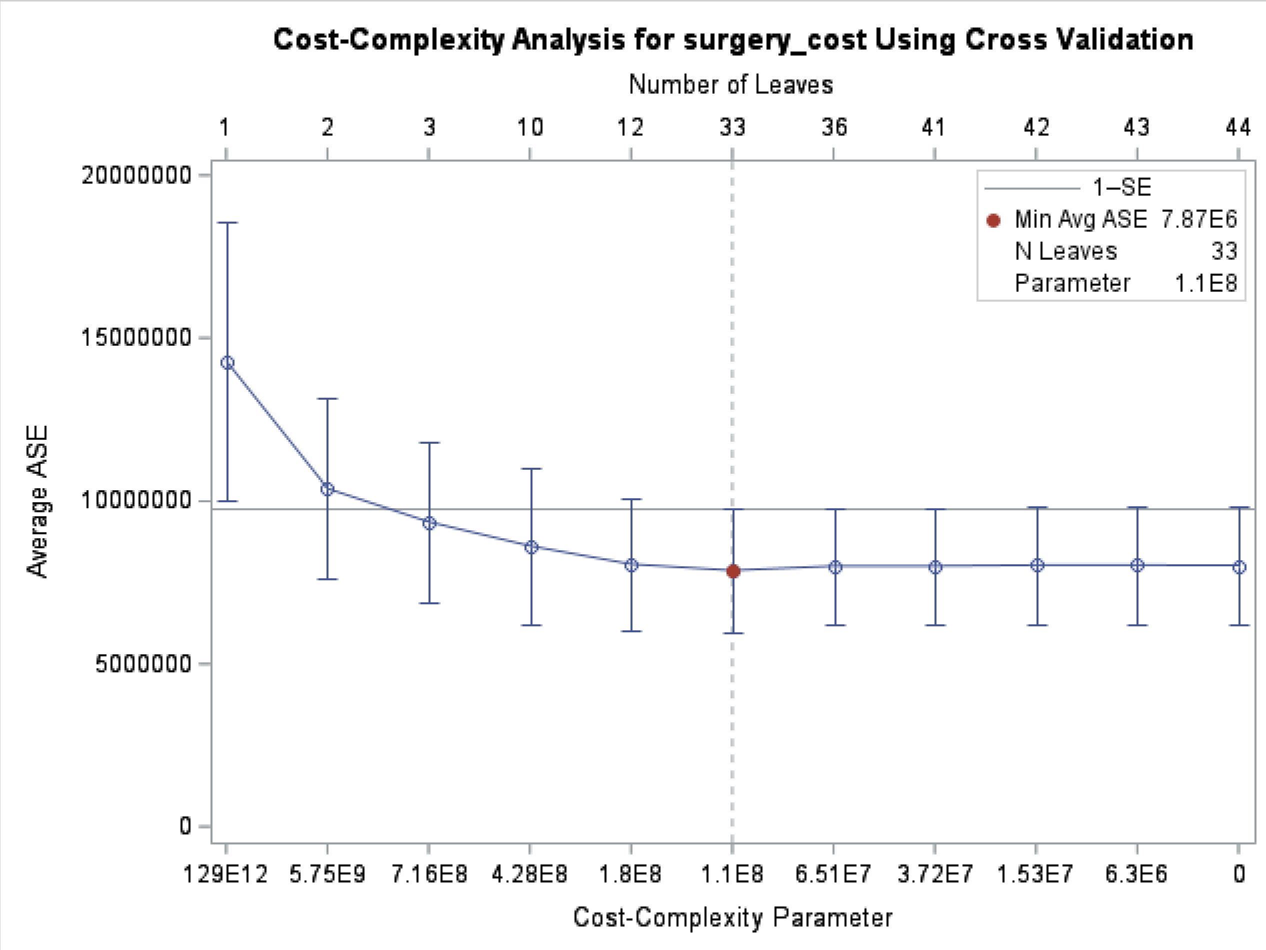
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	32

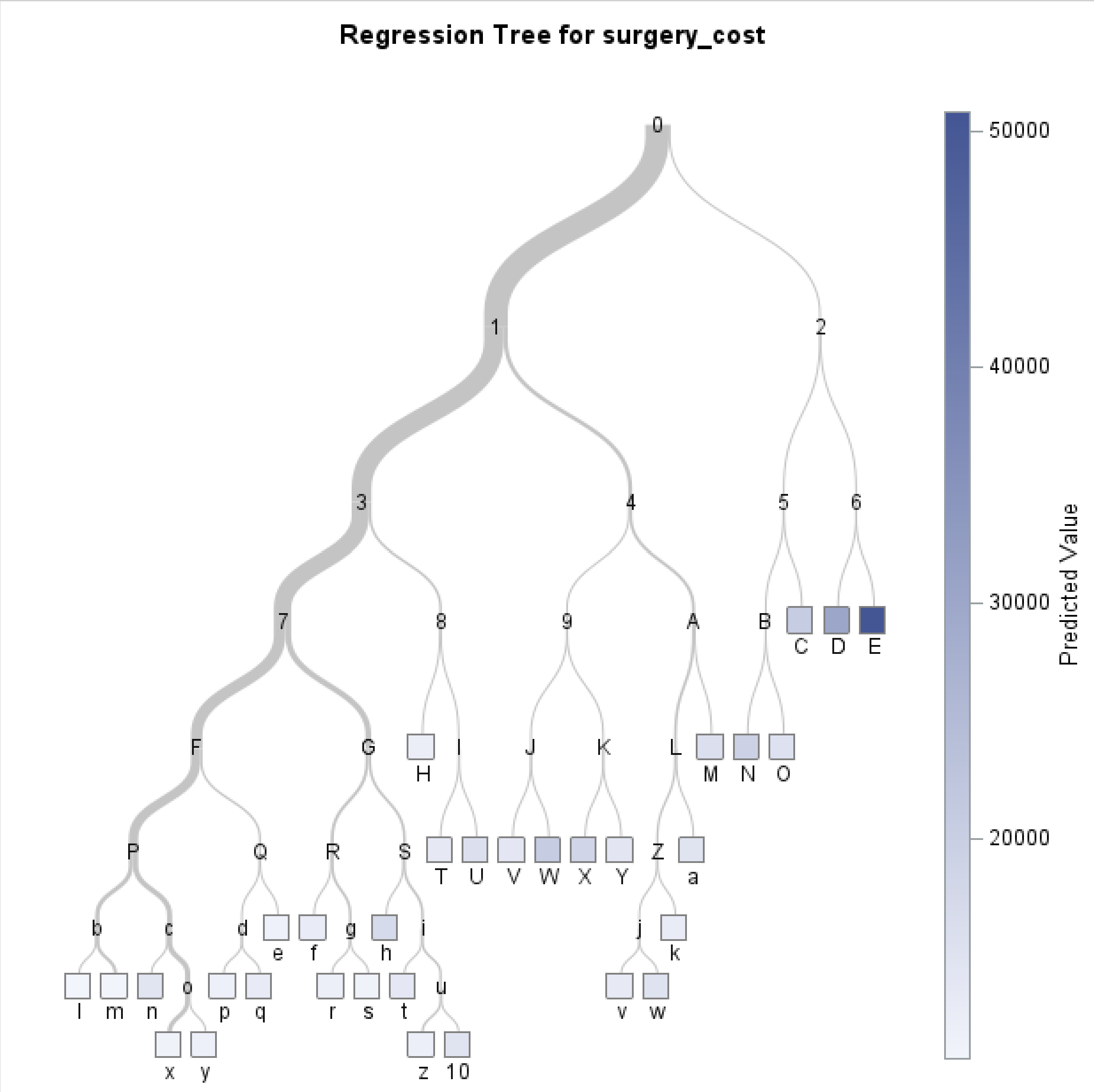
Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure

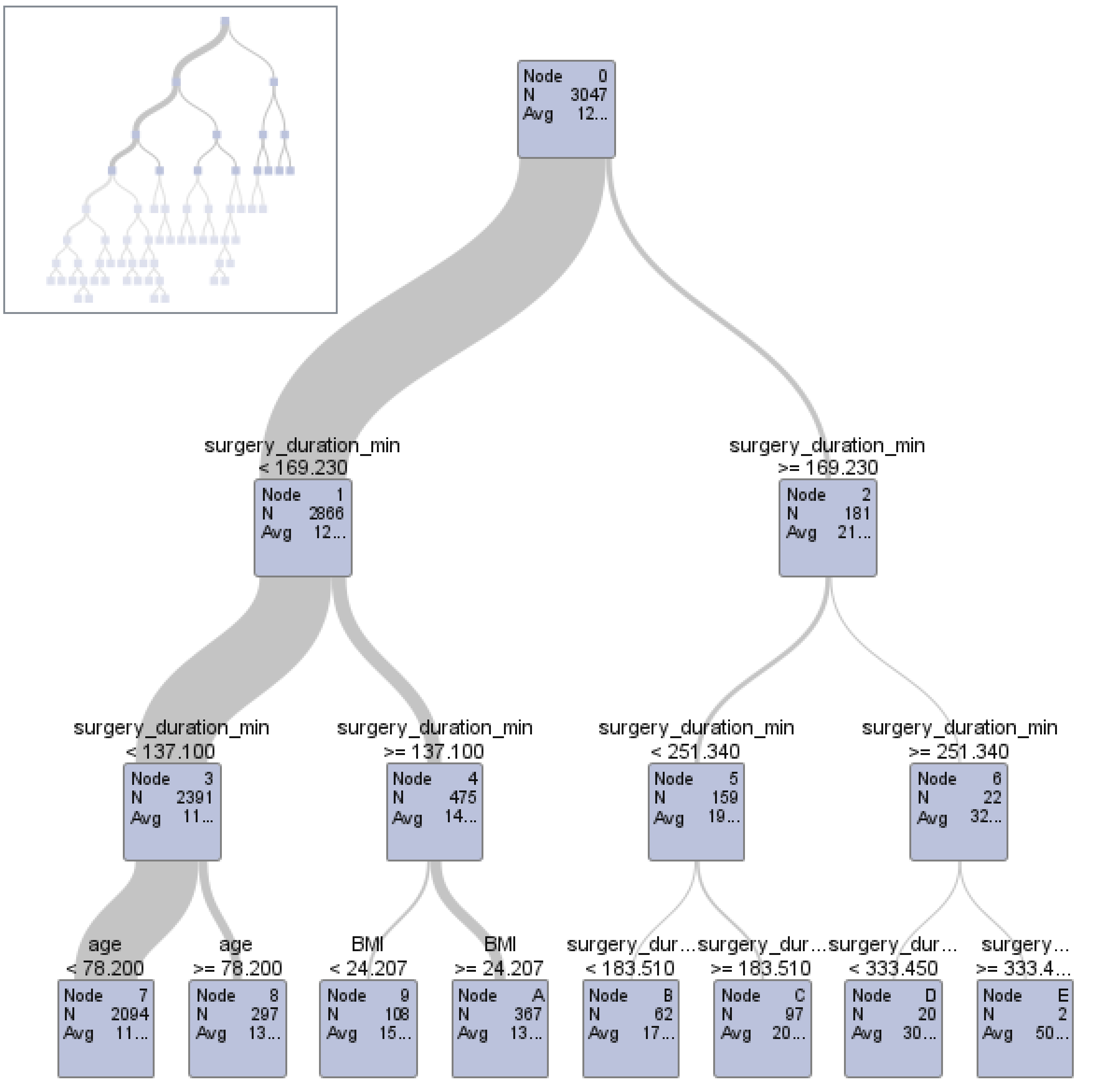


The SAS System

The HPSPLIT Procedure



Subtree Starting at Node=0



The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
32	6469173	1.971E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2588	37412.2	9
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

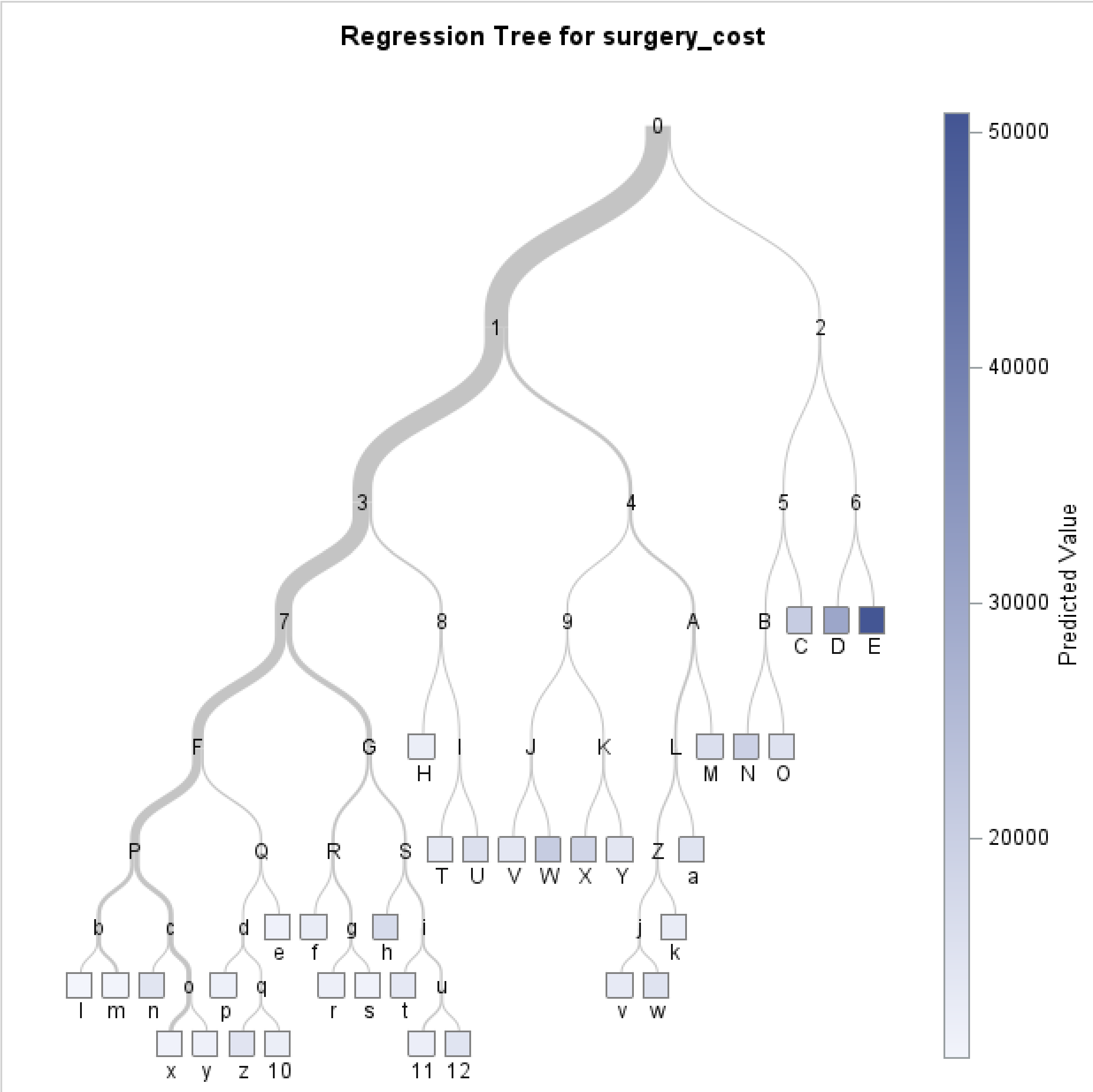
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

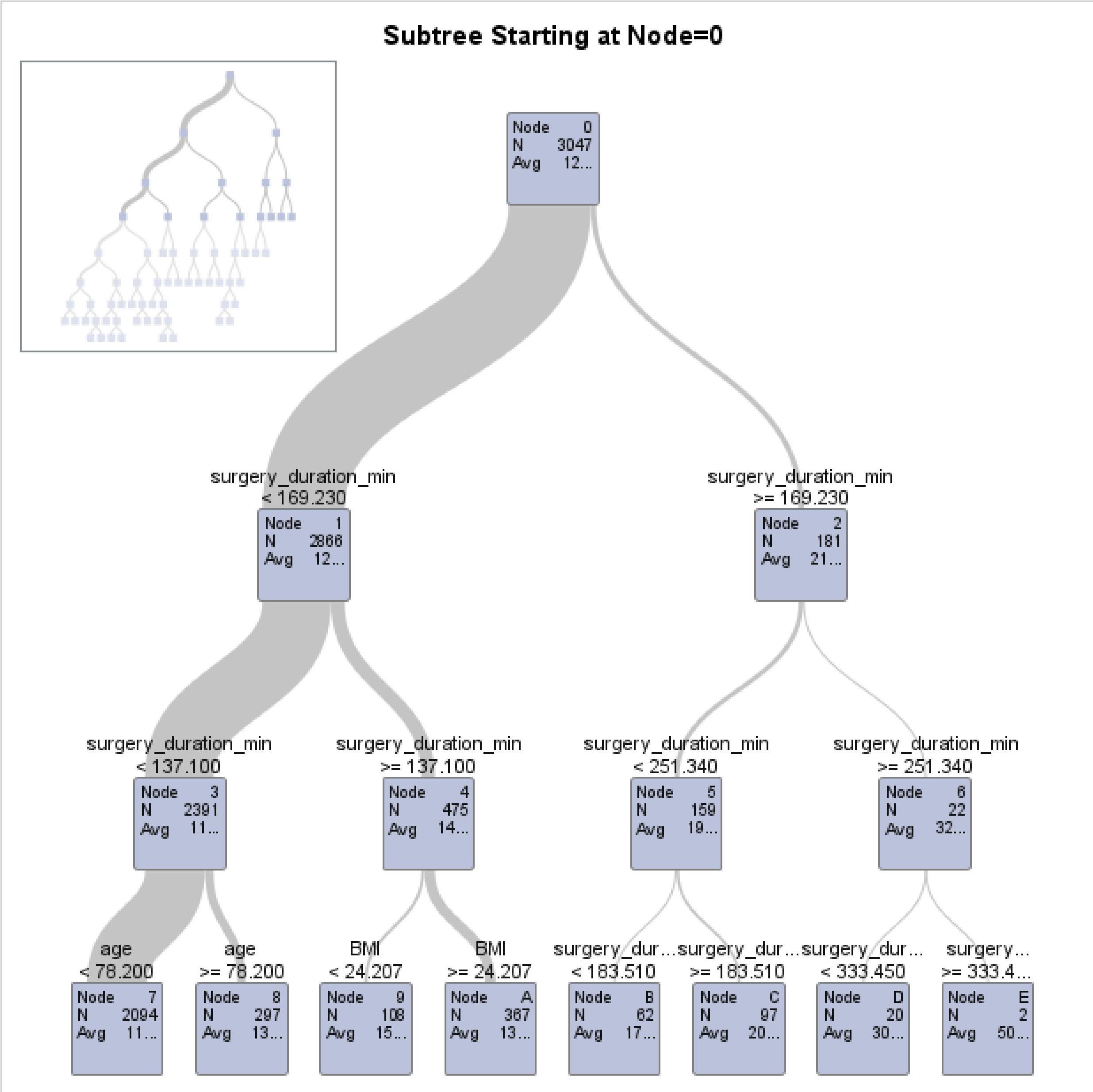
Model Information

Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	33

Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_duration_min
< 251.340
Node 5
N 159
Avg 19...
surgery_duration_min
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...
surgery_dur...
< 333.450
Node D
N 20
Avg 30...
surgery...
>= 333.4...
Node E
N 2
Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
33	6449015	1.965E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2645	38224.2	10
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

accuracy10	accuracy15	accuracy20
0.507227	0.670171	0.805519

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	CARD_DATA
Random Number Seed	122470
Sampling Rate	0.8
Sample Size	1600
Selection Probability	0.8
Sampling Weight	0
Output Data Set	CARD_DATA

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

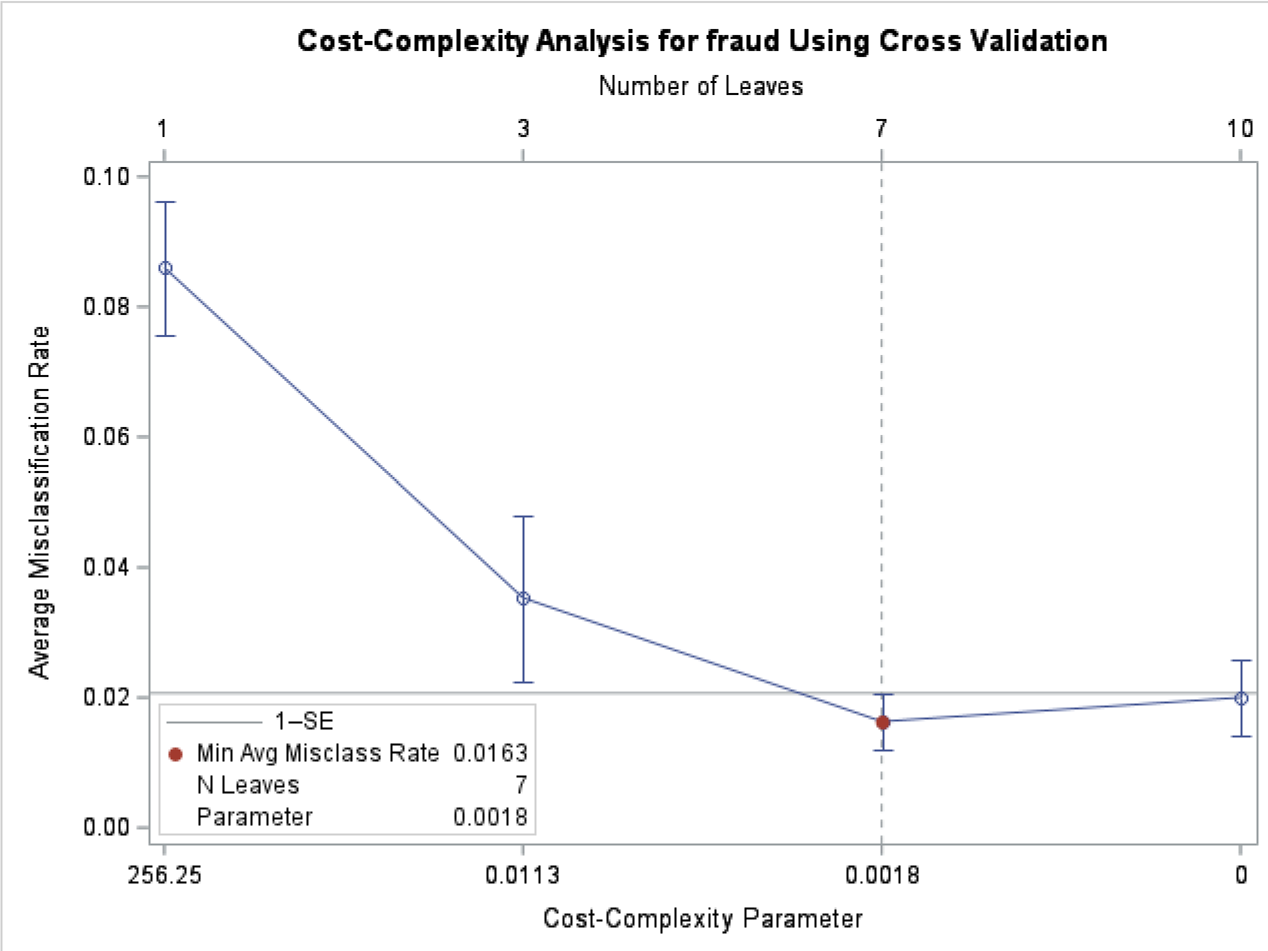
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

Model Information

Split Criterion Used	Gini
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	12
Number of Leaves After Pruning	7
Model Event Level	1

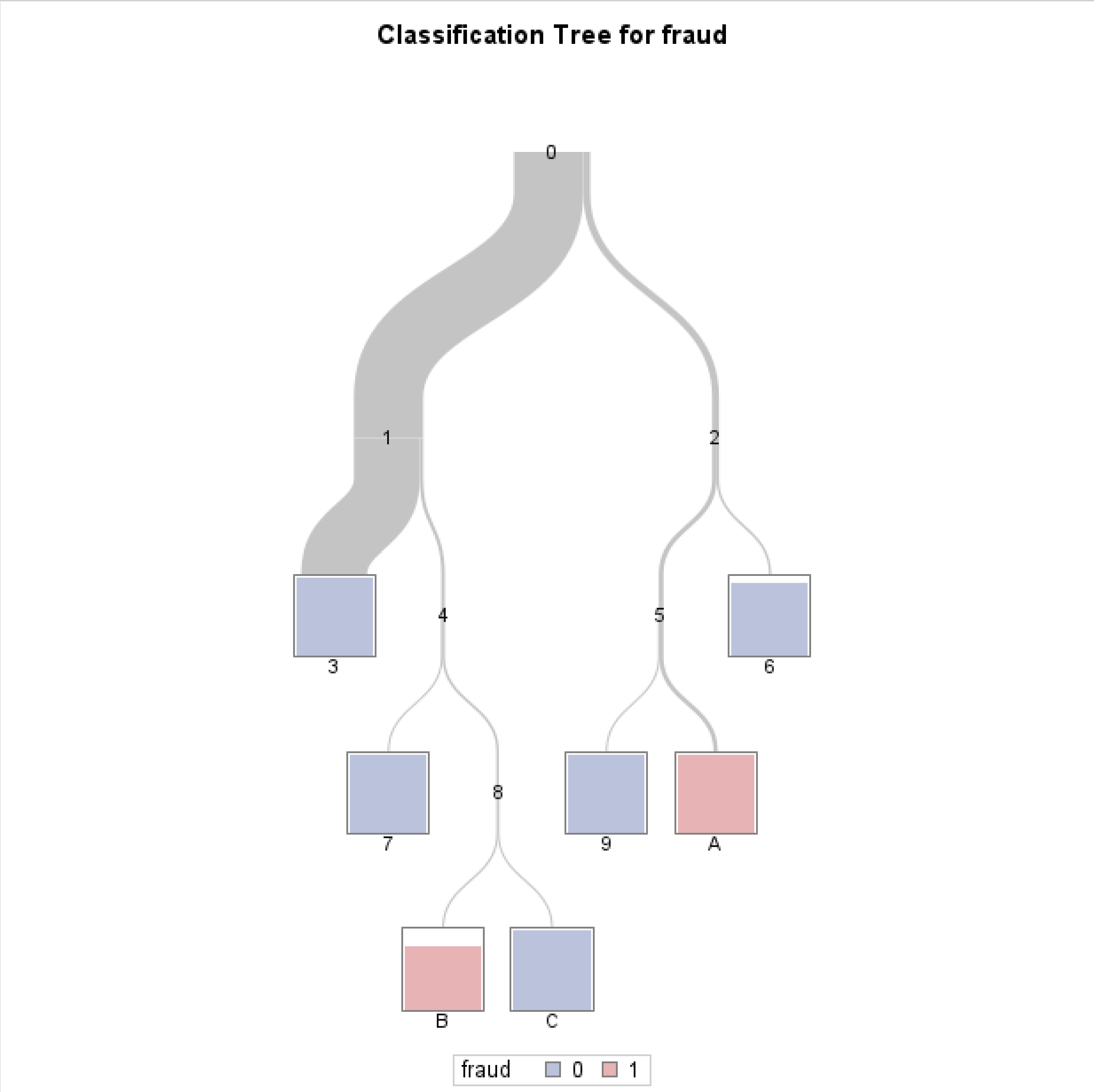
Number of Observations Read	1600
Number of Observations Used	1600

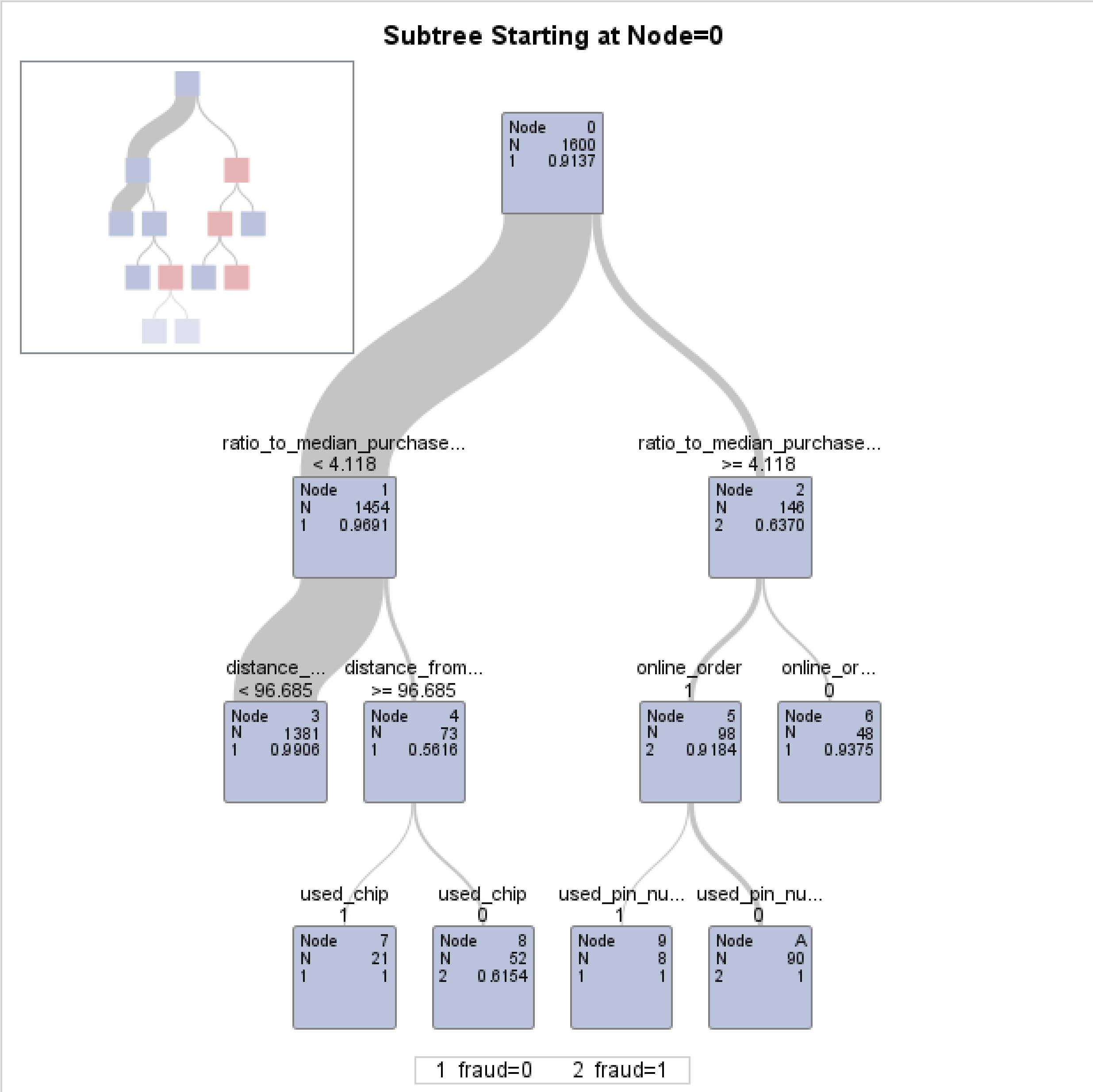
The HPSPLIT Procedure



The SAS System

The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 4.118
Node 1
N 1454
1 0.9691
ratio_to_median_purchase...
>= 4.118
Node 2
N 146
2 0.6370
distance_...
< 96.685
Node 3
N 1381
1 0.9906
distance_from...
>= 96.685
Node 4
N 73
1 0.5616
online_order
1
Node 5
N 98
2 0.9184
online_or...
0
Node 6
N 48
1 0.9375
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 52
2 0.6154
used_pin_nu...
1
Node 9
N 8
1 1
used_pin_nu...
0
Node A
N 90
2 1
1 fraud=0
2 fraud=1

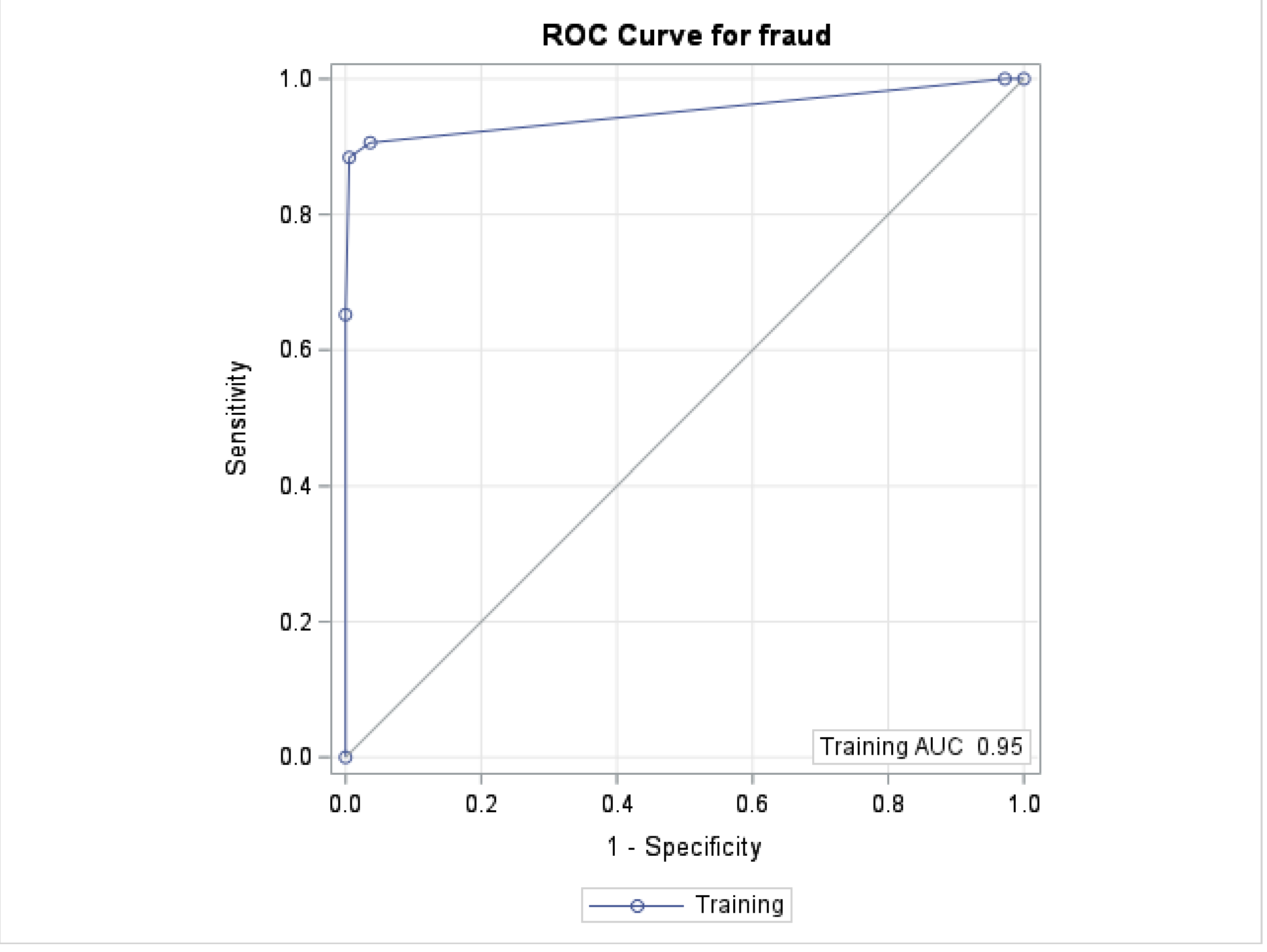
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1454	8	0.0055
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
7	0.0138	0.0150	0.8841	0.9945	0.0945	0.0276	44.1802	0.9514



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.3883	3.8333	1
used_chip	0.3410	3.3660	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	**trueclassrate**
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

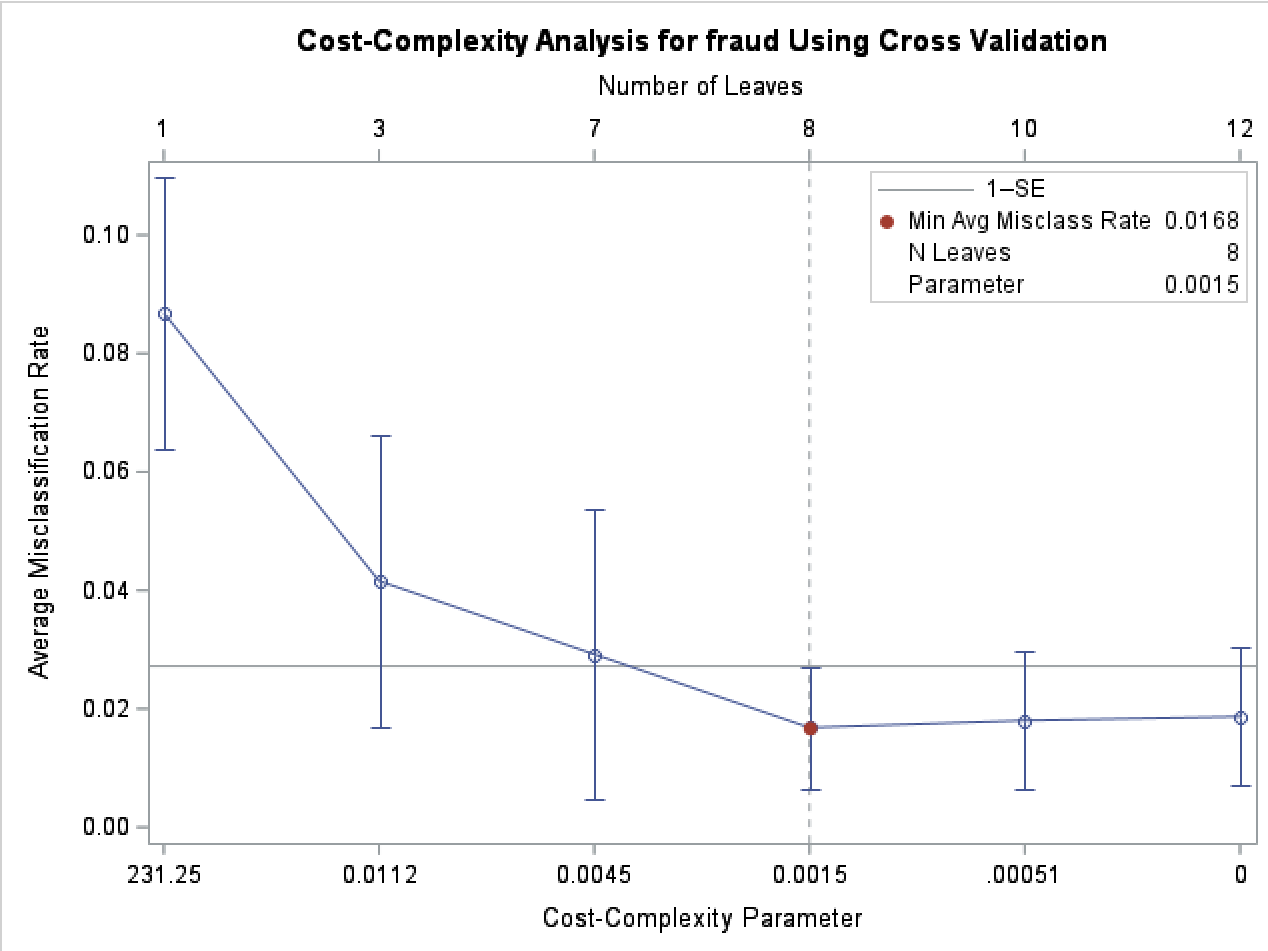
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED2	V9	Output	On Client

Model Information

Split Criterion Used	Entropy
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	13
Number of Leaves After Pruning	8
Model Event Level	1

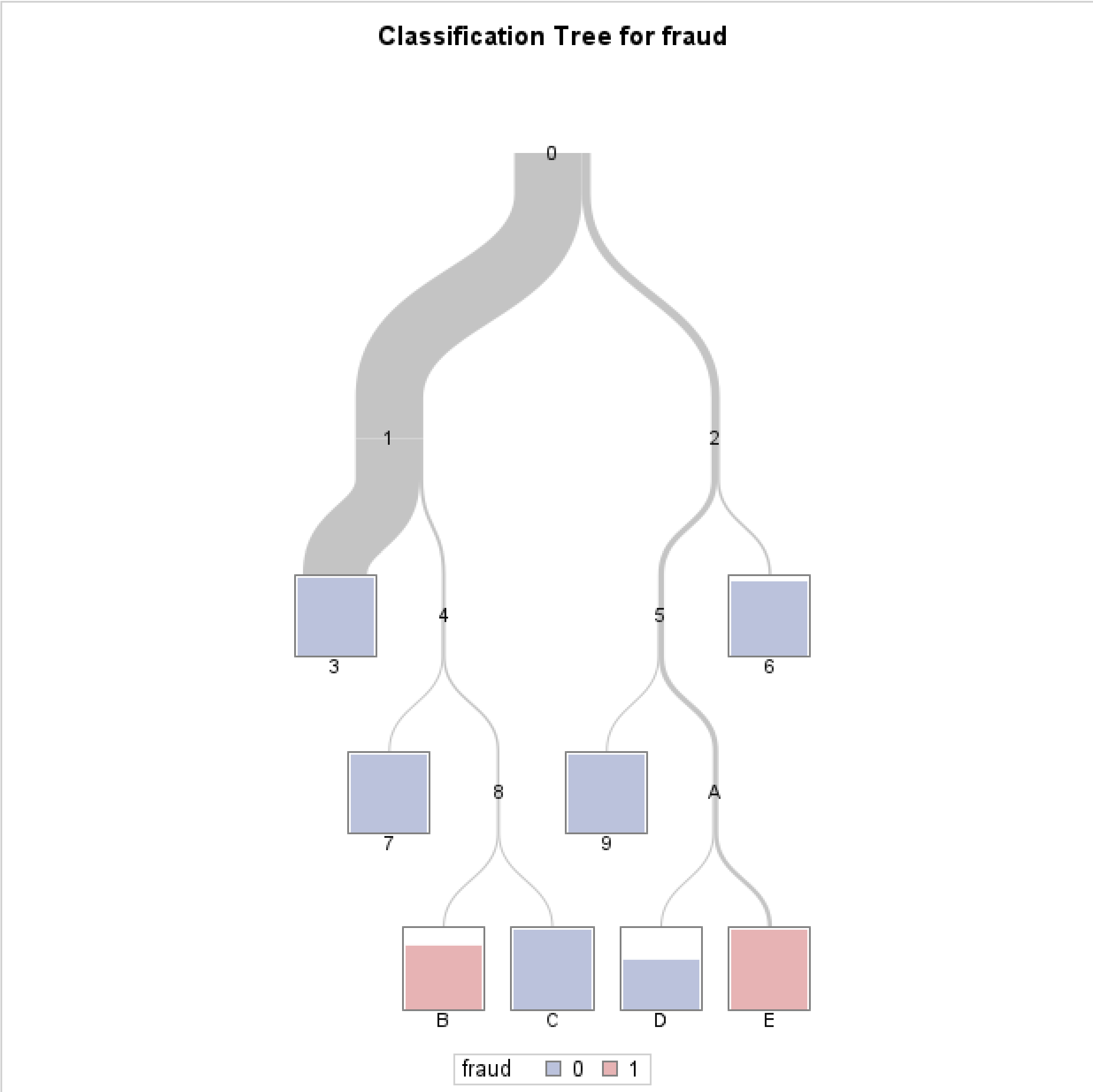
Number of Observations Read	1600
Number of Observations Used	1600

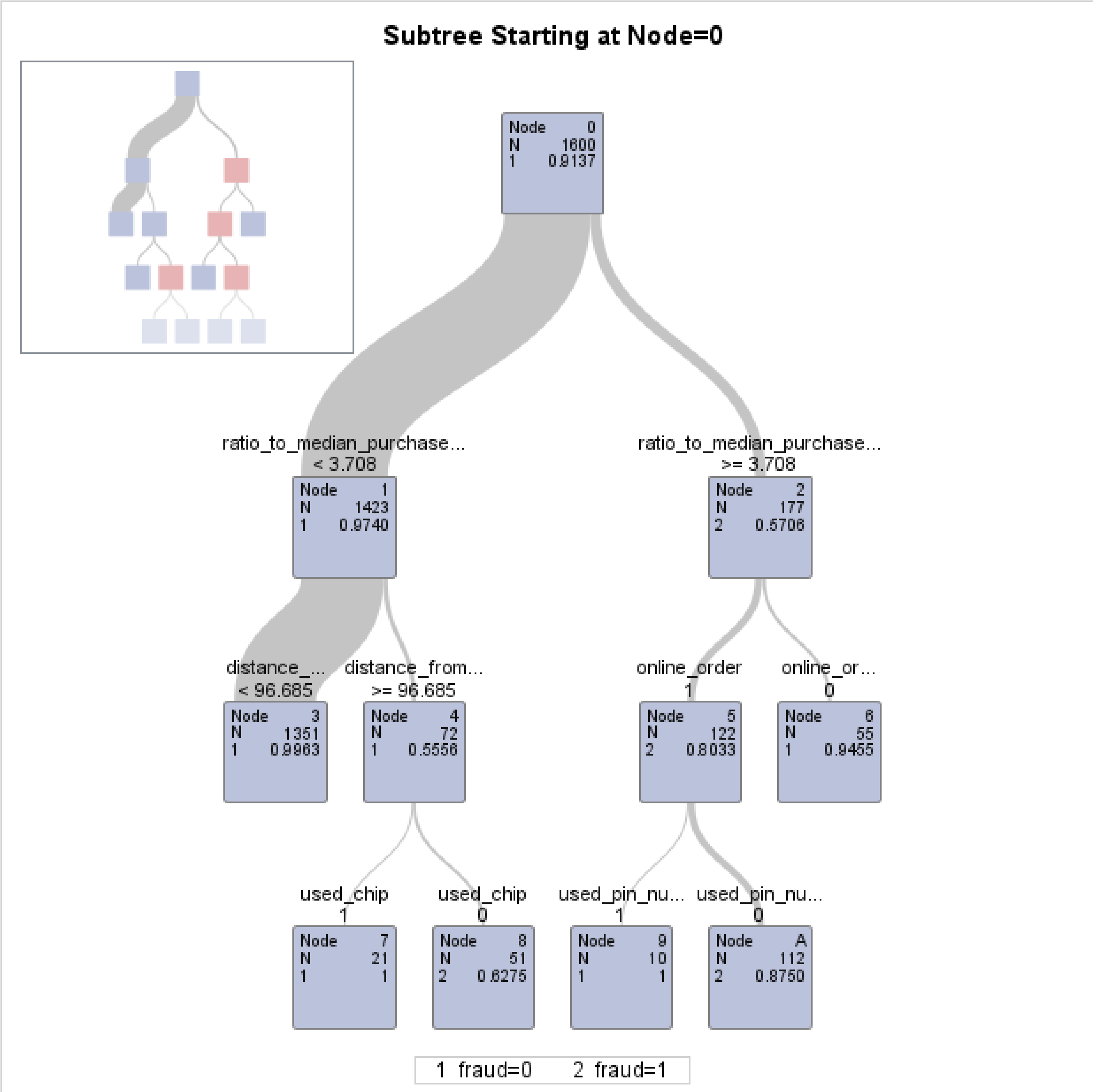
The HPSPLIT Procedure



The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 3.708
Node 1
N 1423
1 0.9740
ratio_to_median_purchase...
>= 3.708
Node 2
N 177
2 0.5706
distance_...
< 96.685
Node 3
N 1351
1 0.9963
distance_from...
>= 96.685
Node 4
N 72
1 0.5556
online_order
1
Node 5
N 122
2 0.8033
online_or...
0
Node 6
N 55
1 0.9455
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 51
2 0.6275
used_pin_nu...
1
Node 9
N 10
1 1
used_pin_nu...
0
Node A
N 112
2 0.8750
1 fraud=0
2 fraud=1

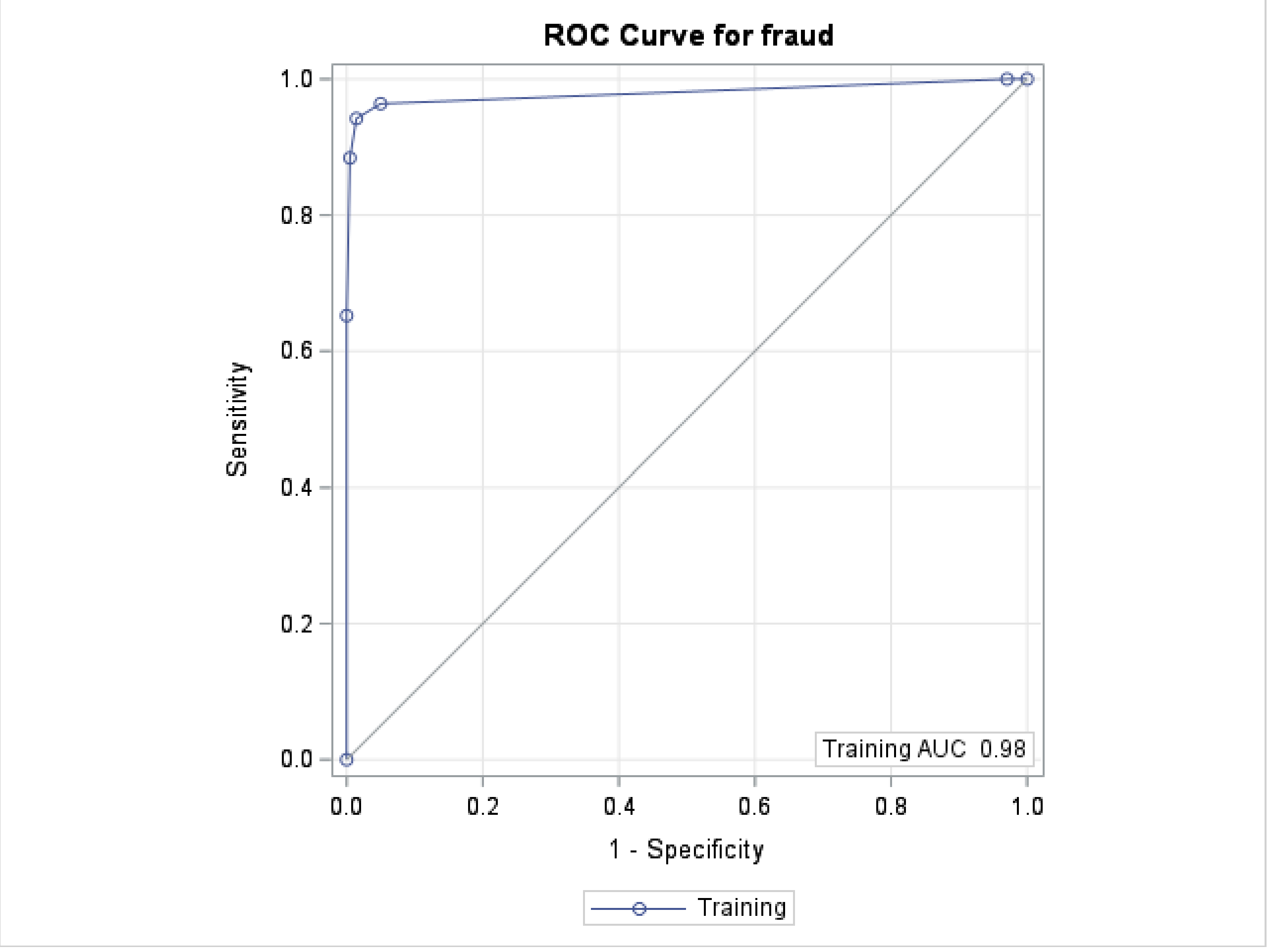
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1455	7	0.0048
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
8	0.0117	0.0144	0.8841	0.9952	0.0698	0.0233	37.3047	0.9797



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	10.3780	2
online_order	0.7137	7.4068	2
distance_from_home	0.4966	5.1534	1
used_pin_number	0.3613	3.7493	1
used_chip	0.3298	3.4223	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	**trueclassrate**
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

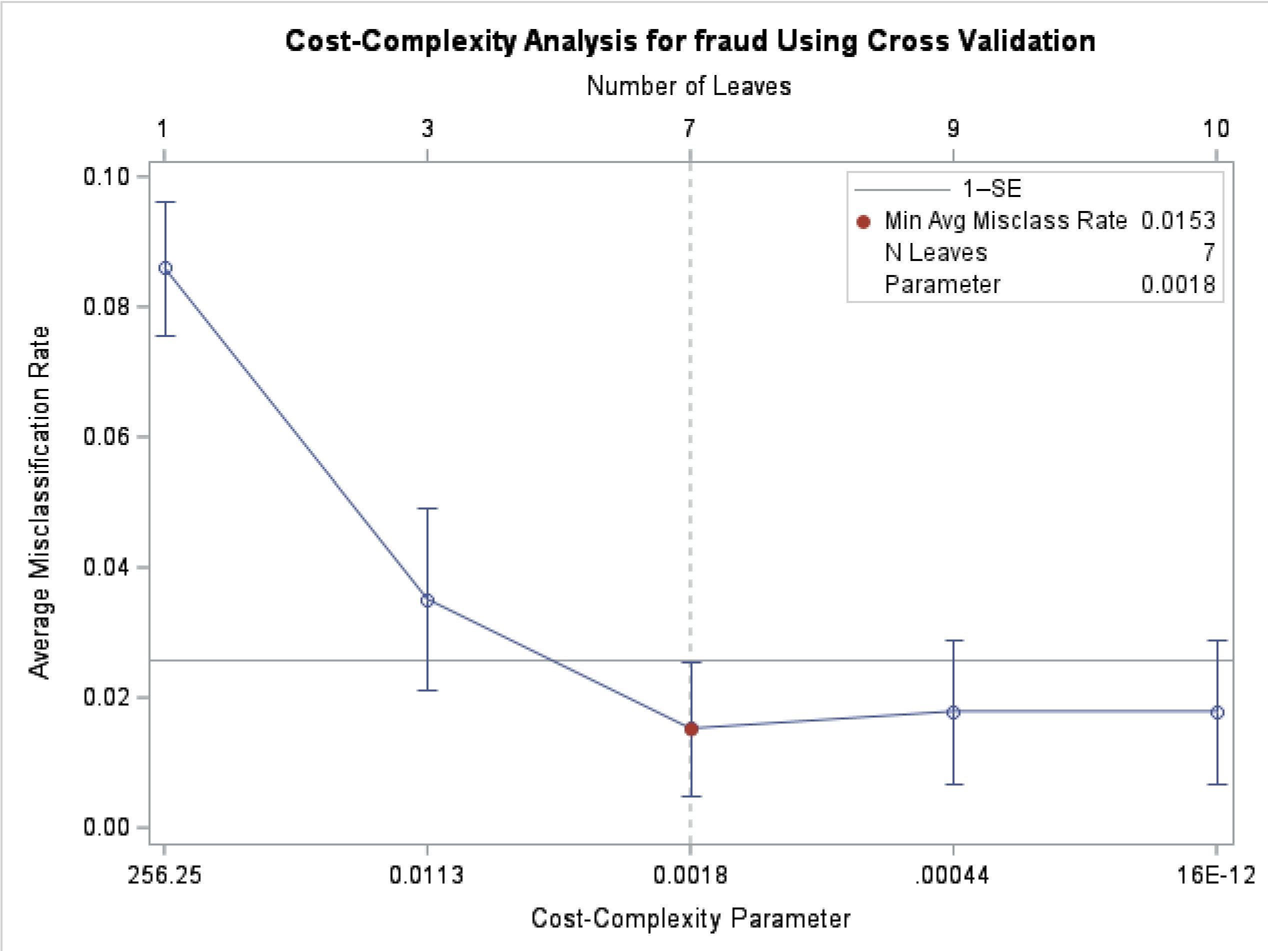
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED3	V9	Output	On Client

Model Information

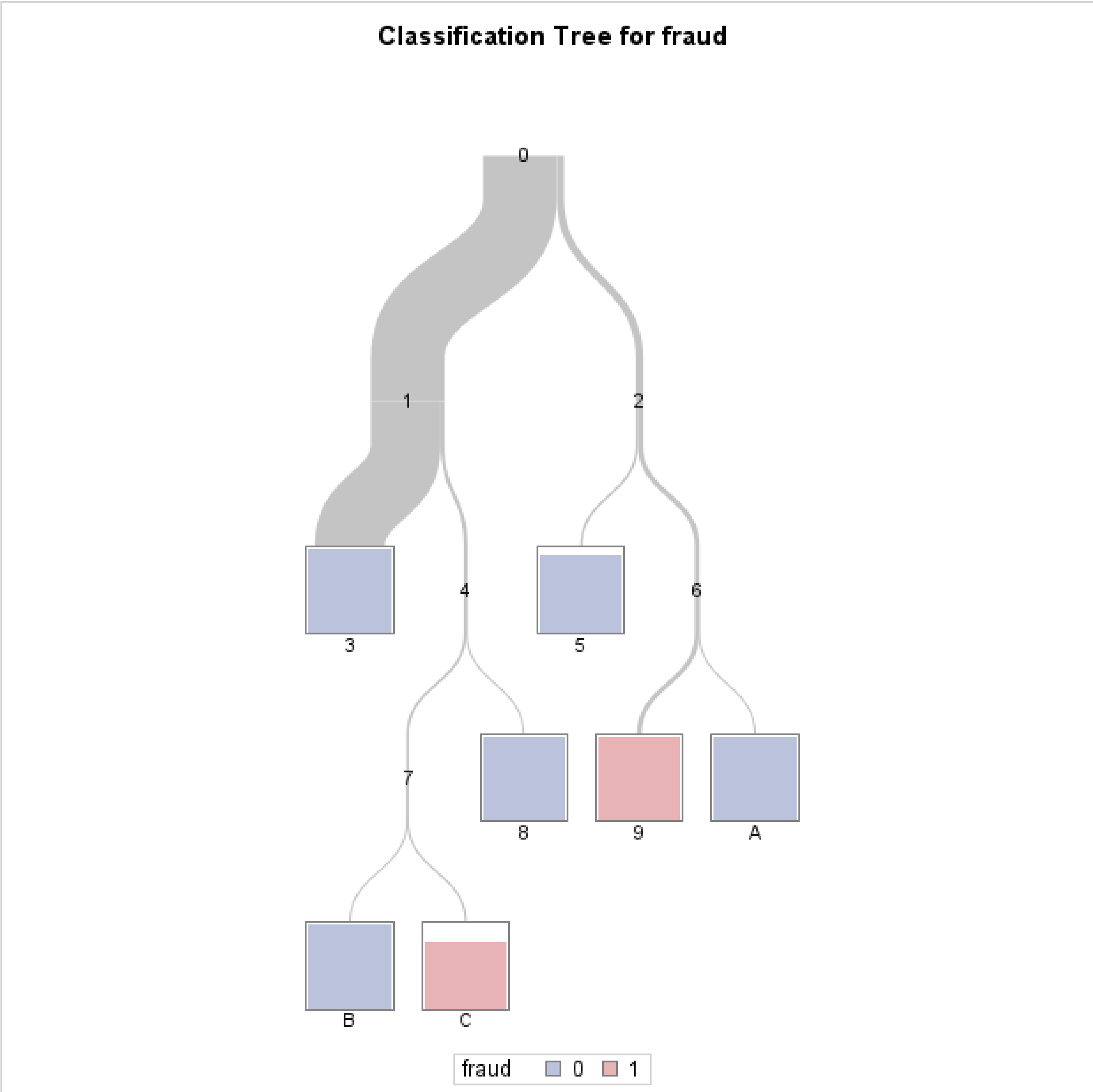
Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	12
Number of Leaves After Pruning	7
Model Event Level	1

Number of Observations Read	1600
Number of Observations Used	1600

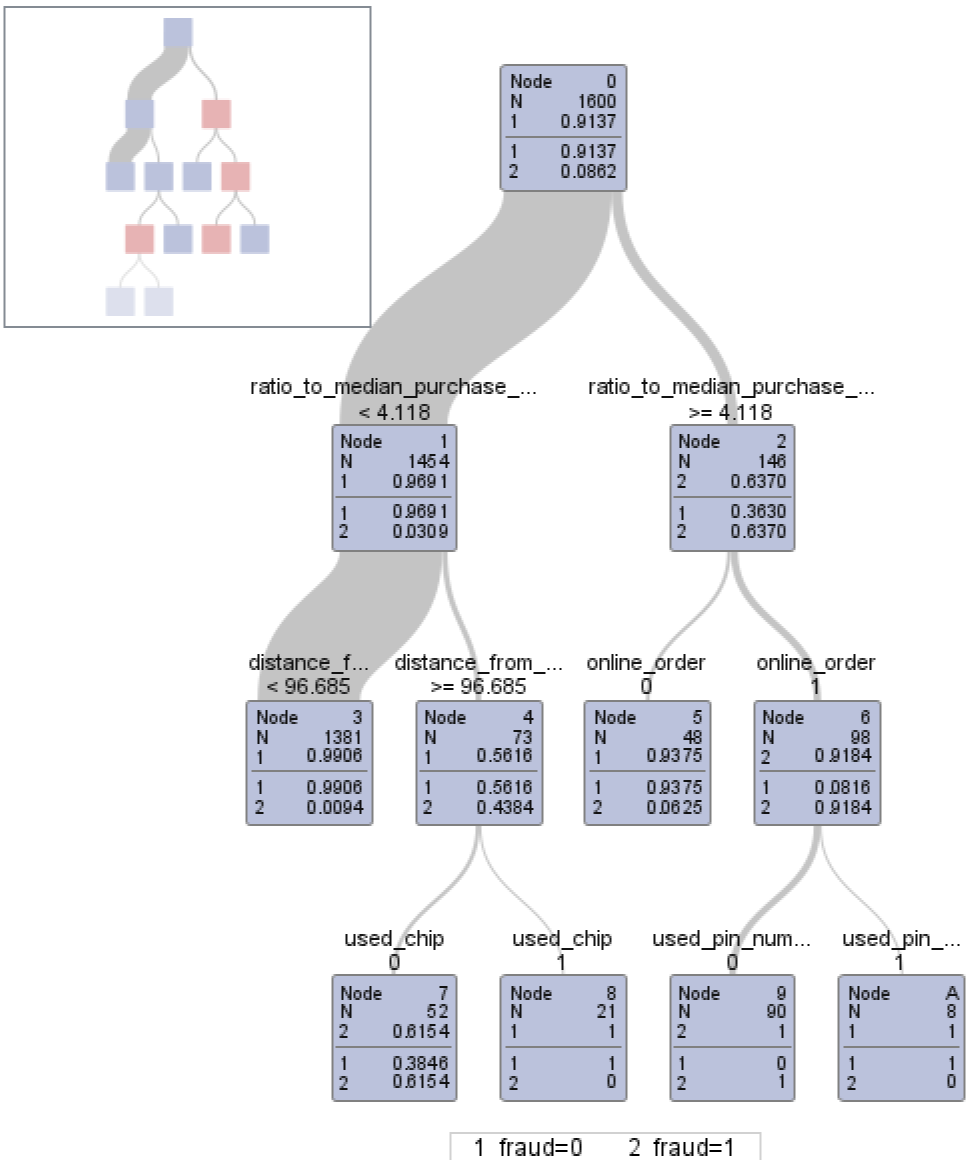
The HPSPLIT Procedure



The HPSPLIT Procedure



Subtree Starting at Node=0



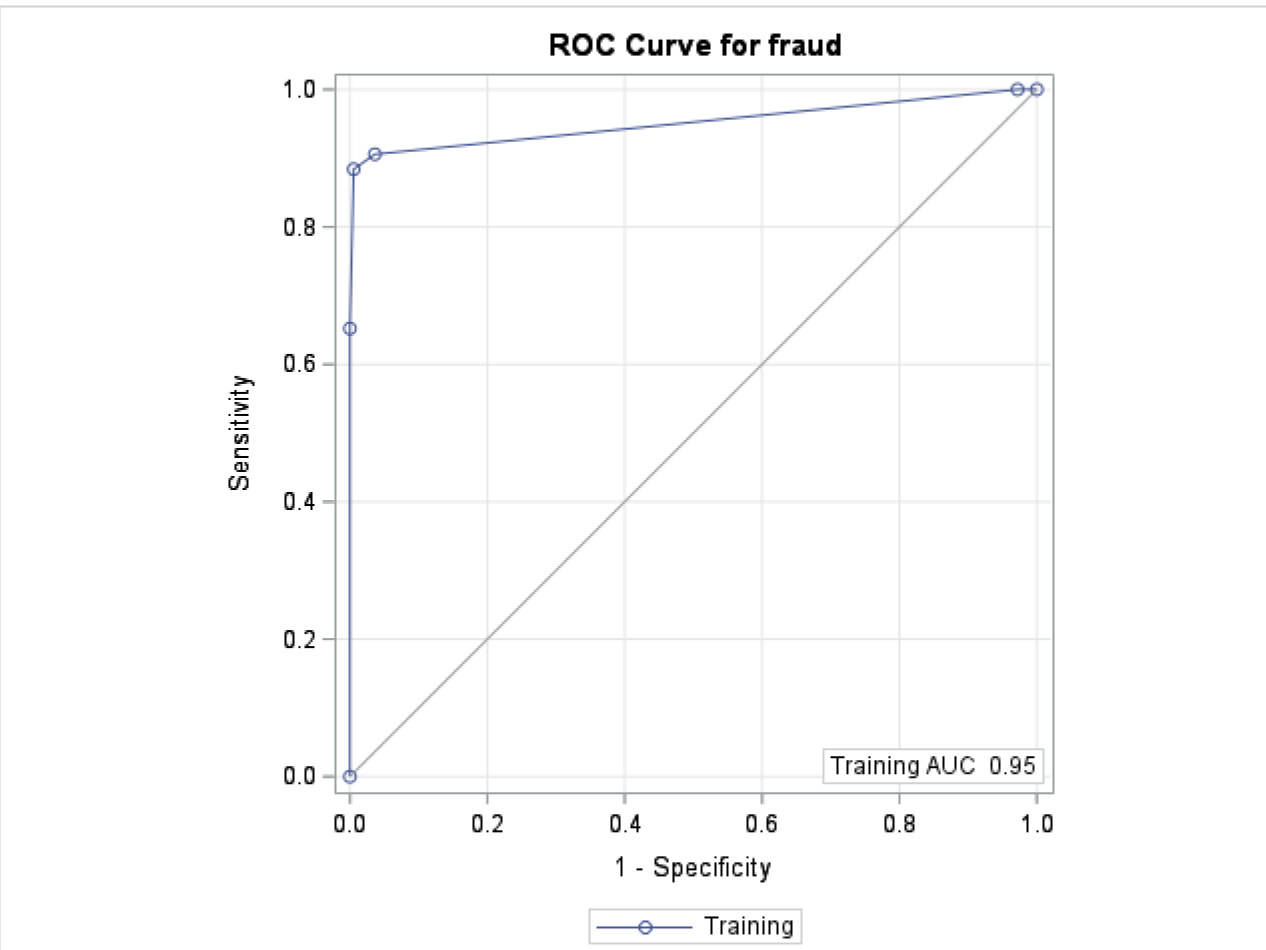
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1454	8	0.0055
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
7	0.0138	0.0150	0.8841	0.9945	0.0945	0.0276	44.1802	0.9514



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.3883	3.8333	1
used_chip	0.3410	3.3660	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	trueclassrate
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

R Code

```
library(readr)

library(rpart)
library(rpart.plot)
library(dplyr)
library(partykit)
library(CHAID)

card_data =
read.csv("C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/card_transdata.csv
",
header=T, sep=",")

# Splitting data into 80% training and 20% testing sets.

set.seed(122470)
sample = sample(c(T,F), nrow(card_data),
replace=T, prob=c(0.8, 0.2))
train = card_data[sample,]
```

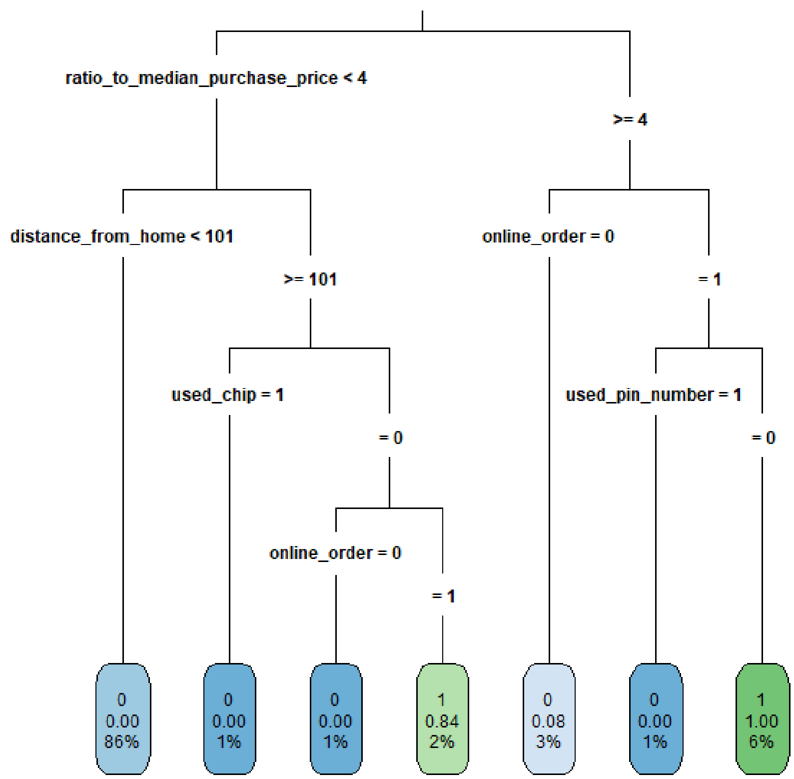
```r
test = card_data[!sample,]

# Fitting pruned binary tree with Gini Splitting Criterion.

tree_gini = rpart(fraud~distance_from_home+distance_from_last_transaction
+ratio_to_median_purchase_price+repeat_retailer+used_chip+used_pin_number
+online_order, data=train, method="class", parms=list(split="Gini"),
maxdepth=7)

rpart.plot(tree_gini, type=3)

# Computing prediction accuracy for testing data for Gini Tree.

pred_values = predict(tree_gini, test)
test = cbind(test, pred_values)

tp = matrix(NA, nrow=nrow(test), ncol=99)
tn = matrix(NA, nrow=nrow(test), ncol=99)

for (i in 1:99) {
    tp[,i] = ifelse(test$fraud=="1" & test$"1">0.01*i,1,0)
    tn[,i] = ifelse(test$fraud=="0" & test$"1"<=0.01*i,1,0)
}

trueclassrate = matrix(NA, nrow=99, ncol=2)
for (i in 1:99) {
    trueclassrate[i,1] = 0.01*i
    trueclassrate[i,2] = sum(tp[,i]+tn[,i])/nrow(test)
}

print(trueclassrate[which(trueclassrate[,2]==max(trueclassrate[,2])),])

# Fitting pruned binary tree with entropy splitting

tree_entropy = rpart(fraud~distance_from_home+distance_from_last_transaction
+ratio_to_median_purchase_price+repeat_retailer+used_chip+used_pin_number
+online_order, data=train, method="class", parms=list(split="Gini"),
maxdepth=7)

rpart.plot(tree_entropy, type=3)

# Computing prediction accuracy with testing data for Entropy Tree.

pred_values2 = predict(tree_entropy, test)
test2 = cbind(test, pred_values2)
```

```r
tp2 = matrix(NA, nrow=nrow(test), ncol=99)
tn2 = matrix(NA, nrow=nrow(test), ncol=99)

for (i in 1:99) {
    tp2[,i] = ifelse(test$fraud=="1" & test$"1">0.01*i,1,0)
    tn2[,i] = ifelse(test$fraud=="0" & test$"1"<=0.01*i,1,0)
}

trueclassrate2 = matrix(NA, nrow=99, ncol=2)
for (i in 1:99) {
    trueclassrate2[i,1] = 0.01*i
    trueclassrate2[i,2] = sum(tp2[,i]+tn2[,i])/nrow(test)
}

print(trueclassrate2[which(trueclassrate2[,2]==max(trueclassrate2[,2])),])

card_data = mutate(card_data, distance_from_home_cat=ntile(distance_from_home,
10),
distance_from_last_transaction_cat=ntile(distance_from_last_transaction, 10),
ratio_to_median_purchase_price_cat=ntile(ratio_to_median_purchase_price, 10))

# Splitting data into 80% training and 20% testing sets.

set.seed(590520)
sample = sample(c(T,F), nrow(card_data), replace=T, prob=c(0.8, 0.2))
train = card_data[sample,]
test = card_data[!sample,]

# Fitting CHAID tree.

tree_CHAID = chaid(as.factor(fraud)~as.factor(distance_from_home_cat)
+as.factor(distance_from_last_transaction_cat)+as.factor(ratio_to_median_purchase
_price_cat)
+as.factor(repeat_retailer)+as.factor(used_chip)+as.factor(used_pin_number)
+as.factor(online_order), data=train, control=chaid_control(maxheight=3))

plot(tree_CHAID, type="simple")

# Computing prediction accuracy for testing data for CHAID tree.

pred_values3 = predict(tree_CHAID, newdata=test)
test3 = cbind(test, pred_values3)

tp3 = matrix(NA, nrow=nrow(test), ncol=99)
```

```r
tn3 = matrix(NA, nrow=nrow(test), ncol=99)

for (i in 1:99) {
    tp3[,i] = ifelse(test$fraud=="1" & test[[1]]>0.01*i,1,0)
    tn3[,i] = ifelse(test$fraud=="0" & test[[1]]<=0.01*i,1,0)
}

trueclassrate3 = matrix(NA, nrow=99, ncol=2)
for (i in 1:99) {
    trueclassrate3[i,1] = 0.01*i
    trueclassrate3[i,2] = sum(tp3[,i]+tn3[,i])/nrow(test)
}

print(trueclassrate3[which(trueclassrate3[,2]==max(trueclassrate3[,2])),])
```

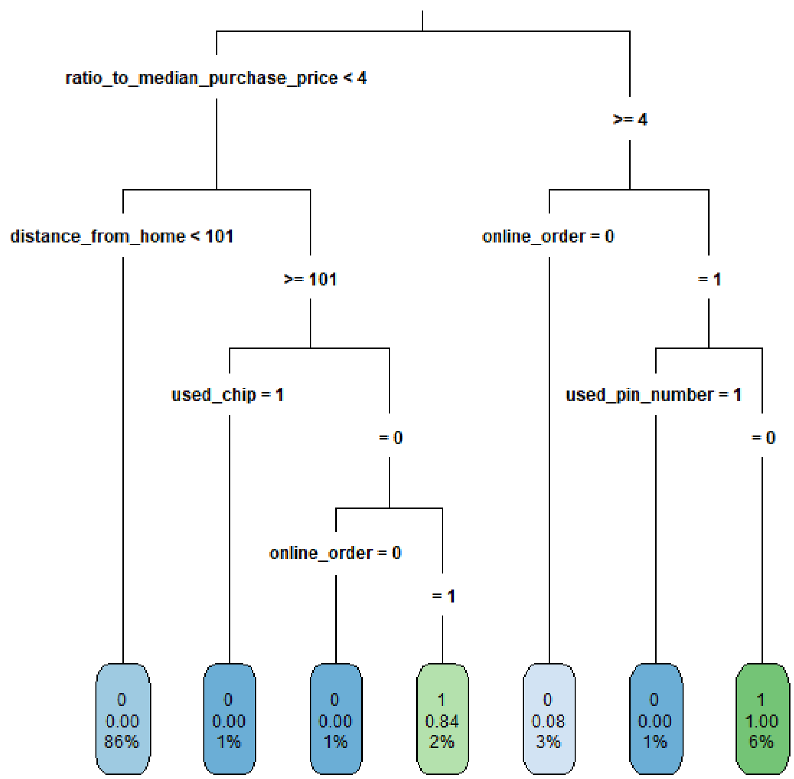
```
[,1]      [,2]
 [1,] 0.08 0.9850746
 [2,] 0.09 0.9850746
 [3,] 0.10 0.9850746
 [4,] 0.11 0.9850746
 [5,] 0.12 0.9850746
 [6,] 0.13 0.9850746
 [7,] 0.14 0.9850746
 [8,] 0.15 0.9850746
 [9,] 0.16 0.9850746
[10,] 0.17 0.9850746
[11,] 0.18 0.9850746
[12,] 0.19 0.9850746
[13,] 0.20 0.9850746
[14,] 0.21 0.9850746
[15,] 0.22 0.9850746
[16,] 0.23 0.9850746
[17,] 0.24 0.9850746
[18,] 0.25 0.9850746
[19,] 0.26 0.9850746
[20,] 0.27 0.9850746
[21,] 0.28 0.9850746
[22,] 0.29 0.9850746
[23,] 0.30 0.9850746
[24,] 0.31 0.9850746
[25,] 0.32 0.9850746
[26,] 0.33 0.9850746
[27,] 0.34 0.9850746
[28,] 0.35 0.9850746
[29,] 0.36 0.9850746
[30,] 0.37 0.9850746
[31,] 0.38 0.9850746
[32,] 0.39 0.9850746
[33,] 0.40 0.9850746
[34,] 0.41 0.9850746
[35,] 0.42 0.9850746
[36,] 0.43 0.9850746
[37,] 0.44 0.9850746
[38,] 0.45 0.9850746
[39,] 0.46 0.9850746
[40,] 0.47 0.9850746
[41,] 0.48 0.9850746
```

```
[42,] 0.49 0.9850746
[43,] 0.50 0.9850746
[44,] 0.51 0.9850746
[45,] 0.52 0.9850746
[46,] 0.53 0.9850746
[47,] 0.54 0.9850746
[48,] 0.55 0.9850746
[49,] 0.56 0.9850746
[50,] 0.57 0.9850746
[51,] 0.58 0.9850746
[52,] 0.59 0.9850746
[53,] 0.60 0.9850746
[54,] 0.61 0.9850746
[55,] 0.62 0.9850746
[56,] 0.63 0.9850746
[57,] 0.64 0.9850746
[58,] 0.65 0.9850746
[59,] 0.66 0.9850746
[60,] 0.67 0.9850746
[61,] 0.68 0.9850746
[62,] 0.69 0.9850746
[63,] 0.70 0.9850746
[64,] 0.71 0.9850746
[65,] 0.72 0.9850746
[66,] 0.73 0.9850746
[67,] 0.74 0.9850746
[68,] 0.75 0.9850746
[69,] 0.76 0.9850746
[70,] 0.77 0.9850746
[71,] 0.78 0.9850746
[72,] 0.79 0.9850746
[73,] 0.80 0.9850746
[74,] 0.81 0.9850746
[75,] 0.82 0.9850746
[76,] 0.83 0.9850746
```

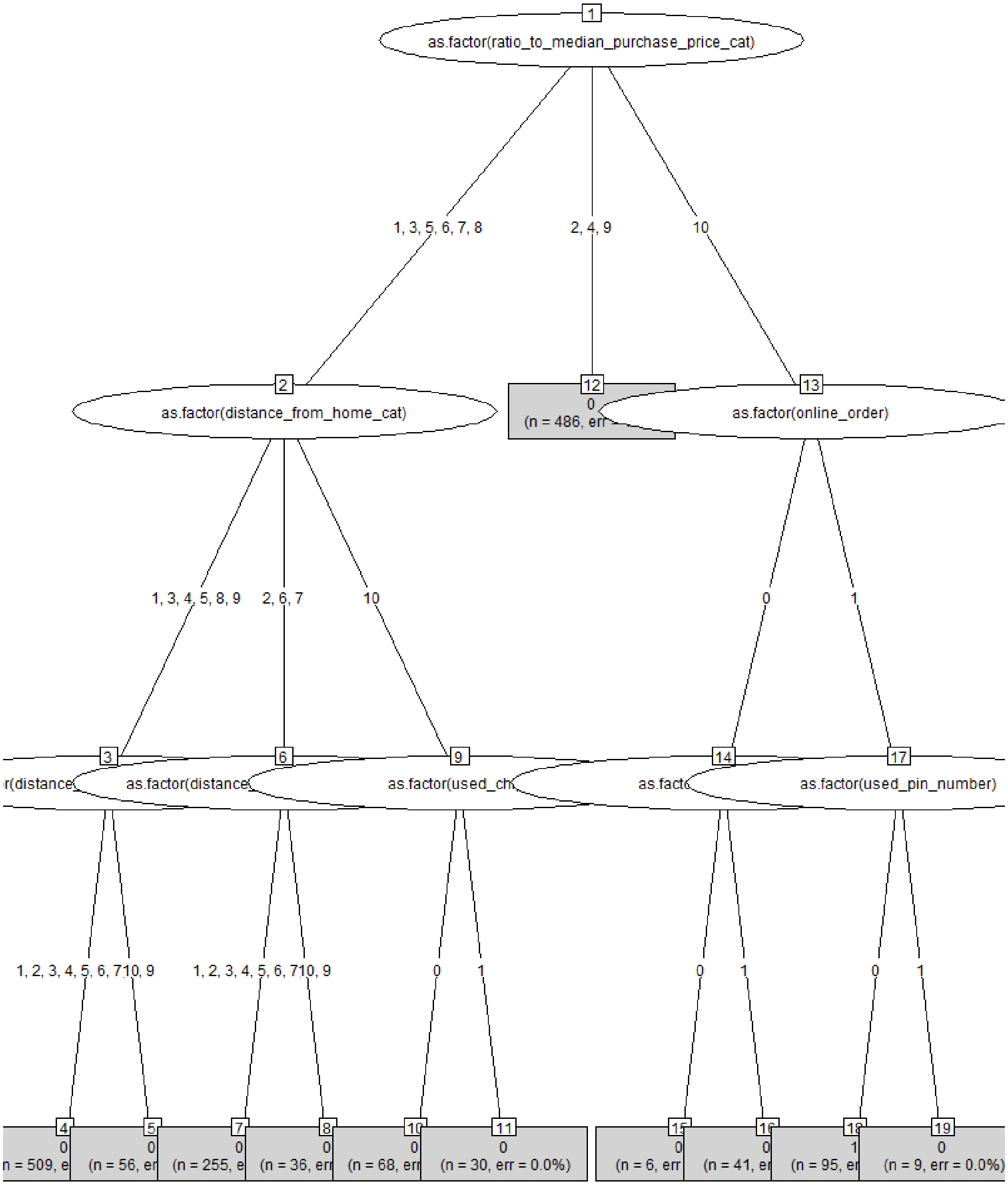

ratio_to_median_purchase_price < 4
>= 4
distance_from_home < 101
>= 101
online_order = 0
= 1
used_chip = 1
= 0
used_pin_number = 1
= 0
online_order = 0
= 1
0
0.00
86%
0
0.00
1%
0
0.00
1%
1
0.84
2%
0
0.08
3%
0
0.00
1%
1
1.00
6%

```
      [,1]      [,2]
 [1,] 0.08 0.9850746
 [2,] 0.09 0.9850746
 [3,] 0.10 0.9850746
 [4,] 0.11 0.9850746
 [5,] 0.12 0.9850746
 [6,] 0.13 0.9850746
 [7,] 0.14 0.9850746
 [8,] 0.15 0.9850746
 [9,] 0.16 0.9850746
[10,] 0.17 0.9850746
[11,] 0.18 0.9850746
[12,] 0.19 0.9850746
[13,] 0.20 0.9850746
[14,] 0.21 0.9850746
[15,] 0.22 0.9850746
[16,] 0.23 0.9850746
[17,] 0.24 0.9850746
[18,] 0.25 0.9850746
[19,] 0.26 0.9850746
[20,] 0.27 0.9850746
[21,] 0.28 0.9850746
[22,] 0.29 0.9850746
[23,] 0.30 0.9850746
[24,] 0.31 0.9850746
[25,] 0.32 0.9850746
[26,] 0.33 0.9850746
[27,] 0.34 0.9850746
[28,] 0.35 0.9850746
[29,] 0.36 0.9850746
[30,] 0.37 0.9850746
[31,] 0.38 0.9850746
[32,] 0.39 0.9850746
[33,] 0.40 0.9850746
[34,] 0.41 0.9850746
[35,] 0.42 0.9850746
[36,] 0.43 0.9850746
[37,] 0.44 0.9850746
[38,] 0.45 0.9850746
[39,] 0.46 0.9850746
[40,] 0.47 0.9850746
[41,] 0.48 0.9850746
[42,] 0.49 0.9850746
[43,] 0.50 0.9850746
[44,] 0.51 0.9850746
[45,] 0.52 0.9850746
[46,] 0.53 0.9850746
[47,] 0.54 0.9850746
[48,] 0.55 0.9850746
[49,] 0.56 0.9850746
[50,] 0.57 0.9850746
[51,] 0.58 0.9850746
[52,] 0.59 0.9850746
[53,] 0.60 0.9850746
[54,] 0.61 0.9850746
[55,] 0.62 0.9850746
[56,] 0.63 0.9850746
[57,] 0.64 0.9850746
[58,] 0.65 0.9850746
[59,] 0.66 0.9850746
[60,] 0.67 0.9850746
[61,] 0.68 0.9850746
[62,] 0.69 0.9850746
[63,] 0.70 0.9850746
```

```
[64,] 0.71 0.9850746
[65,] 0.72 0.9850746
[66,] 0.73 0.9850746
[67,] 0.74 0.9850746
[68,] 0.75 0.9850746
[69,] 0.76 0.9850746
[70,] 0.77 0.9850746
[71,] 0.78 0.9850746
[72,] 0.79 0.9850746
[73,] 0.80 0.9850746
[74,] 0.81 0.9850746
[75,] 0.82 0.9850746
[76,] 0.83 0.9850746
```


ratio_to_median_purchase_price < 4
>= 4
distance_from_home < 101
>= 101
online_order = 0
= 1
used_chip = 1
= 0
used_pin_number = 1
= 0
online_order = 0
= 1
0
0.00
86%
0
0.00
1%
0
0.00
1%
1
0.84
2%
0
0.08
3%
0
0.00
1%
1
1.00
6%

```
     [,1]      [,2]
[1,] 0.95 0.1418093
[2,] 0.96 0.1418093
[3,] 0.97 0.1418093
[4,] 0.98 0.1418093
[5,] 0.99 0.1418093
```



as.factor(ratio_to_median_purchase_price_cat)
1, 3, 5, 6, 7, 8
2, 4, 9
10
as.factor(distance_from_home_cat)
0
(n = 486, err
as.factor(online_order)
1, 3, 4, 5, 8, 9
2, 6, 7
10
0
1
as.factor(used_pin_number)
1, 2, 3, 4, 5, 6, 710, 9
1, 2, 3, 4, 5, 6, 710, 9
0
1
0
1
0
1
(n = 30, err = 0.0%)
(n = 9, err = 0.0%)

Python Code

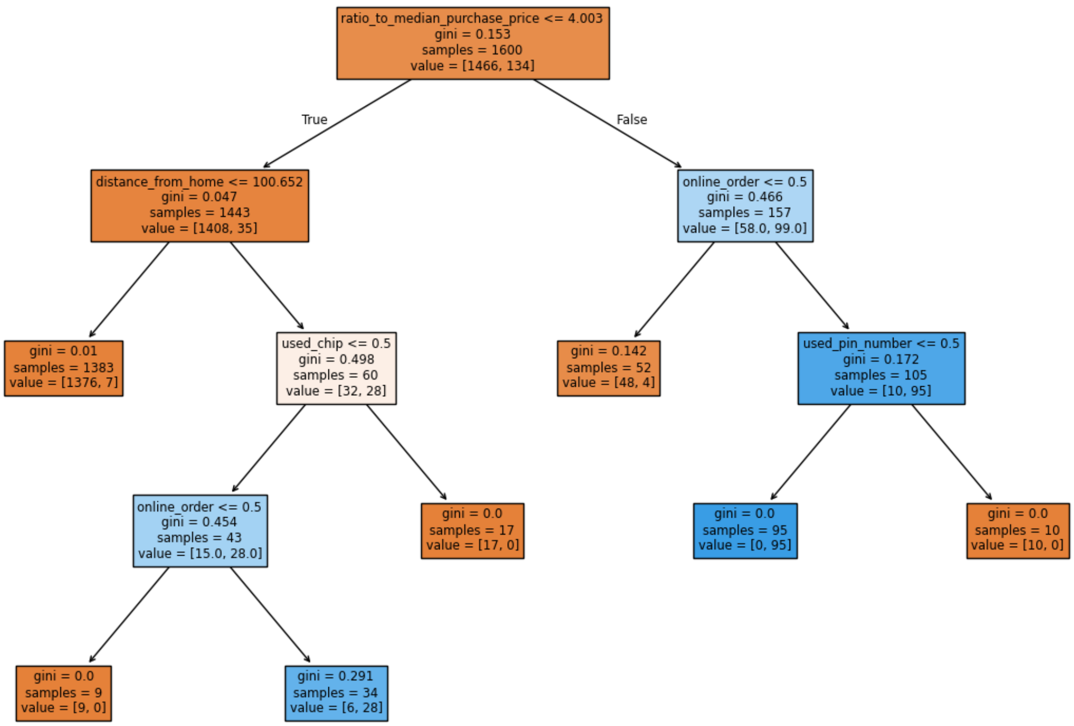
```
# STAT 574 HW1 Problem 2: Card Transaction Data (Python)

#Import all necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from chefboost import Chefboost

# Import data and conduct preprocessing.

card_path = "C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/\
card_transdata.csv"
card_data = pd.read_csv(card_path)
X = card_data.iloc[:,0:7].values
y = card_data.iloc[:,7].values

# (a) Spltting the data into 80% training and 20% testing sets. Building
# a classification tree for fraudulent activity using the Gini criterion.
# Pruning tree using the cost-complexity pruning algorithm.

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,
                                                    random_state=122470)

# Fitting the binary classification tree with Gini splitting criterion.

gini_tree = DecisionTreeClassifier(max_leaf_nodes=7, criterion='gini',
                                   random_state=590520)
gini_tree_fit = gini_tree.fit(X_train, y_train)

# Plotting fitted tree

fig = plt.figure(figsize=(15, 10))
tree.plot_tree(gini_tree_fit, feature_names=['distance_from_home',
    'distance_from_last_transaction', 'ratio_to_median_purchase_price',
    'repeat_retailer', 'used_chip', 'used_pin_number', 'online_order'],
    filled=True)
# (b) Compute the prediction accuracy for the training data, using the range
# of classification thresholds between 0.01 and 0.99. What thresholds
```

```python
# correspond to the largest prediction accuracy?

def accuracy():
    y_pred = gini_tree_fit.predict_proba(X_test)
    total = len(y_pred)
    trueclassrate = []
    cutoff = []
    for i in range(99):
        tp = 0
        tn = 0
        cutoff.append(0.01*(i+1))
        for sub1, sub2 in zip(y_pred[::,1], y_test):
            tp_ind = 1 if (sub1>0.01*(i+1) and sub2==1) else 0
            tn_ind = 1 if (sub1<0.01*(i+1) and sub2==0) else 0
            tp += tp_ind
            tn += tn_ind
        rate = (tp+tn)/total
        trueclassrate.append(rate)

    df = pd.DataFrame({'trueclassrate': trueclassrate, 'cutoff': cutoff})
    max_rate = max(trueclassrate)
    optimal = df[df['trueclassrate']==max_rate]
    print(optimal)

accuracy()

# (c) Fitting binary tree using Entropy Splitting Criterion and
# cost-complexity pruning algorithm

entropy_tree = DecisionTreeClassifier(max_leaf_nodes=7, criterion='entropy',
                                   random_state=590520)
entropy_tree_fit = entropy_tree.fit(X_train, y_train)

# Plotting fitted tree

fig = plt.figure(figsize=(15, 10))
tree.plot_tree(entropy_tree_fit, feature_names=['distance_from_home',
    'distance_from_last_transaction', 'ratio_to_median_purchase_price',
    'repeat_retailer', 'used_chip', 'used_pin_number', 'online_order'],
    filled=True)

#(d) Computing the prediction accuracy of the entropy tree for the training
# data, using the cutoffs for the predicted probability of fraud ranging
# between 0.01 and 0.99. List the cutoffs that give the maximum prediction
# accuracy.
```

```python
def accuracy2():
    y_pred2 = entropy_tree_fit.predict_proba(X_test)
    total2 = len(y_pred2)
    trueclassrate2 = []
    cutoff2 = []
    for i in range(99):
        tp = 0
        tn = 0
        cutoff2.append(0.01*(i+1))
        for sub1, sub2 in zip(y_pred2[::,1], y_test):
            tp_ind = 1 if (sub1>0.01*(i+1) and sub2==1) else 0
            tn_ind = 1 if (sub1<0.01*(i+1) and sub2==0) else 0
            tp += tp_ind
            tn += tn_ind
        rate2 = (tp+tn)/total2
        trueclassrate2.append(rate2)

    df = pd.DataFrame({'trueclassrate': trueclassrate2, 'cutoff': cutoff2})
    max_rate2 = max(trueclassrate2)
    optimal2 = df[df['trueclassrate']==max_rate2]
    print(optimal2)

accuracy2()
# (e) Fitting binary classification tree using CHAID criterion and cost-
# complexity pruning algorithm.

transaction = pd.read_csv(card_path)
fraud_code = {1: 'fraud', 0: 'not fraud'}
transaction['fraud'] = transaction['fraud'].map(fraud_code)
X = transaction.iloc[:,0:7].values
y = transaction.iloc[:,7].values

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,
                                                    random_state=868692)
X_train = pd.DataFrame(X_train, columns=['distance_from_home',
    'distance_from_last_transaction', 'ratio_to_median_purchase_price',
    'repeat_retailer', 'used_chip', 'used_pin_number', 'online_order'])

y_train = pd.DataFrame(y_train, columns=['fraud'])
train_data = pd.concat([X_train, y_train], axis=1)

config = {'algorithm': 'CHAID', 'max_depth': 7}
tree_chaid = Chefboost.fit(train_data, config, target_label='fraud')
#(f) Computing the prediction accuracy of the CHAID tree for the training
```

```
# data, using the cutoffs for the predicted probability of fraud ranging
# between 0.01 and 0.99. List the cutoffs that give the maximum prediction
# accuracy.

transaction = pd.read_csv(card_path)

X = transaction.iloc[:,0:7].values
y = transaction.iloc[:,7].values

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,
                                                    random_state=868692)
X_train = pd.DataFrame(X_train, columns=['distance_from_home',
    'distance_from_last_transaction', 'ratio_to_median_purchase_price',
    'repeat_retailer', 'used_chip', 'used_pin_number', 'online_order'])

y_train = pd.DataFrame(y_train, columns=['fraud'])
train_data = pd.concat([X_train, y_train], axis=1)

config = {'algorithm': 'CHAID', 'max_depth': 7}
tree_chaid = Chefboost.fit(train_data, config, target_label='fraud')

X_test = pd.DataFrame(X_test, columns=['distance_from_home',
    'distance_from_last_transaction', 'ratio_to_median_purchase_price',
    'repeat_retailer', 'used_chip', 'used_pin_number', 'online_order'])

def accuracy3():
    y_pred3 = []
    for i in range(len(y_test)):
        y_pred3.append(Chefboost.predict(tree_chaid, X_test.iloc[i,:]))
    total3 = len(y_pred3)
    trueclassrate3 = []
    cutoff3 = []
    for i in range(99):
        tp = 0
        tn = 0
        cutoff3.append(0.01*(i+1))
        for sub1, sub2 in zip(y_pred3, y_test):
            tp_ind = 1 if (float(sub1)>0.01*(i+1) and sub2==1) else 0
            tn_ind = 1 if (float(sub1)<0.01*(i+1) and sub2==0) else 0
            tp += tp_ind
            tn += tn_ind
        rate3 = (tp+tn)/total3
        trueclassrate3.append(rate3)

    df = pd.DataFrame({'trueclassrate': trueclassrate3, 'cutoff': cutoff3})
```

```
    max_rate3 = max(trueclassrate3)
    optimal3 = df[df['trueclassrate']==max_rate3]
    print(optimal3)

accuracy3()
```



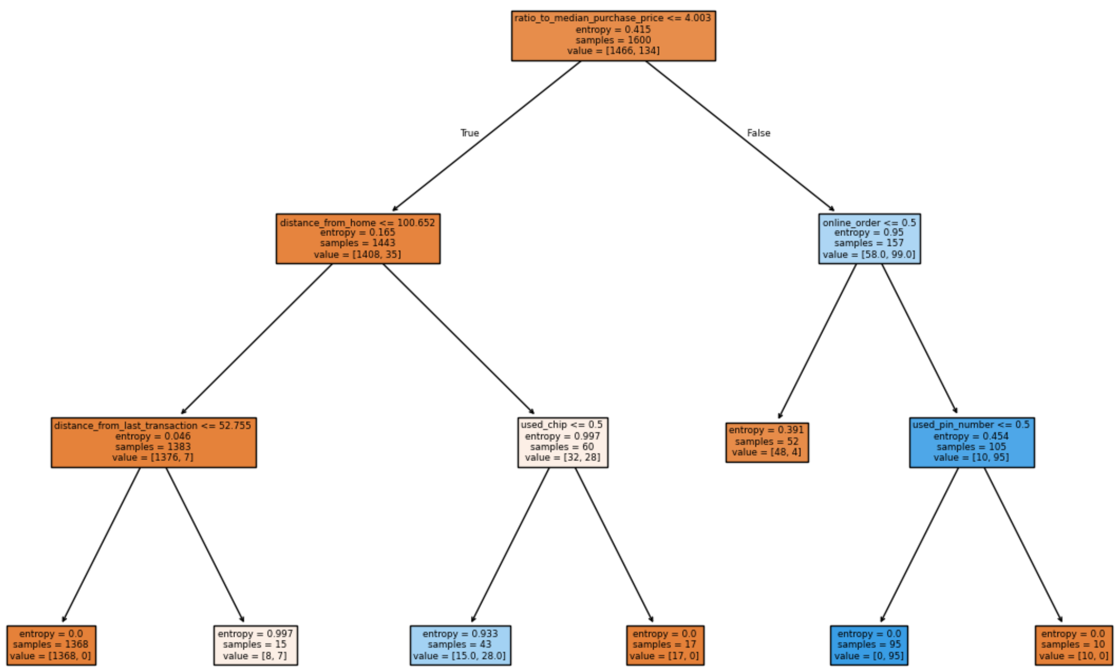
Gini Binary Classification Tree

Cutoffs for Gini Classification Tree

	trueclassrate	cutoff
7	0.99	0.08
8	0.99	0.09
9	0.99	0.10
10	0.99	0.11
11	0.99	0.12
..	...	...

77	0.99	0.78
78	0.99	0.79
79	0.99	0.80
80	0.99	0.81
81	0.99	0.82

[75 rows x 2 columns]



Entropy Binary Classification Tree

Cutoffs for Entropy tree

	trueclassrate	cutoff
7	0.9825	0.08
8	0.9825	0.09
9	0.9825	0.10

10	0.9825	0.11
11	0.9825	0.12
12	0.9825	0.13
13	0.9825	0.14
14	0.9825	0.15
15	0.9825	0.16
16	0.9825	0.17
17	0.9825	0.18
18	0.9825	0.19
19	0.9825	0.20
20	0.9825	0.21
21	0.9825	0.22
22	0.9825	0.23
23	0.9825	0.24
24	0.9825	0.25
25	0.9825	0.26
26	0.9825	0.27
27	0.9825	0.28
28	0.9825	0.29
29	0.9825	0.30
30	0.9825	0.31
31	0.9825	0.32
32	0.9825	0.33
33	0.9825	0.34
34	0.9825	0.35
35	0.9825	0.36
36	0.9825	0.37
37	0.9825	0.38

38	0.9825	0.39
39	0.9825	0.40
40	0.9825	0.41
41	0.9825	0.42
42	0.9825	0.43
43	0.9825	0.44
44	0.9825	0.45
45	0.9825	0.46
46	0.9825	0.47
47	0.9825	0.48
48	0.9825	0.49
49	0.9825	0.50
50	0.9825	0.51
51	0.9825	0.52
52	0.9825	0.53
53	0.9825	0.54
54	0.9825	0.55
55	0.9825	0.56
56	0.9825	0.57
57	0.9825	0.58
58	0.9825	0.59
59	0.9825	0.60
60	0.9825	0.61
61	0.9825	0.62
62	0.9825	0.63
63	0.9825	0.64
64	0.9825	0.65

CHAID Binary Classification Tree

25-02-11 16:20:55 - CHAID tree is going to be built...

25-02-11 16:21:01 - -------------------------

25-02-11 16:21:01 - finished in 6.514265298843384 seconds

25-02-11 16:21:02 - -------------------------

25-02-11 16:21:02 - Evaluate train set

25-02-11 16:21:02 - -------------------------

25-02-11 16:21:02 - Accuracy: 99.0625% on 1600 instances

25-02-11 16:21:02 - Labels: ['not fraud' 'fraud']

25-02-11 16:21:02 - Confusion matrix: [[1459, 14], [1, 126]]

25-02-11 16:21:02 - Precision: 99.0496%, Recall: 99.9315%, F1: 99.4886%

Cutoffs for CHAID Classification Tree

```
trueclassrate  cutoff
0           0.99    0.01
1           0.99    0.02
2           0.99    0.03
3           0.99    0.04
4           0.99    0.05
..           ...     ...
94           0.99    0.95
95           0.99    0.96
96           0.99    0.97
97           0.99    0.98
98           0.99    0.99

[99 rows x 2 columns]
```

The Gini Tree yields the largest maximum prediction accuracy according to the results of the three codes.

Problem 3.

SAS Code

```
proc import out=card_data
datafile="C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/card_transdata.csv
"
dbms=csv replace;

/* Splitting the data into 80% training and 20% testing sets*/

proc surveyselect data=card_data rate=0.8 seed=122470
out=card_data outall method=srs;
run;

/*Gini-splitting and cost-complexity pruning*/

proc hpsplit data=card_data maxdepth=7;
    class repeat_retailer used_chip used_pin_number online_order fraud;
    model fraud(event="1")=distance_from_home distance_from_last_transaction
ratio_to_median_purchase_price repeat_retailer used_chip used_pin_number
online_order;
    grow gini;
    prune costcomplexity;
    partition rolevar=selected(train="1");
    output out=predicted;
    ID selected;
run;

/* (a) Computing the confusion matrix using the 0.5 cutoff for the
predicted probability of fraud*/

data test;
    set predicted;
    if(selected="0");
    tp = (P_fraud1 > 0.5 and fraud="1");
    fp = (P_fraud1 > 0.5 and fraud="0");
    tn = (P_fraud0 > 0.5 and fraud="0");
    fn = (P_fraud0 > 0.5 and fraud="1");
run;

proc sql;
    create table confusion as
    select sum(tp) as tp, sum(fp) as fp, sum(tn) as tn,
    sum(fn) as fn, count(*) as total
    from test;
    select * from confusion;
```

```
quit;

/* (b) Compute the prediction performance measures: accuracy,
misclassification rate, sensitivity, False positive rate, precision,
negative predictive value, F1 score*/

proc sql;
    select (tp+tn)/total as accuracy, (fp+fn)/total as
    misclassrate, tp/(tp+fn) as sensitivity,
    fn/(tp+fn) as FNR, tn/(fp+tn) as specificity,
    fp/(fp+tn) as FPR, tp/(tp+fp) as precision,
    tn/(fn+tn) as NPV, 2*tp/(2*tp+fn+fp) as F1score
    from confusion;
quit;
```

The SAS System

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	HOSPITAL
Random Number Seed	479576
Sampling Rate	0.8
Sample Size	3047
Selection Probability	0.800158
Sampling Weight	0
Output Data Set	HOSPITAL

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

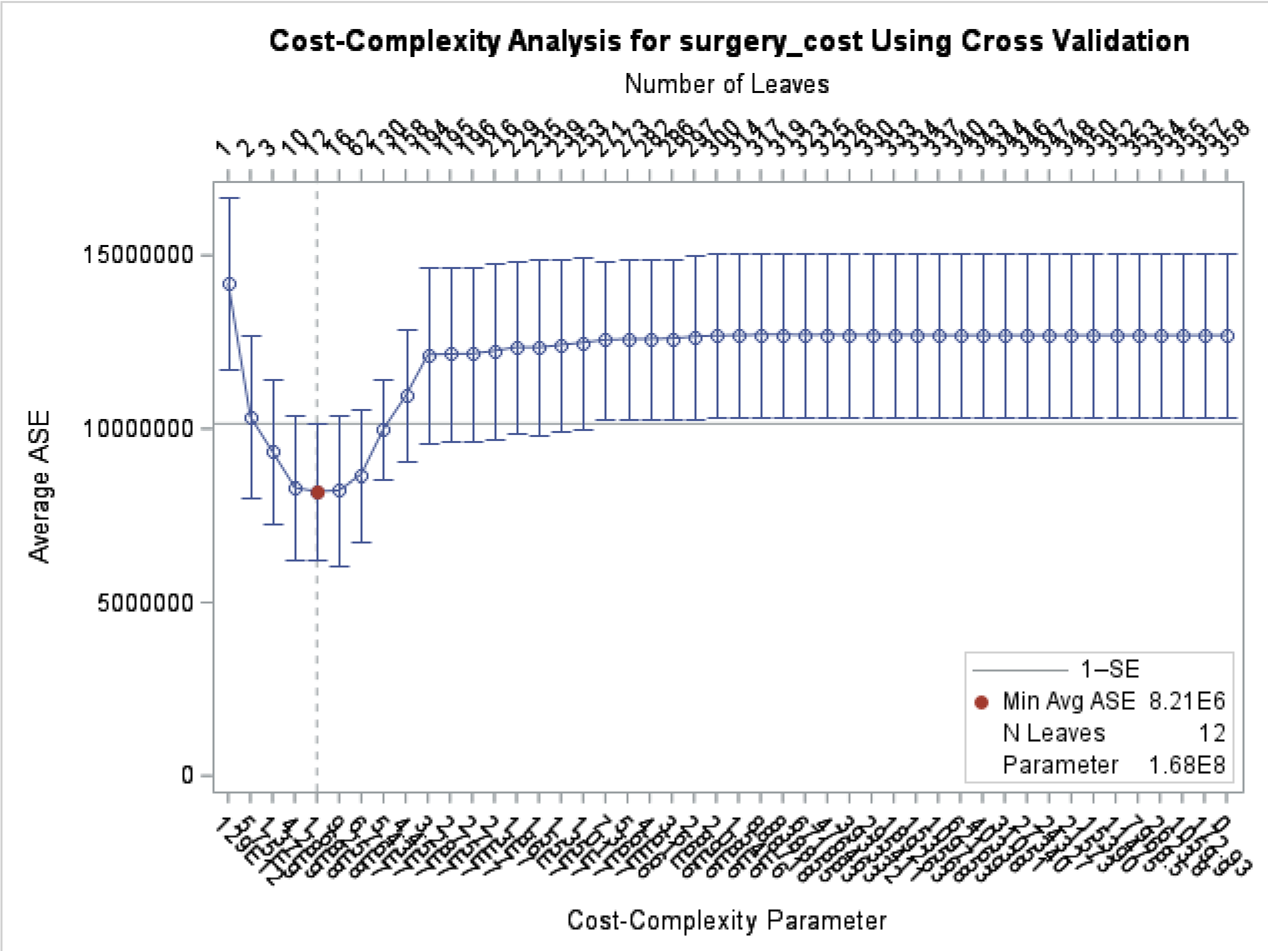
Model Information

Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	13

Number of Observations Read	3047
Number of Observations Used	3047

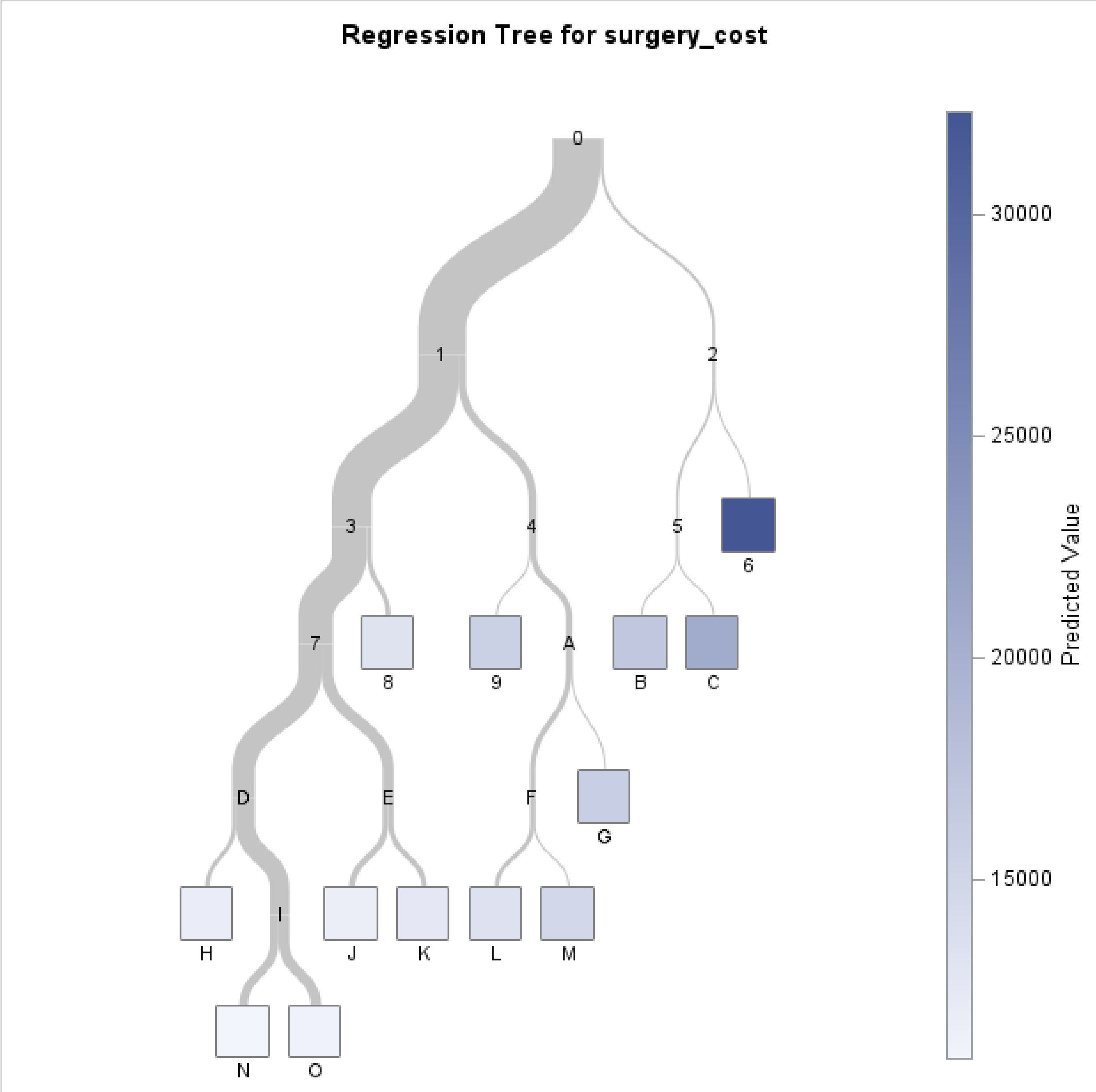
The SAS System

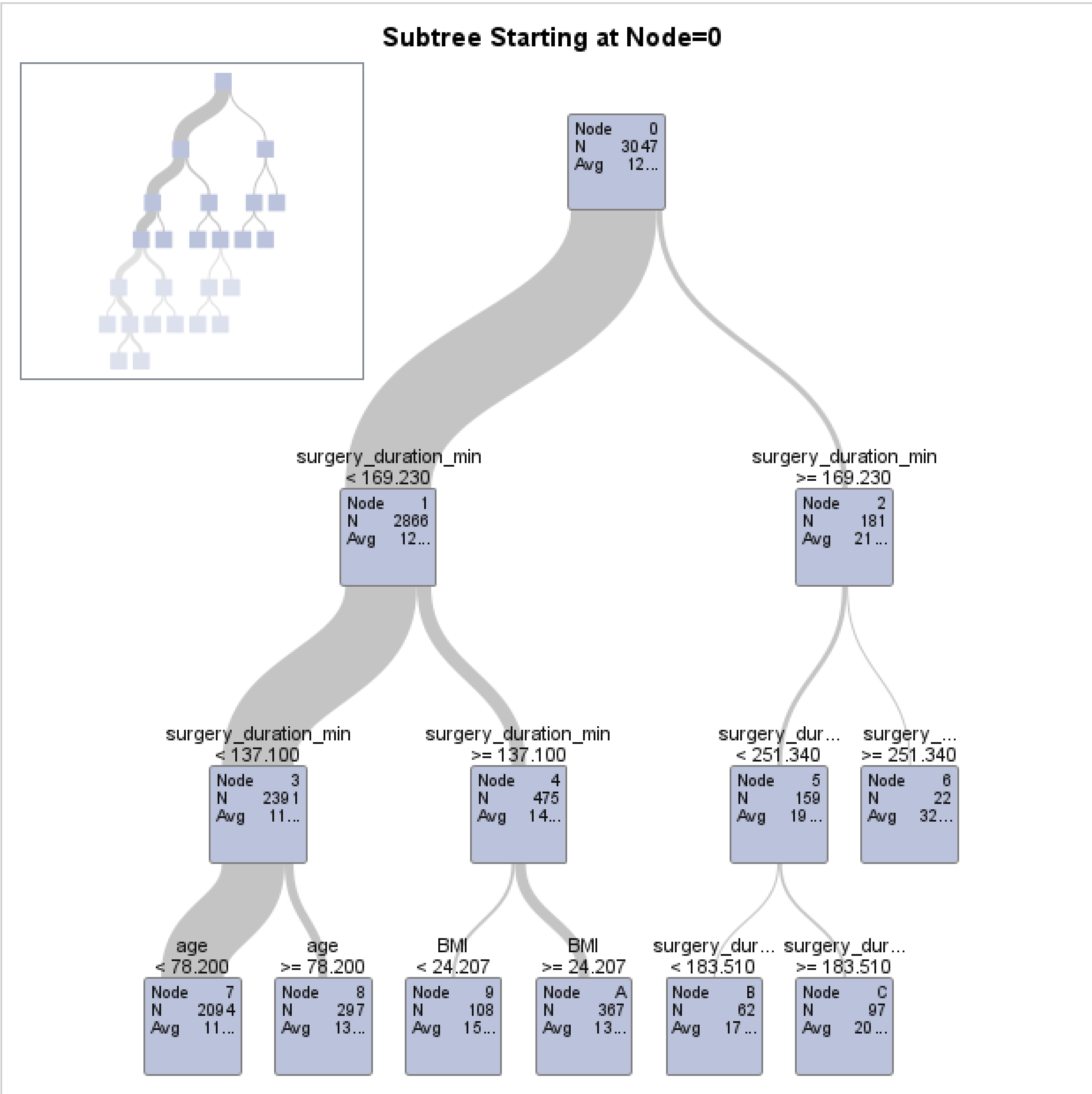
The HPSPLIT Procedure



The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_dur...
< 251.340
Node 5
N 159
Avg 19...
surgery_...
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
13	7242180	2.207E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	140554	6
age	0.2287	32141.5	4
BMI	0.1187	16676.8	1
ASA	0.0862	12112.9	1

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

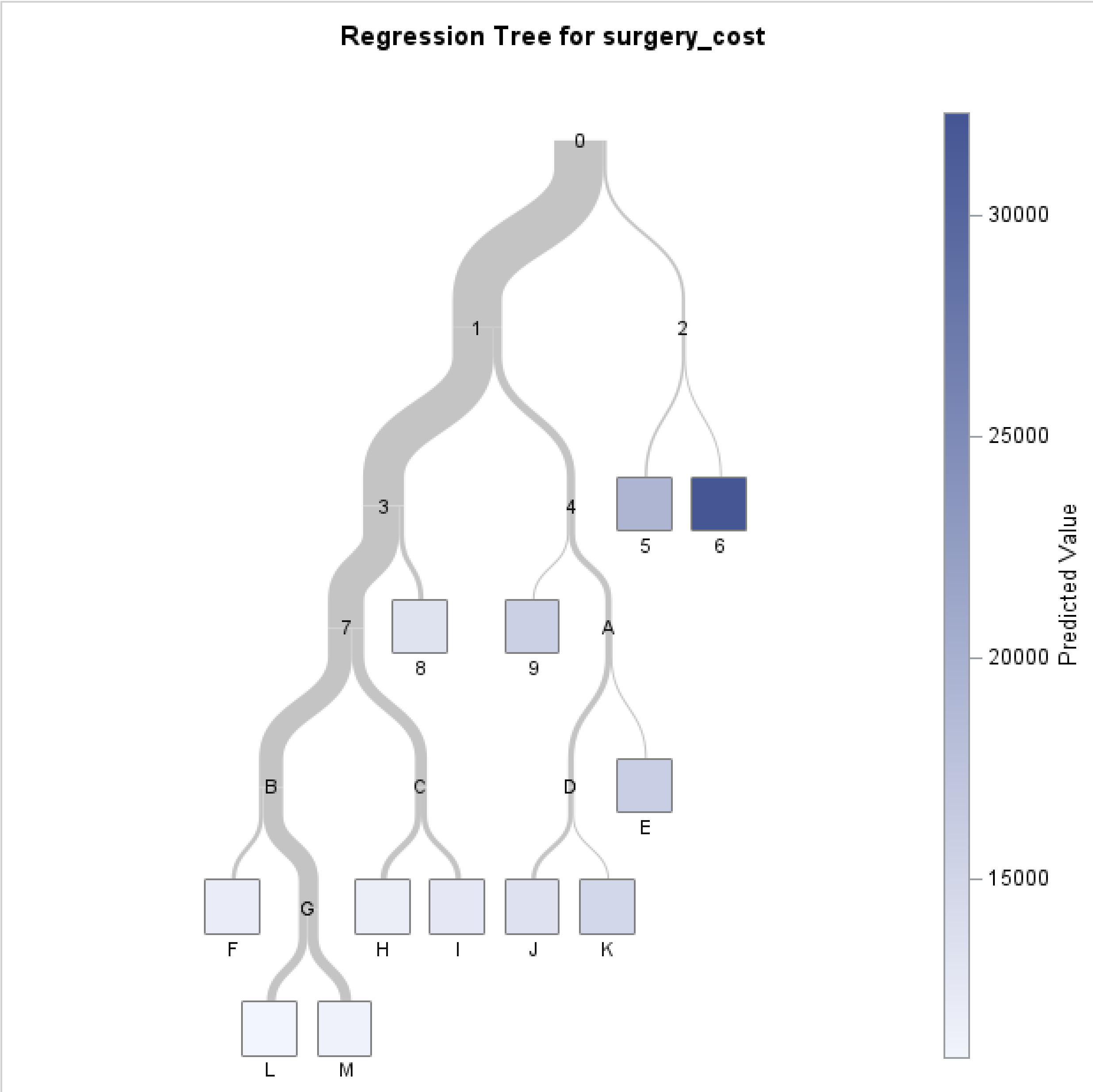
Model Information

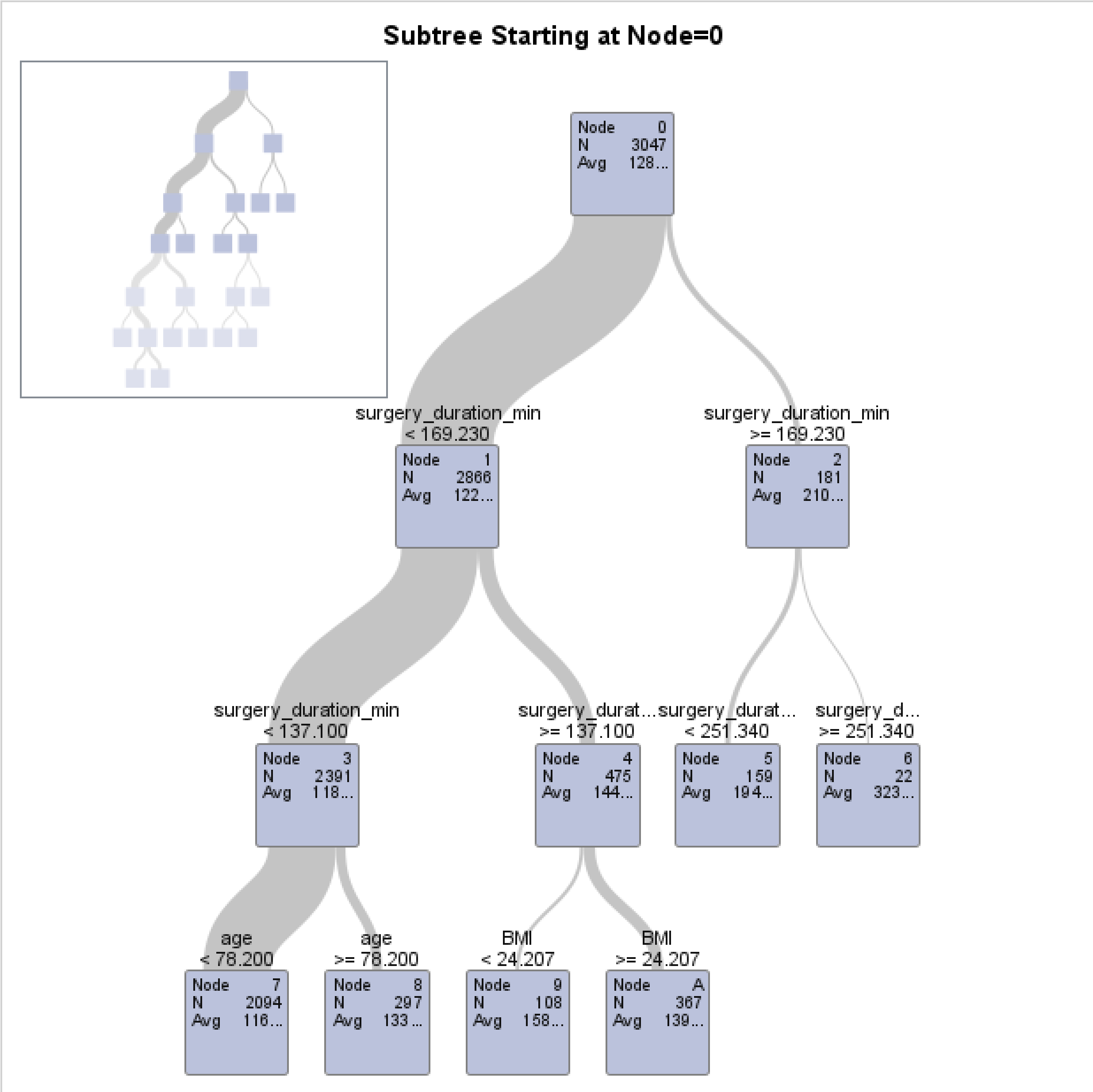
Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	12

Number of Observations Read	3047
Number of Observations Used	3047

The SAS System

The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 3047
Avg 128...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 122...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 210...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 118...
surgery_durat...
>= 137.100
Node 4
N 475
Avg 144...
surgery_durat...
< 251.340
Node 5
N 159
Avg 194...
surgery_d...
>= 251.340
Node 6
N 22
Avg 323...
age
< 78.200
Node 7
N 2094
Avg 116...
age
>= 78.200
Node 8
N 297
Avg 133...
BMI
< 24.207
Node 9
N 108
Avg 158...
BMI
>= 24.207
Node A
N 367
Avg 139...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
12	7415512	2.26E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	138663	5
age	0.2318	32141.5	4
BMI	0.1203	16676.8	1
ASA	0.0874	12112.9	1

accuracy10	accuracy15	accuracy20
0.51117	0.660972	0.78318

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

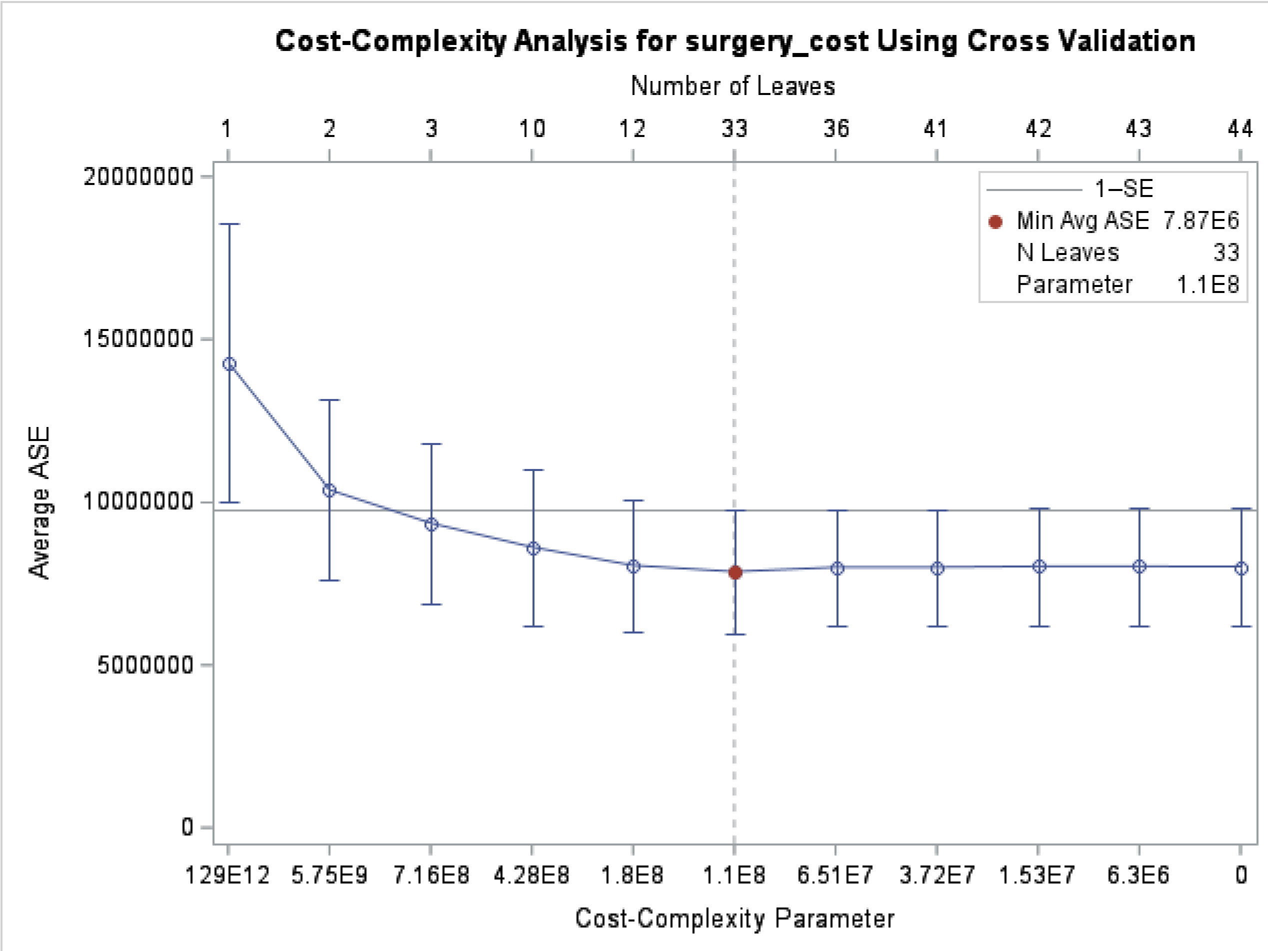
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

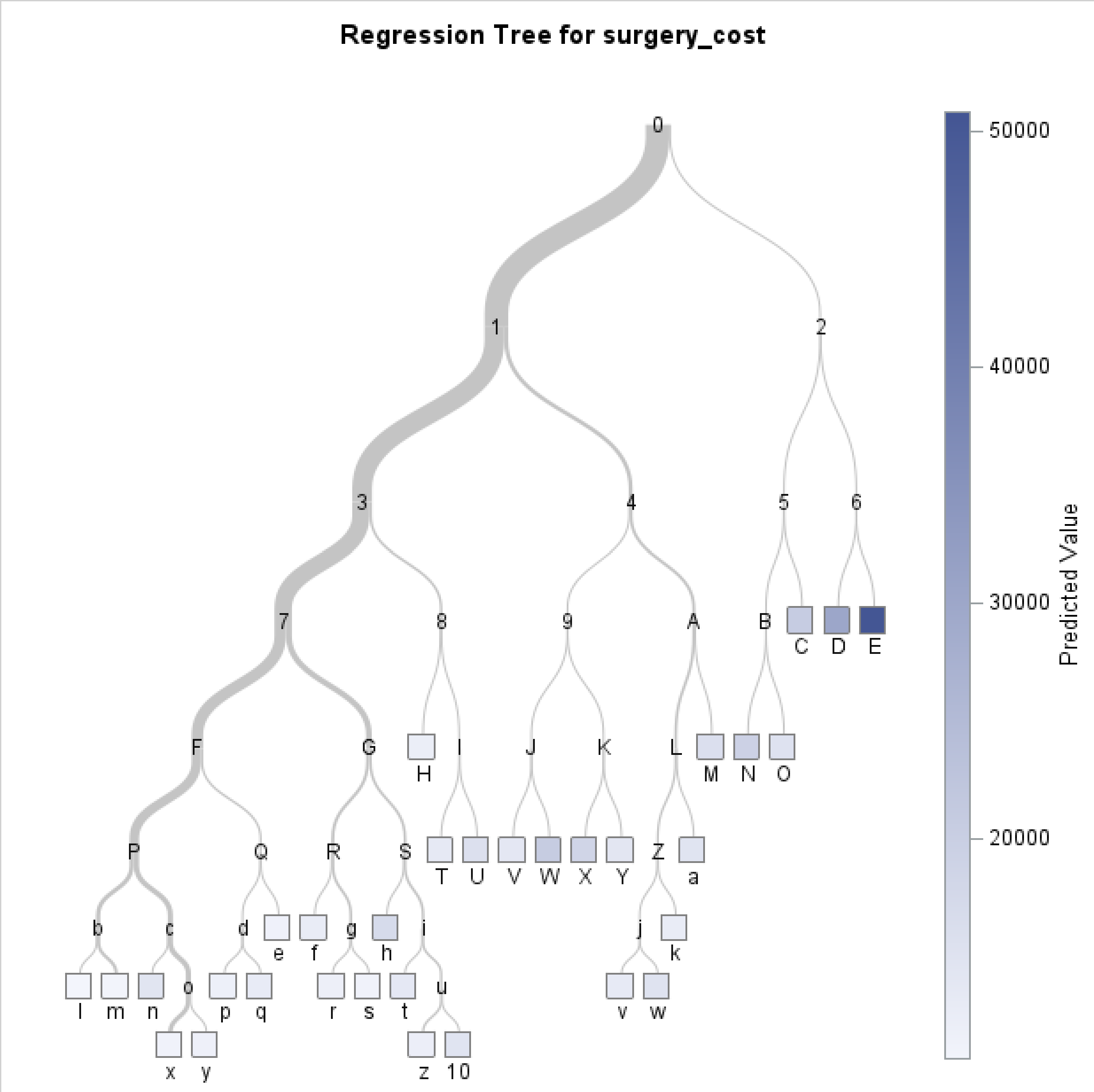
Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	32

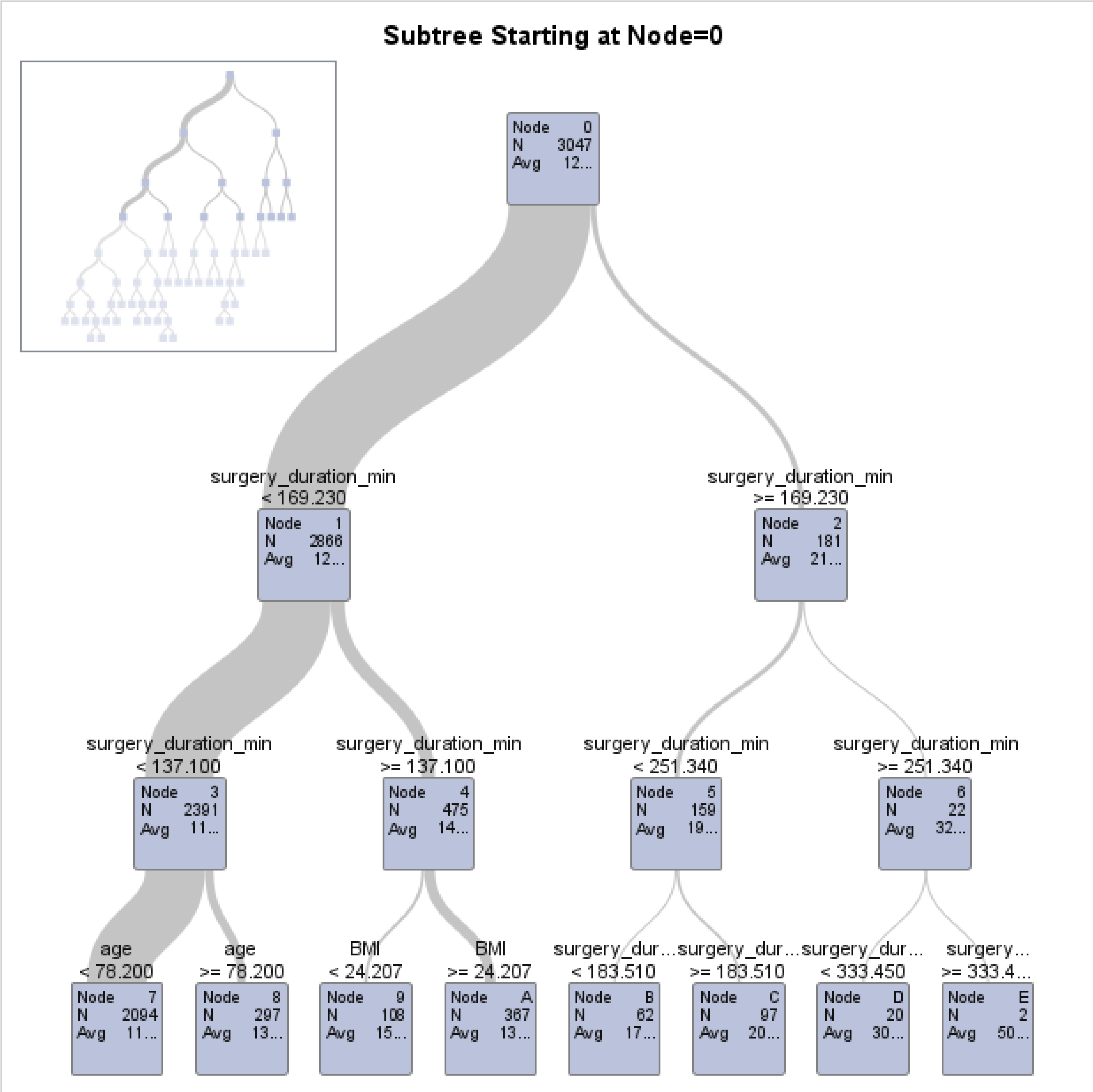
Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure



The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_duration_min
< 251.340
Node 5
N 159
Avg 19...
surgery_duration_min
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...
surgery_dur...
< 333.450
Node D
N 20
Avg 30...
surgery...
>= 333.4...
Node E
N 2
Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
32	6469173	1.971E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2588	37412.2	9
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

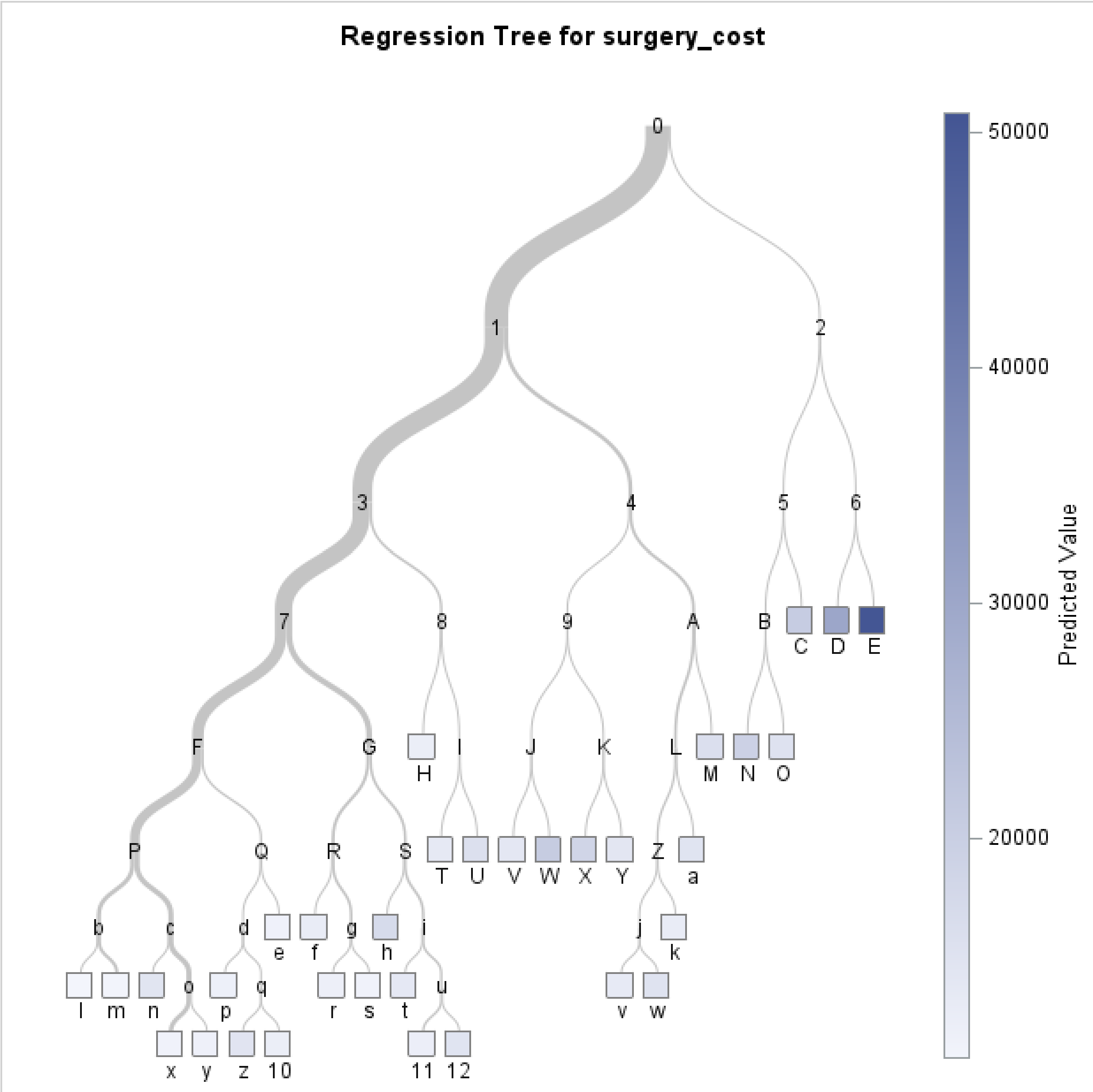
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

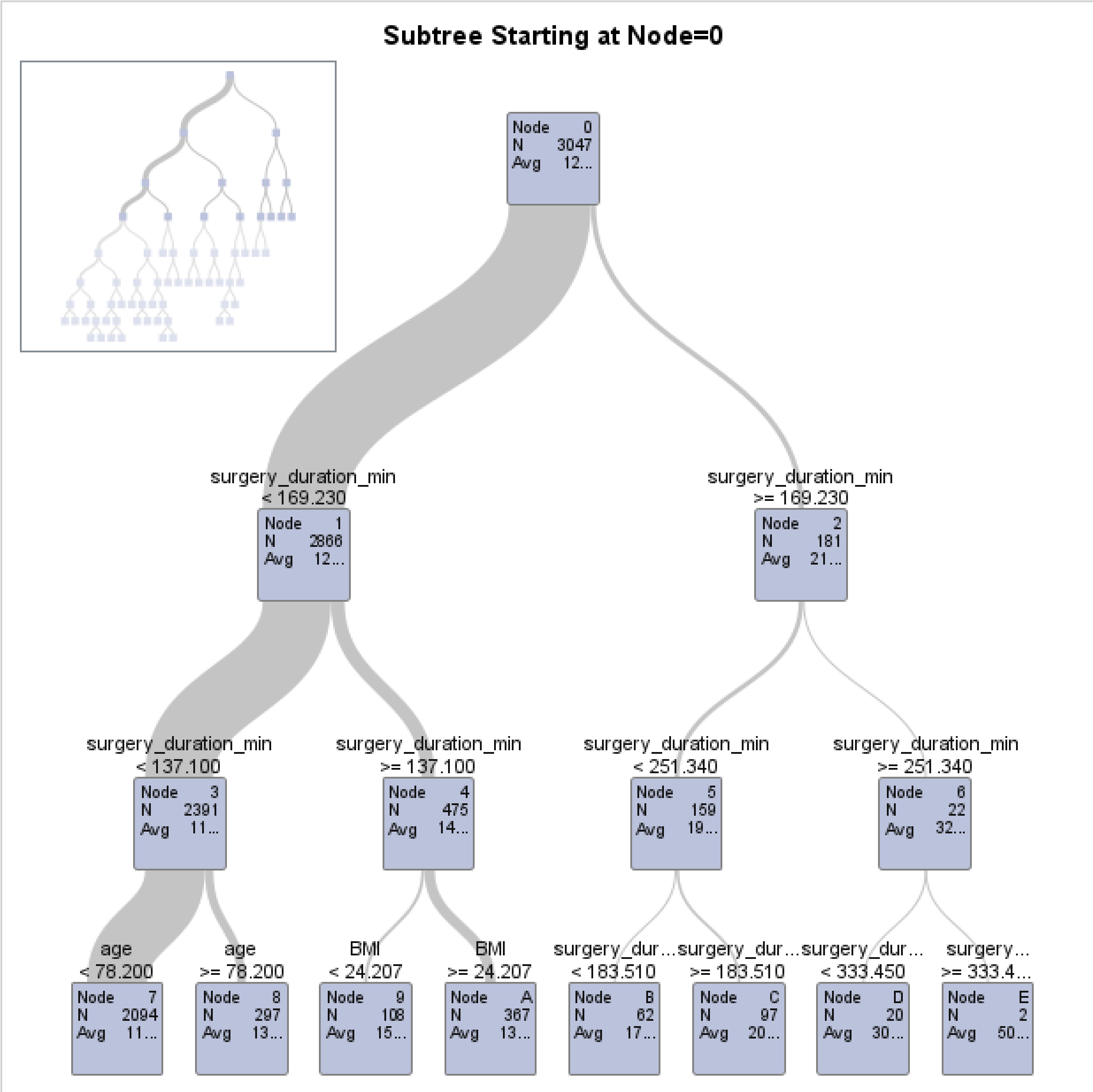
Model Information

Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	33

Number of Observations Read	3047
Number of Observations Used	3047

The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_duration_min
< 251.340
Node 5
N 159
Avg 19...
surgery_duration_min
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...
surgery_dur...
< 333.450
Node D
N 20
Avg 30...
surgery...
>= 333.4...
Node E
N 2
Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
33	6449015	1.965E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2645	38224.2	10
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

accuracy10	accuracy15	accuracy20
0.507227	0.670171	0.805519

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	CARD_DATA
Random Number Seed	122470
Sampling Rate	0.8
Sample Size	1600
Selection Probability	0.8
Sampling Weight	0
Output Data Set	CARD_DATA

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

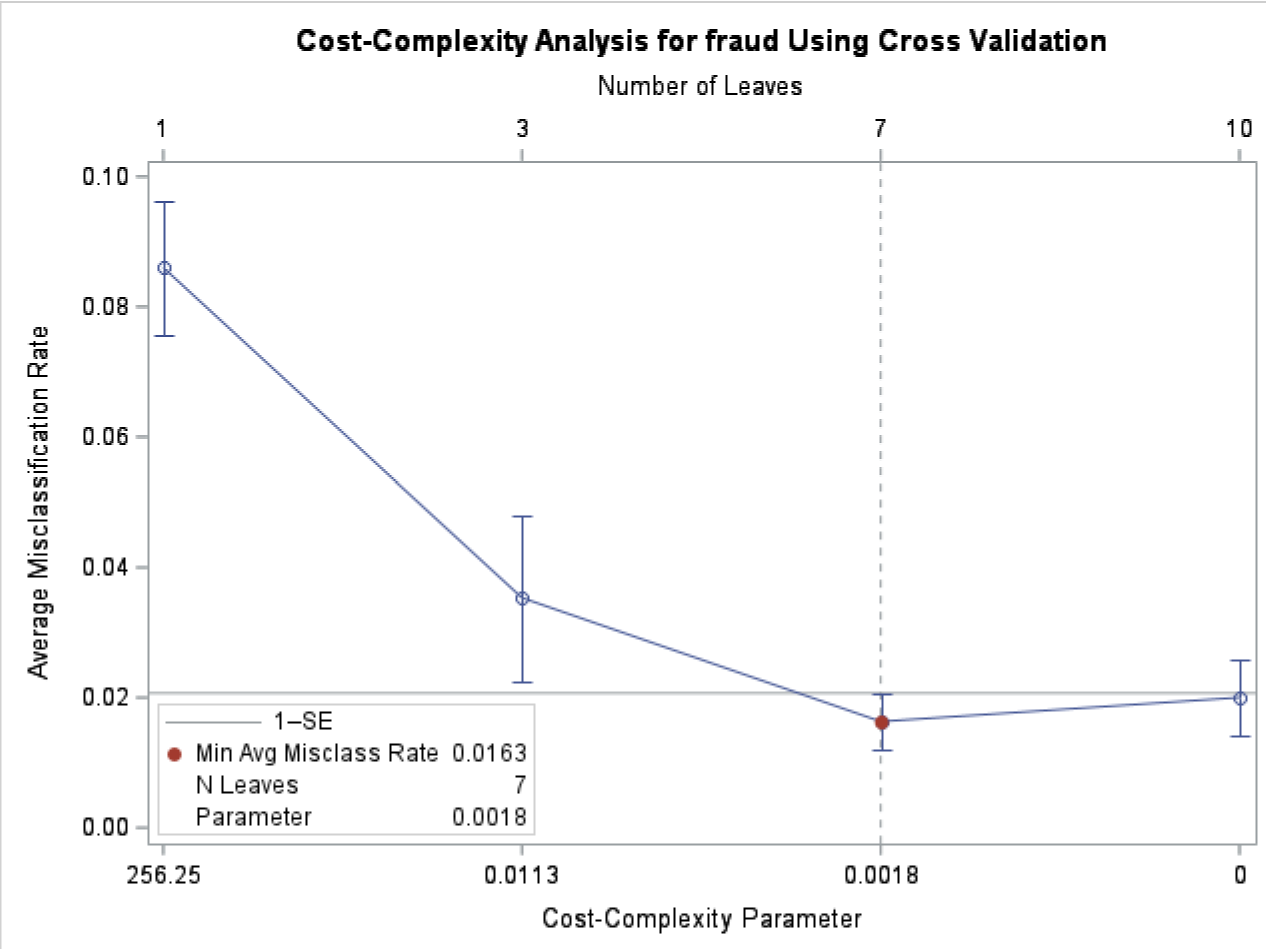
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

Model Information

Split Criterion Used	Gini
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	12
Number of Leaves After Pruning	7
Model Event Level	1

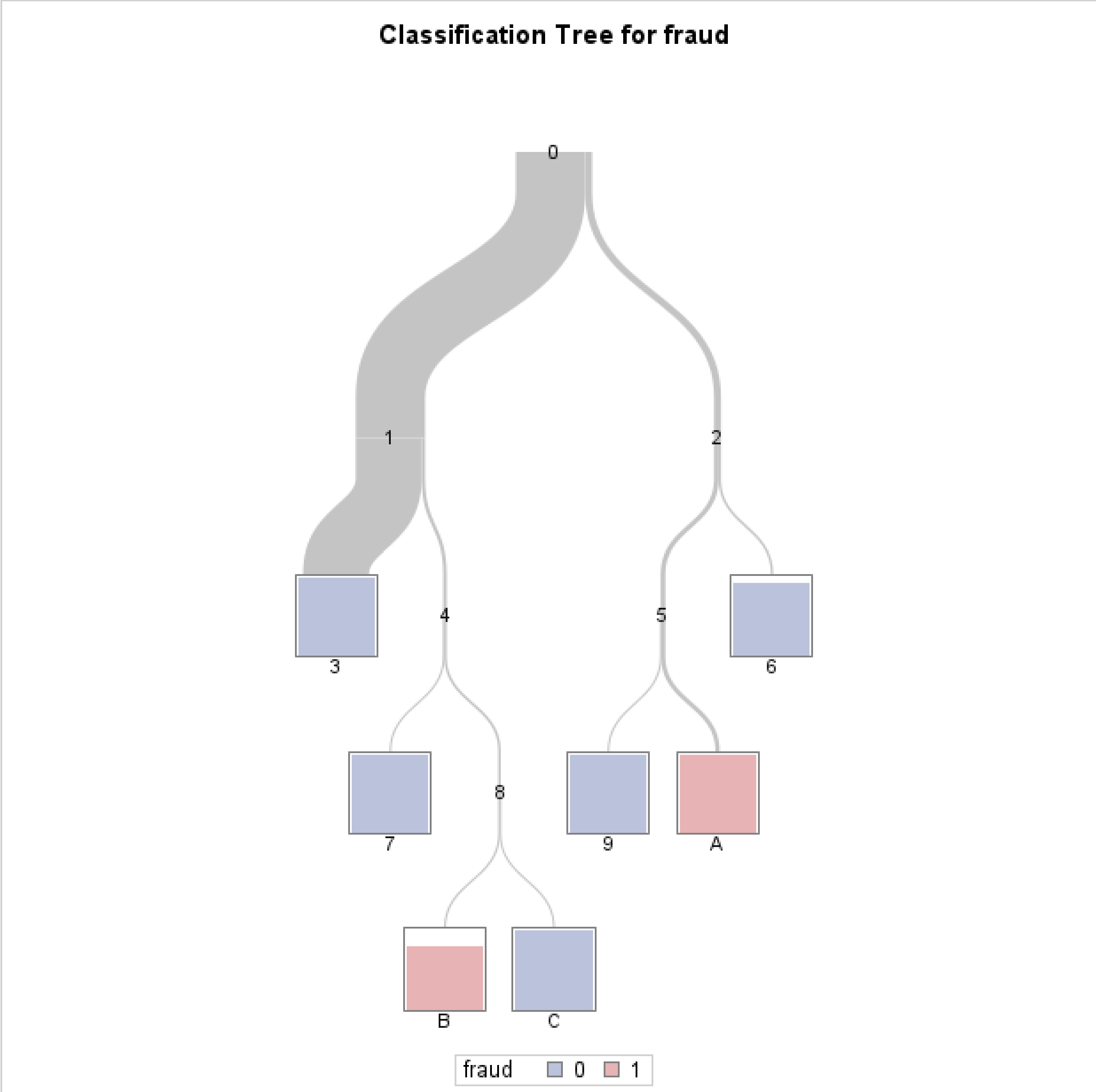
Number of Observations Read	1600
Number of Observations Used	1600

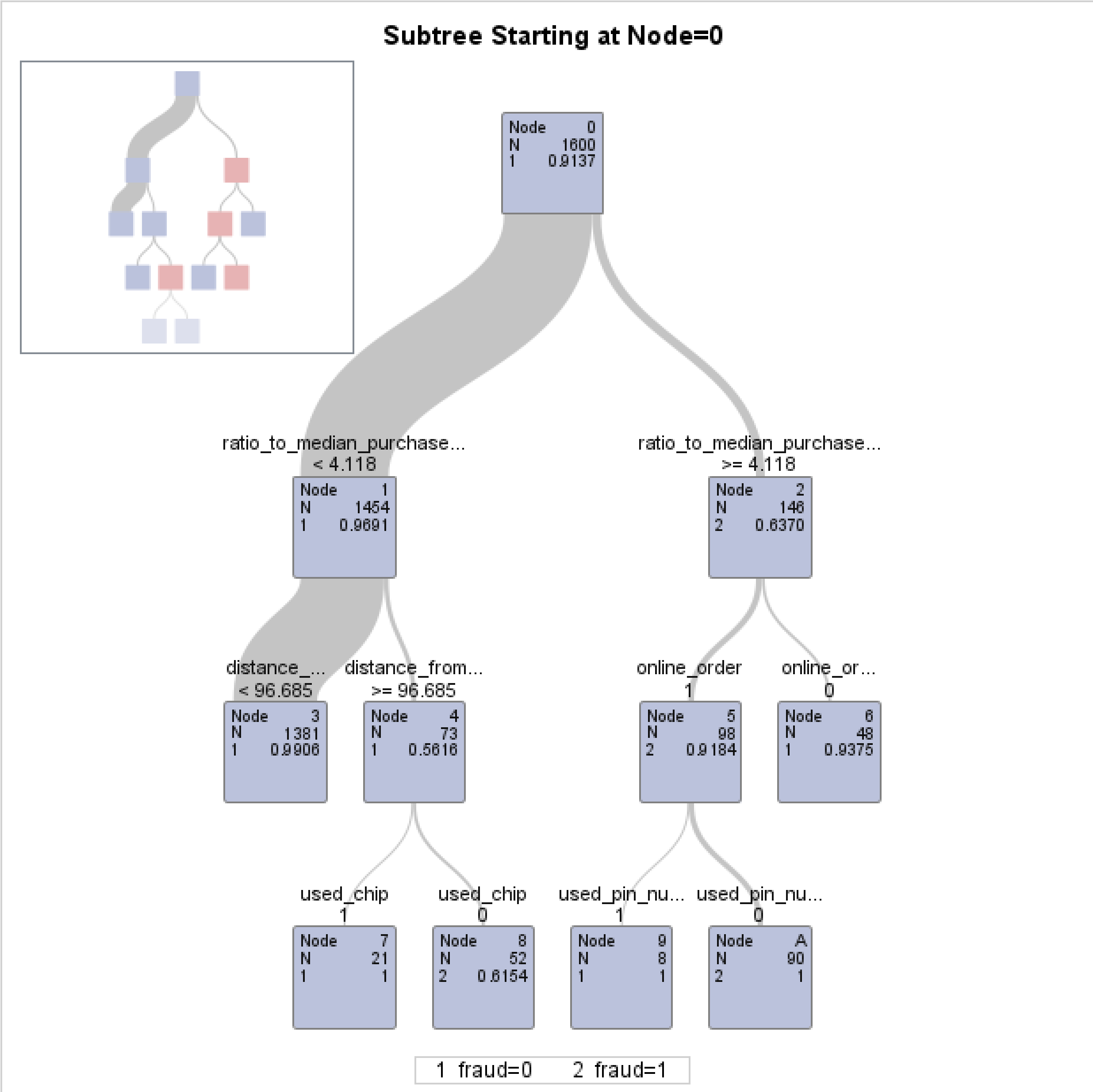
The HPSPLIT Procedure



The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 4.118
Node 1
N 1454
1 0.9691
ratio_to_median_purchase...
>= 4.118
Node 2
N 146
2 0.6370
distance_...
< 96.685
Node 3
N 1381
1 0.9906
distance_from...
>= 96.685
Node 4
N 73
1 0.5616
online_order
1
Node 5
N 98
2 0.9184
online_or...
0
Node 6
N 48
1 0.9375
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 52
2 0.6154
used_pin_nu...
1
Node 9
N 8
1 1
used_pin_nu...
0
Node A
N 90
2 1
1 fraud=0 2 fraud=1

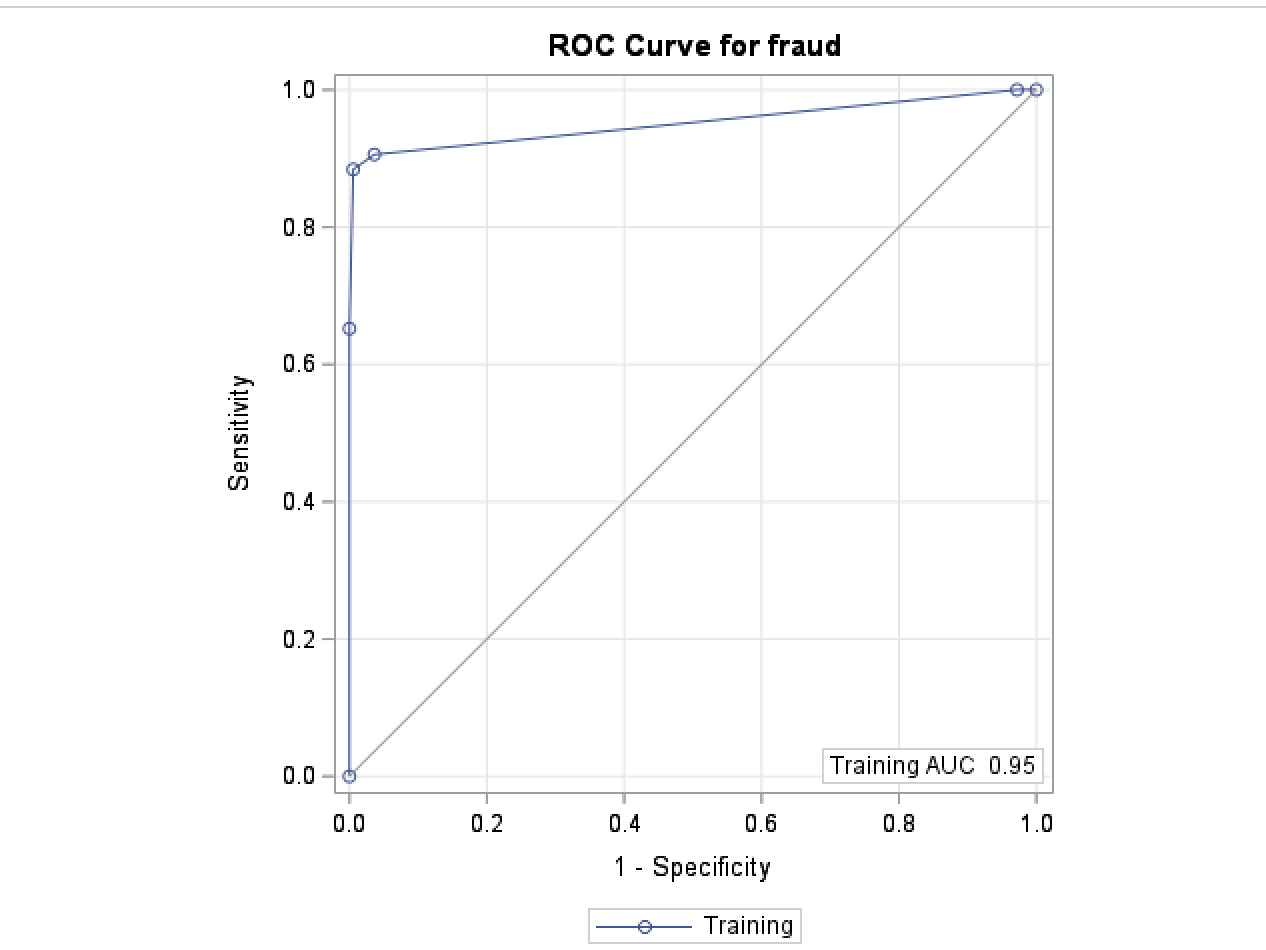
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1454	8	0.0055
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
7	0.0138	0.0150	0.8841	0.9945	0.0945	0.0276	44.1802	0.9514



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.3883	3.8333	1
used_chip	0.3410	3.3660	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	trueclassrate
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

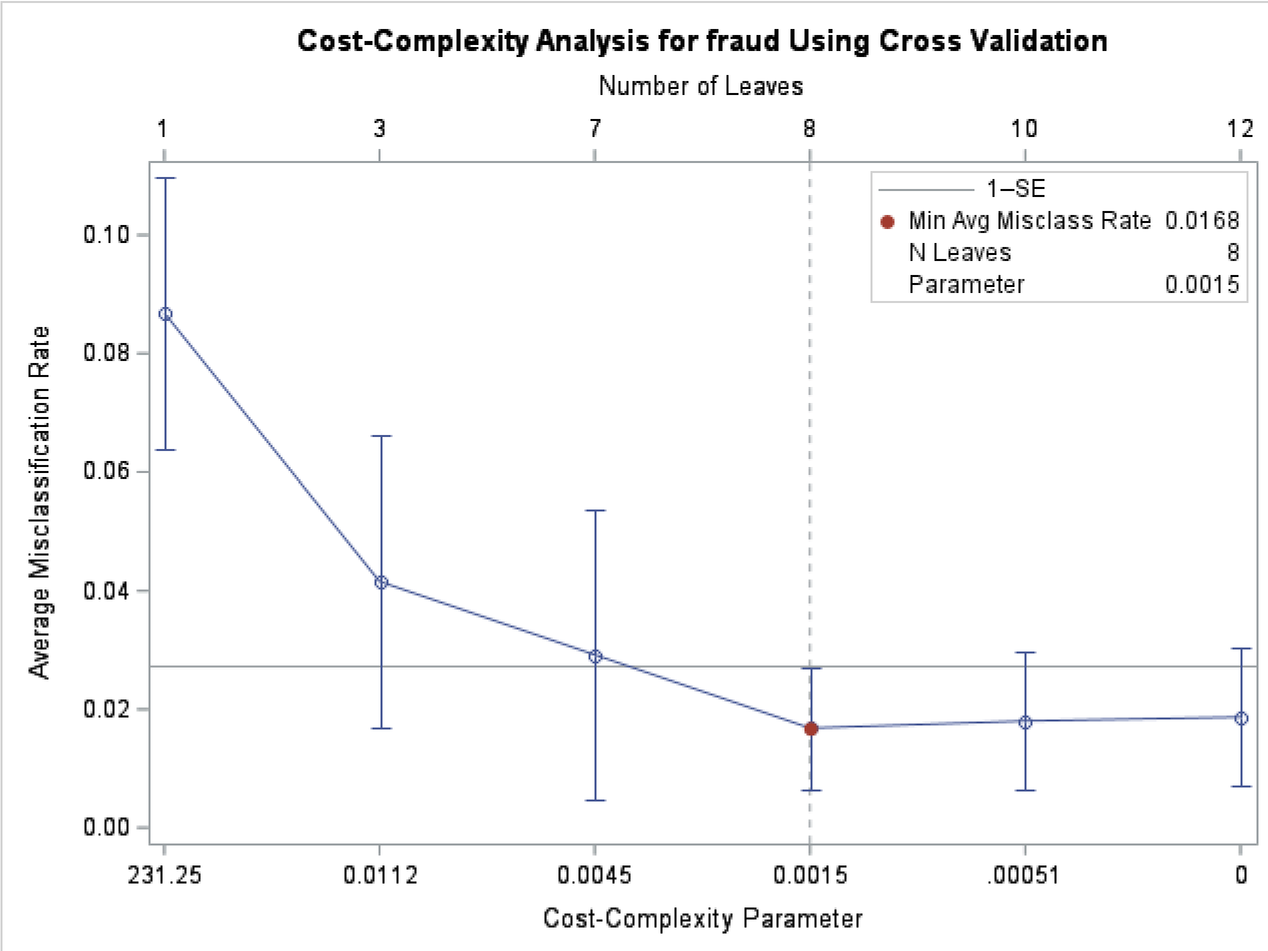
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED2	V9	Output	On Client

Model Information

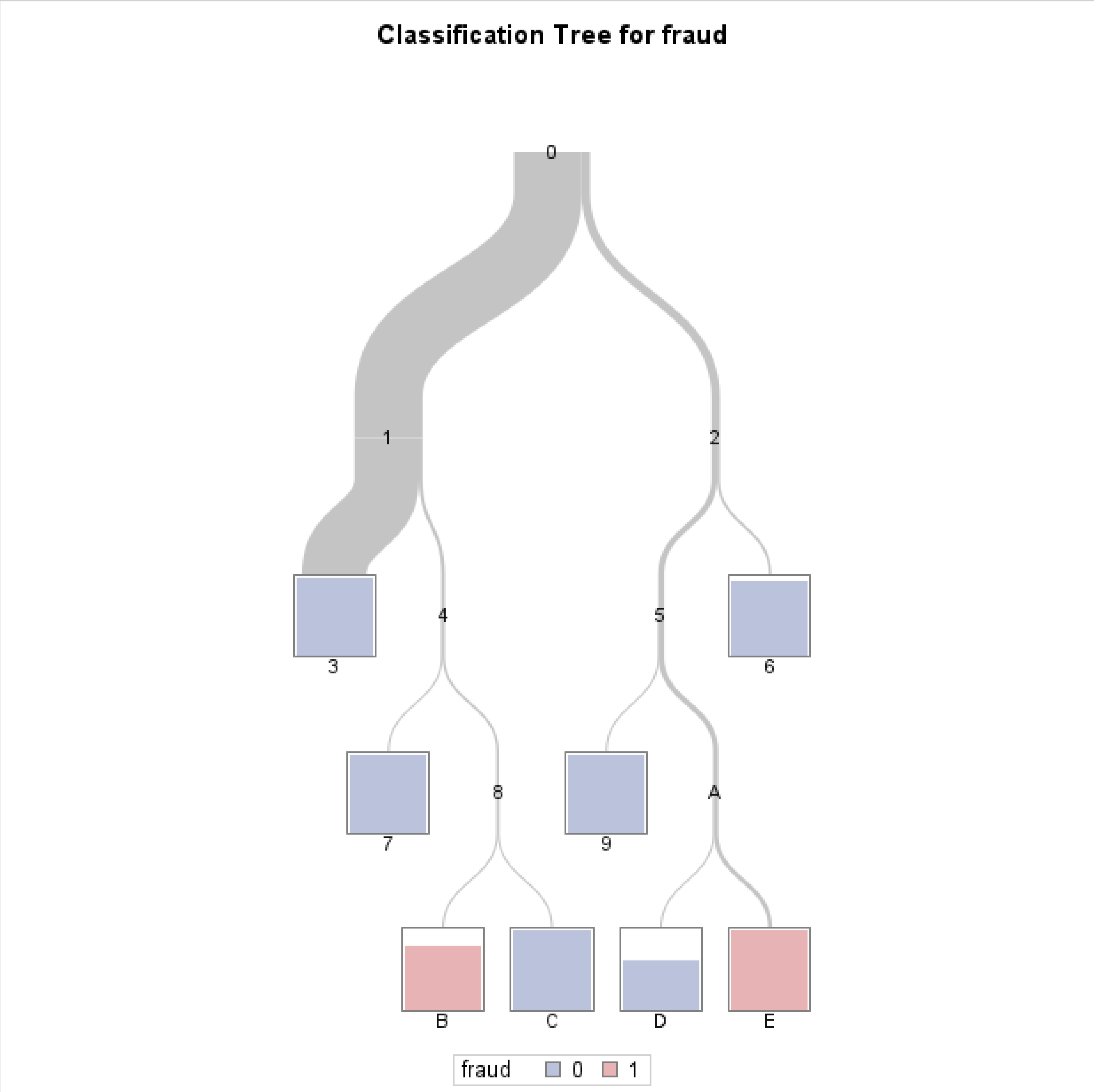
Split Criterion Used	Entropy
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	13
Number of Leaves After Pruning	8
Model Event Level	1

Number of Observations Read	1600
Number of Observations Used	1600

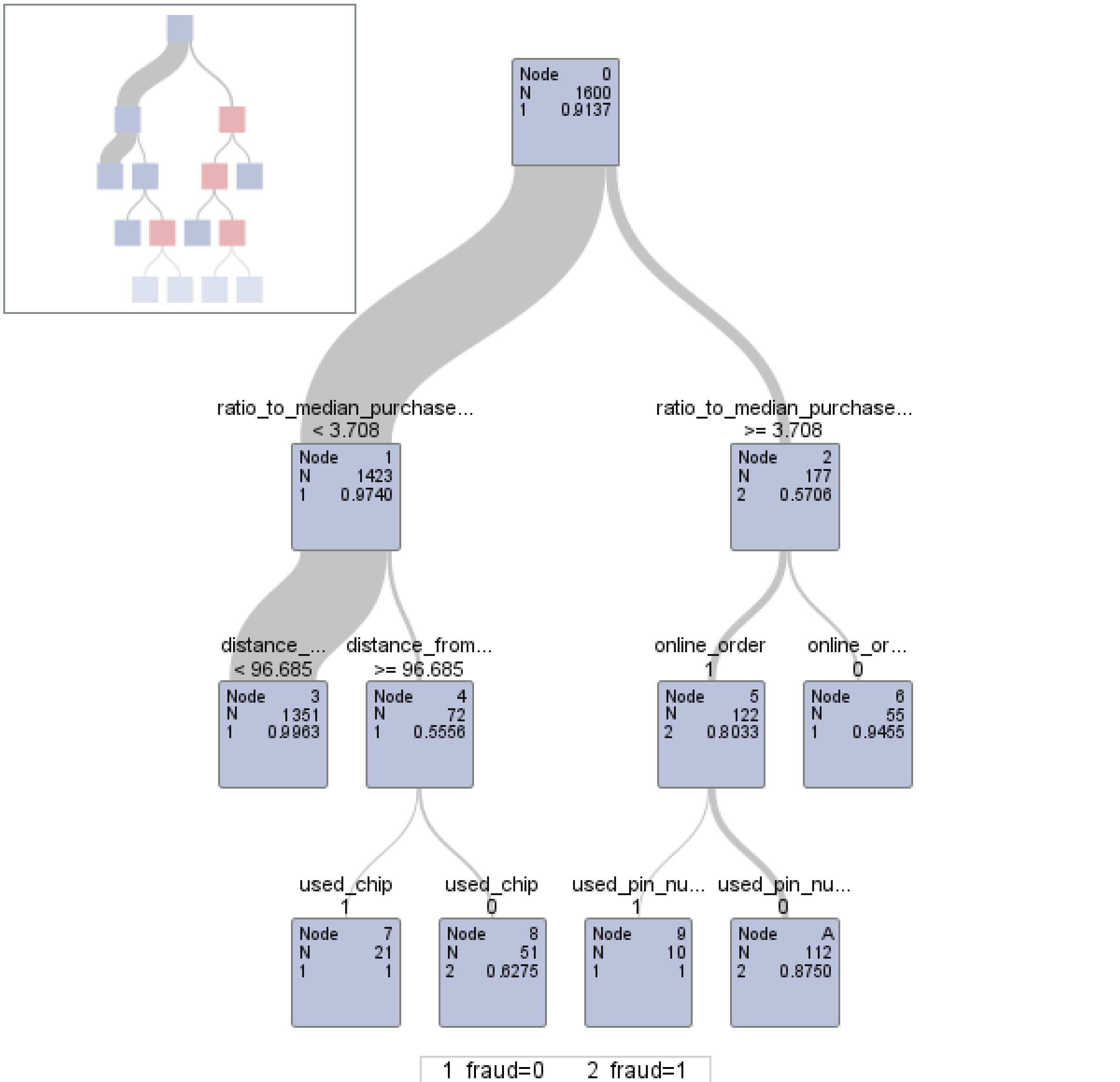
The HPSPLIT Procedure



The HPSPLIT Procedure



Subtree Starting at Node=0



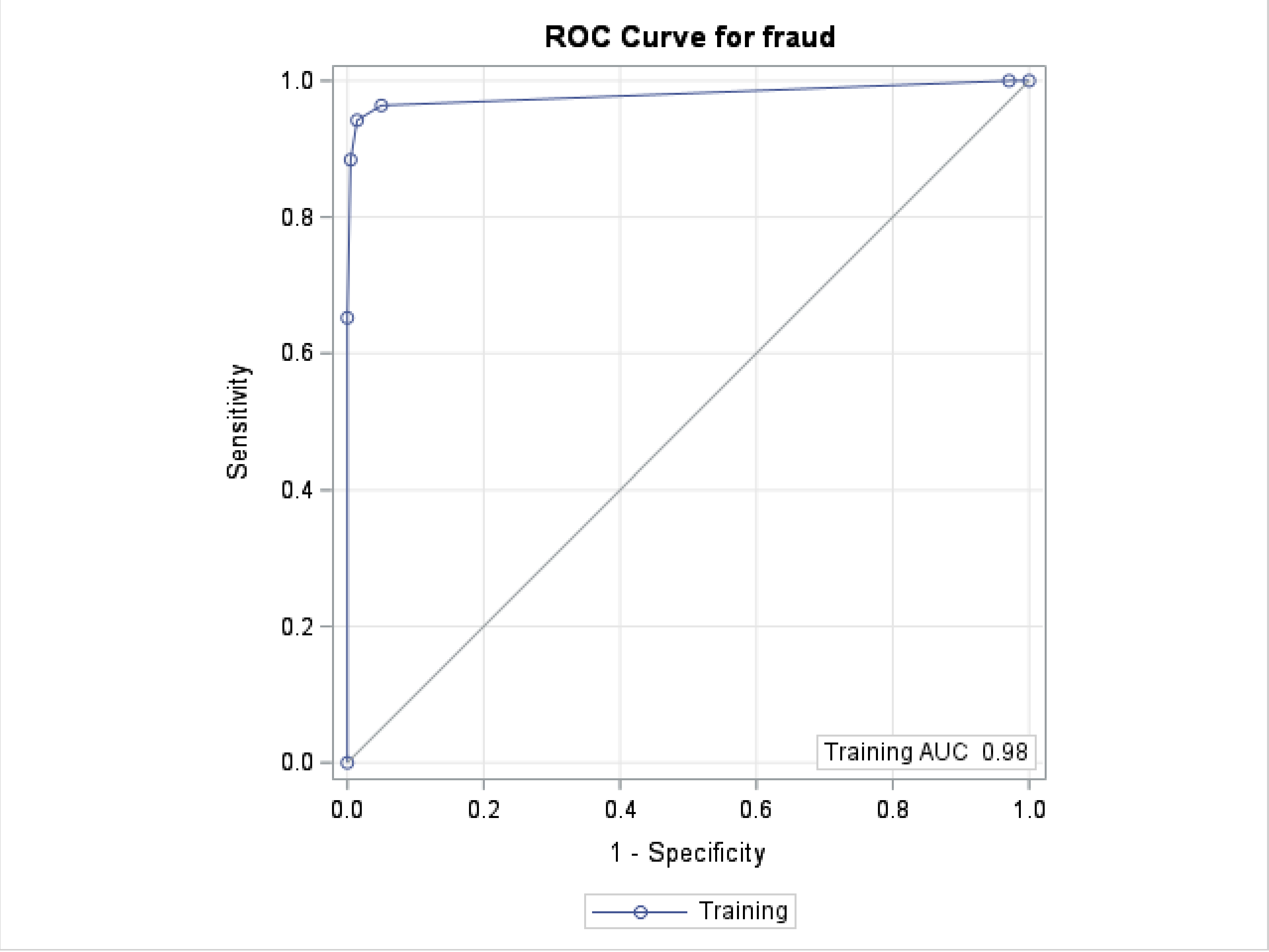
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1455	7	0.0048
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
8	0.0117	0.0144	0.8841	0.9952	0.0698	0.0233	37.3047	0.9797



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	10.3780	2
online_order	0.7137	7.4068	2
distance_from_home	0.4966	5.1534	1
used_pin_number	0.3613	3.7493	1
used_chip	0.3298	3.4223	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	**trueclassrate**
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED3	V9	Output	On Client

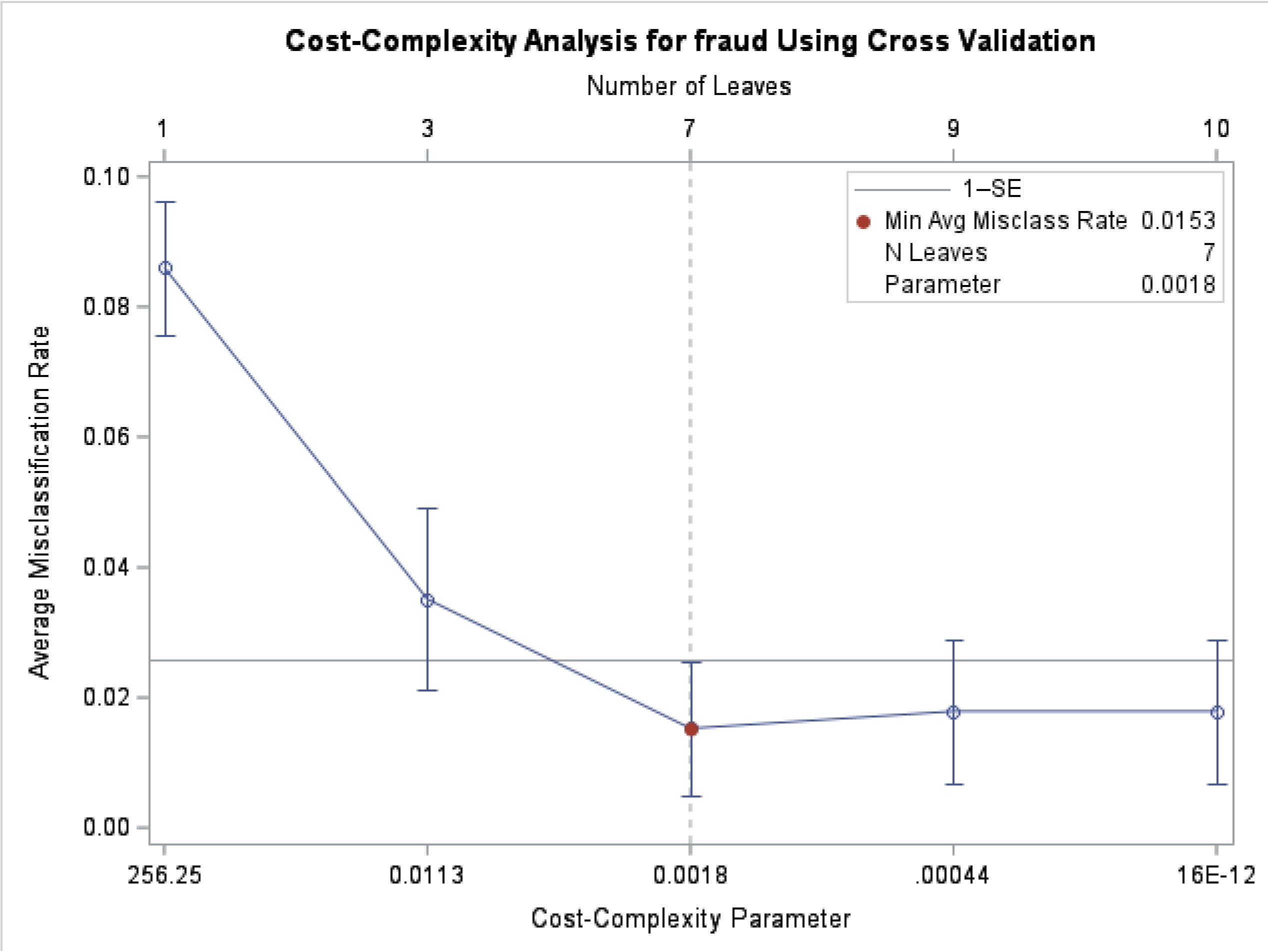
Model Information

Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	12
Number of Leaves After Pruning	7
Model Event Level	1

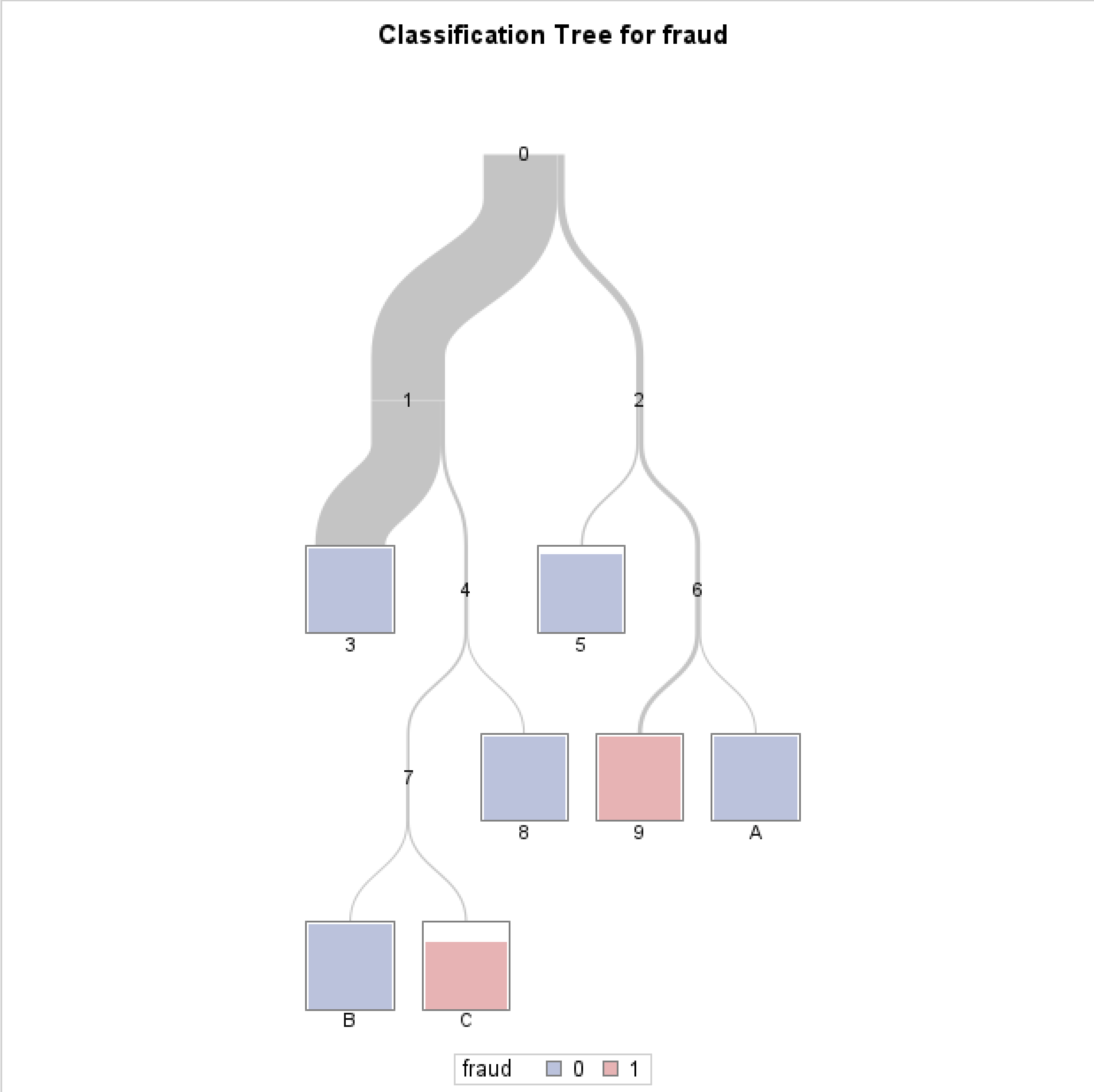
Number of Observations Read	1600
Number of Observations Used	1600

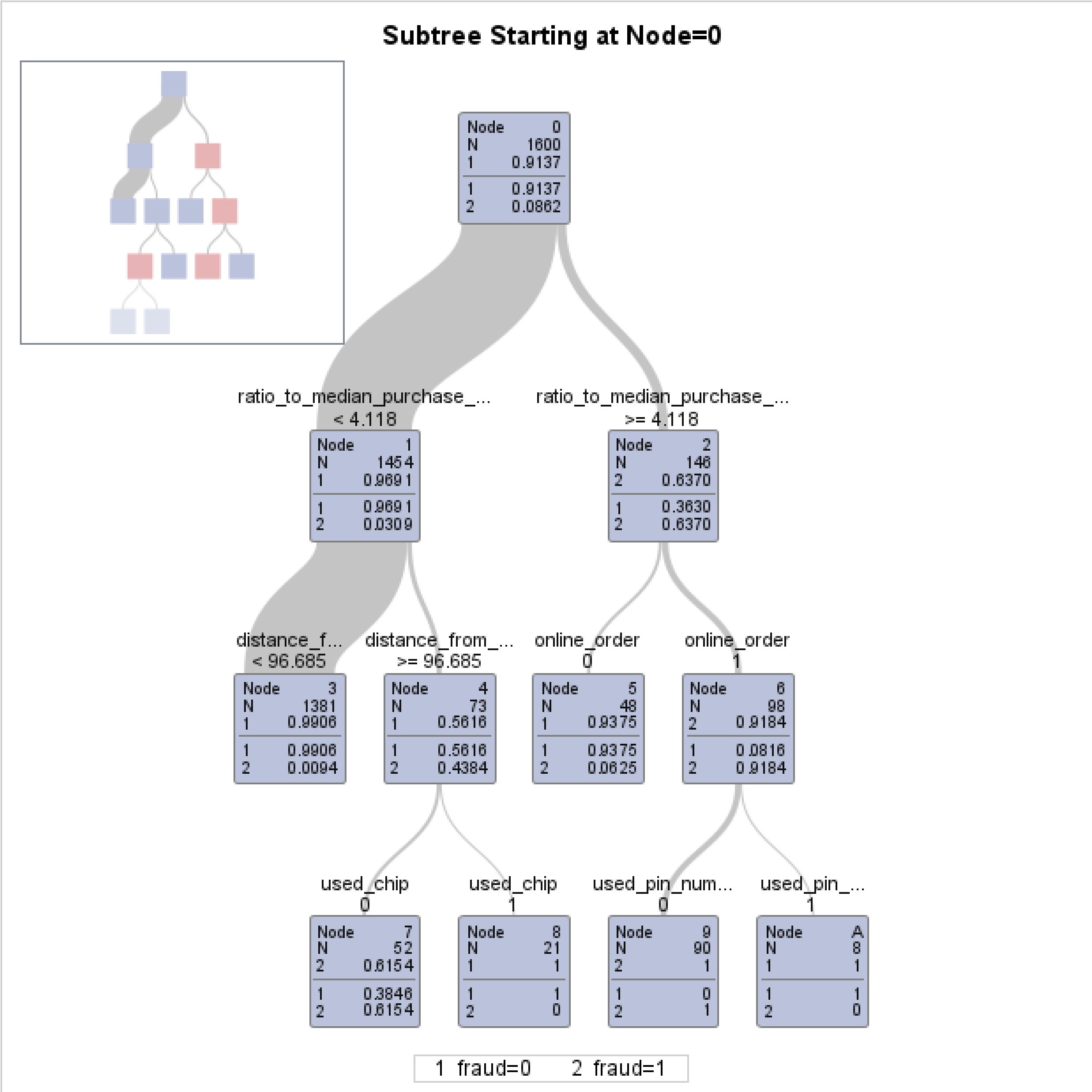
The SAS System

The HPSPLIT Procedure



The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
1 0.9137
2 0.0862
ratio_to_median_purchase_...
< 4.118
ratio_to_median_purchase_...
>= 4.118
Node 1
N 1454
1 0.9691
1 0.9691
2 0.0309
Node 2
N 146
2 0.6370
1 0.3630
2 0.6370
distance_f...
< 96.685
distance_from_...
>= 96.685
online_order
0
online_order
1
Node 3
N 1381
1 0.9906
1 0.9906
2 0.0094
Node 4
N 73
1 0.5616
1 0.5616
2 0.4384
Node 5
N 48
1 0.9375
1 0.9375
2 0.0625
Node 6
N 98
2 0.9184
1 0.0816
2 0.9184
used_chip
0
used_chip
1
used_pin_num...
0
used_pin_...
1
Node 7
N 52
2 0.6154
1 0.3846
2 0.6154
Node 8
N 21
1 1
1 1
2 0
Node 9
N 90
2 1
1 0
2 1
Node A
N 8
1 1
1 1
2 0
1 fraud=0 2 fraud=1

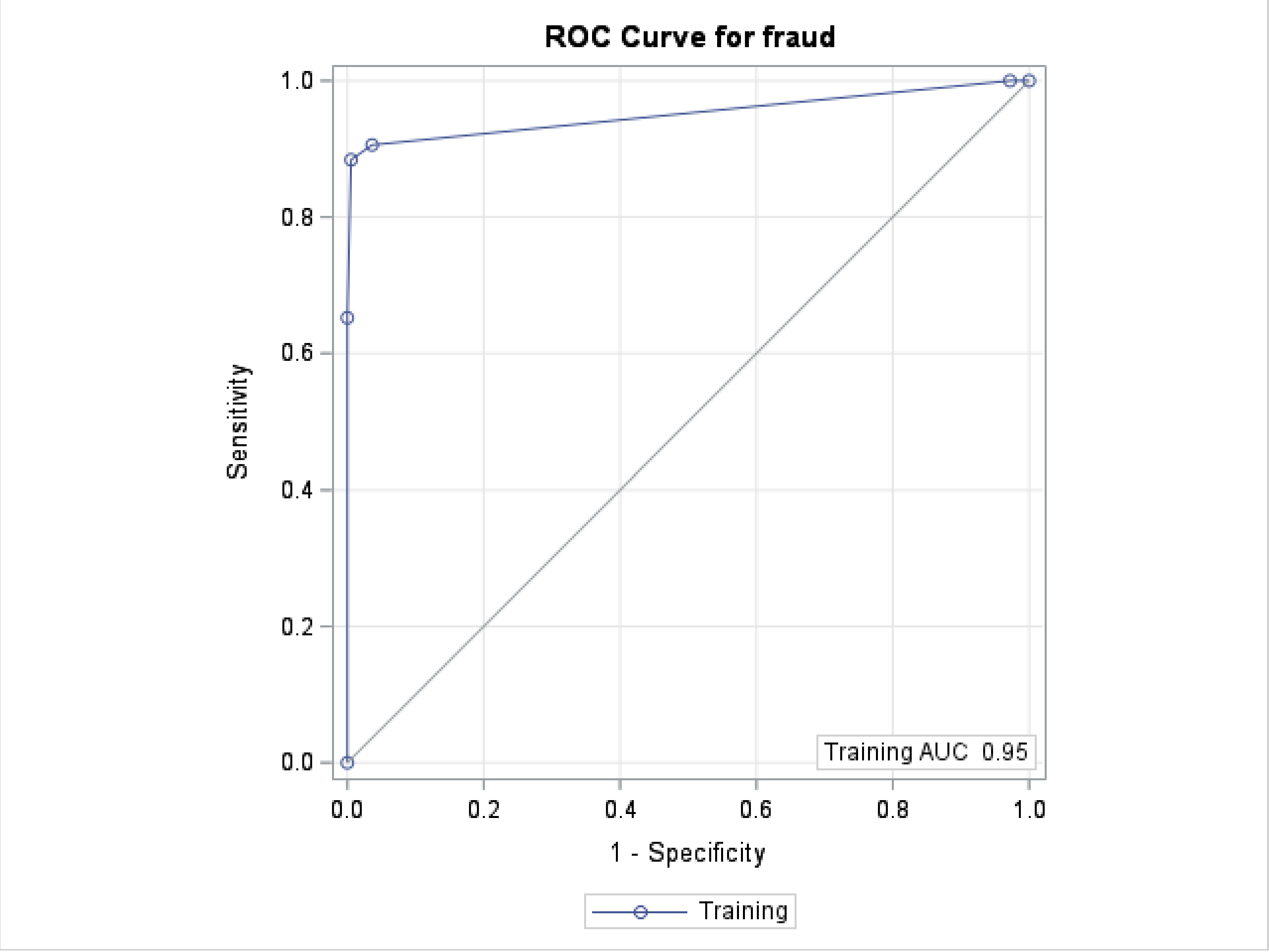
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1454	8	0.0055
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
7	0.0138	0.0150	0.8841	0.9945	0.0945	0.0276	44.1802	0.9514



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.3883	3.8333	1
used_chip	0.3410	3.3660	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	trueclassrate
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	CARD_DATA
Random Number Seed	122470
Sampling Rate	0.8
Sample Size	1600
Selection Probability	0.8
Sampling Weight	0
Output Data Set	CARD_DATA

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

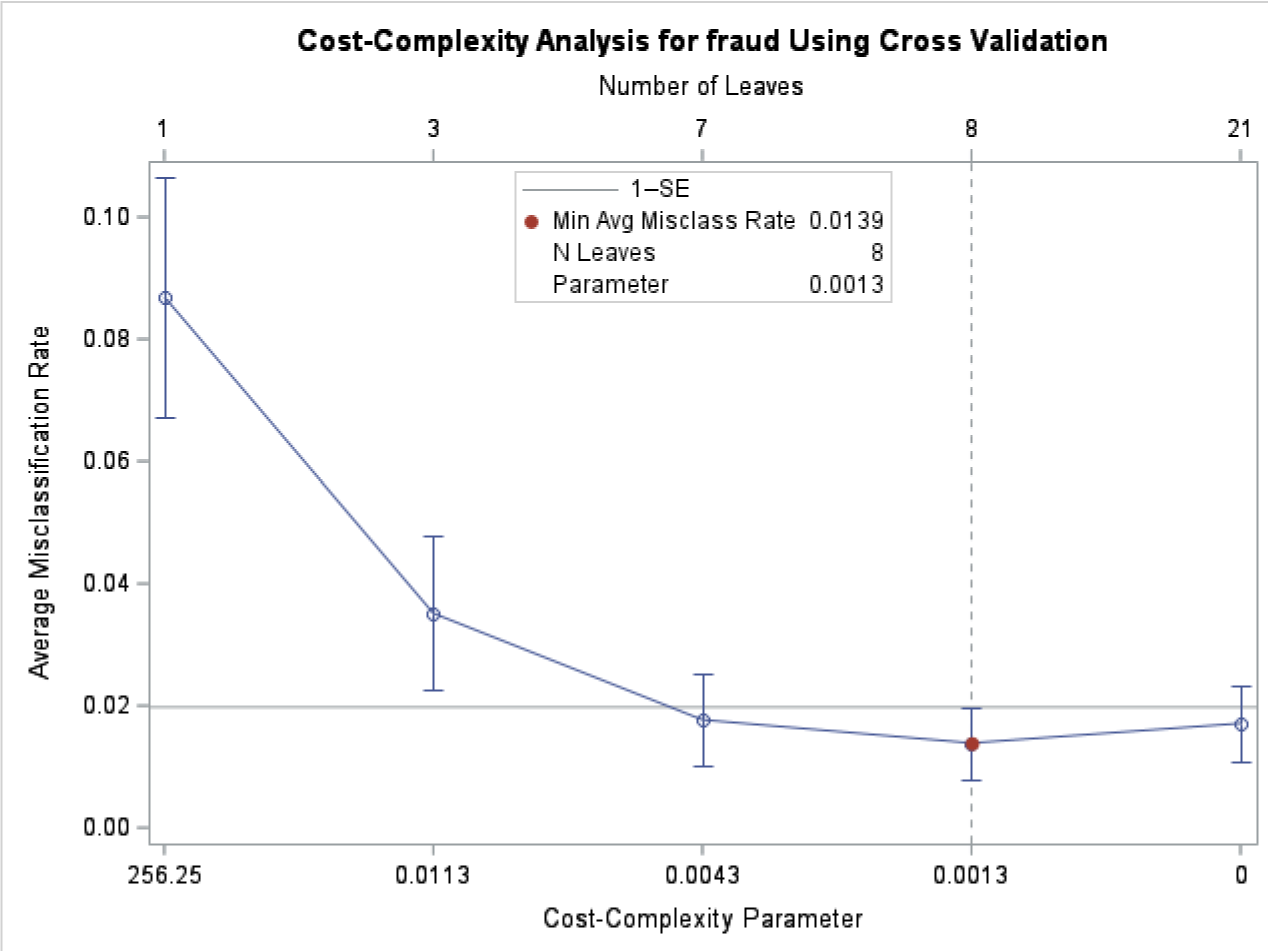
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

Model Information

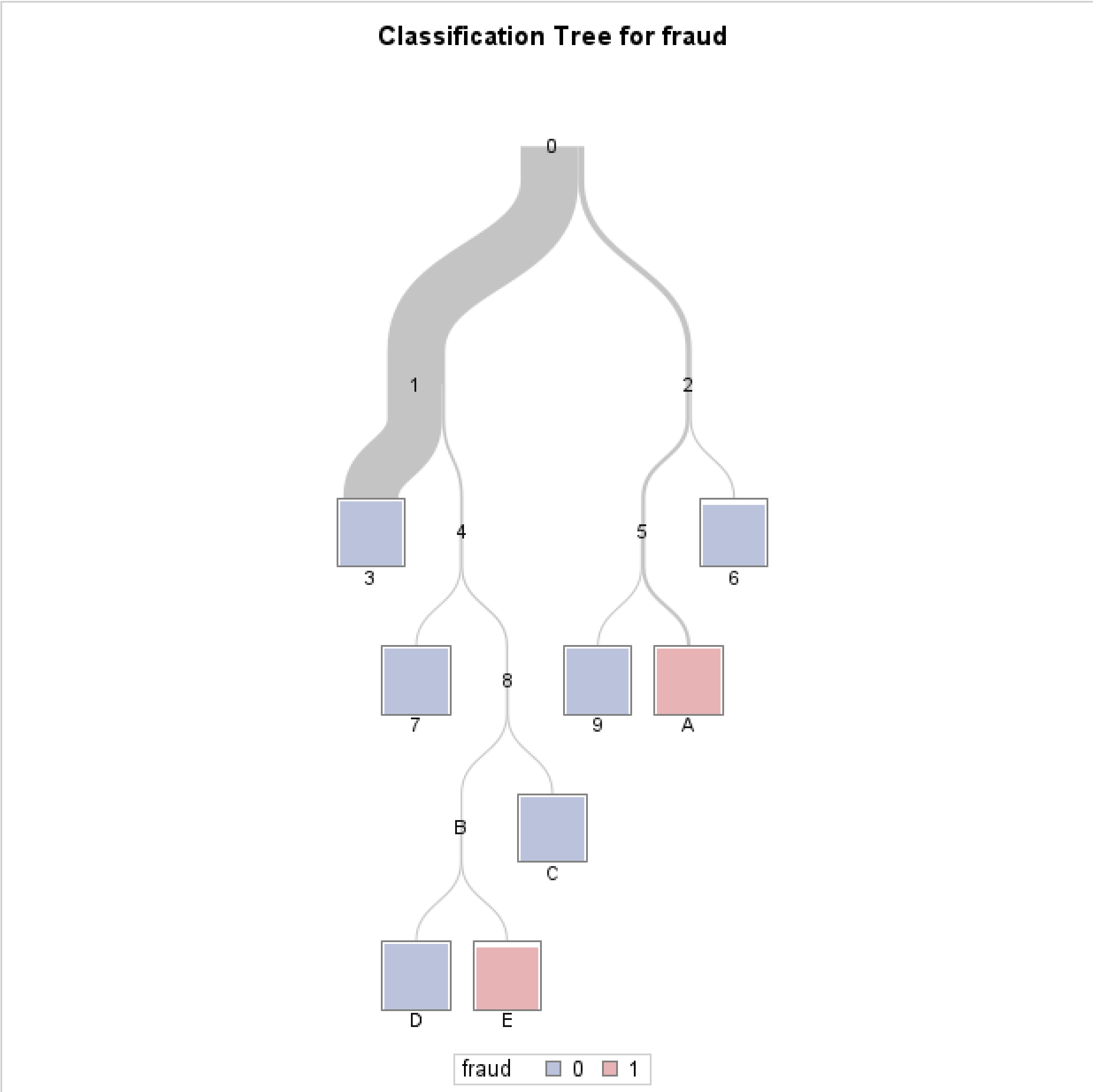
Split Criterion Used	Gini
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	7
Maximum Tree Depth Achieved	7
Tree Depth	5
Number of Leaves Before Pruning	26
Number of Leaves After Pruning	8
Model Event Level	1

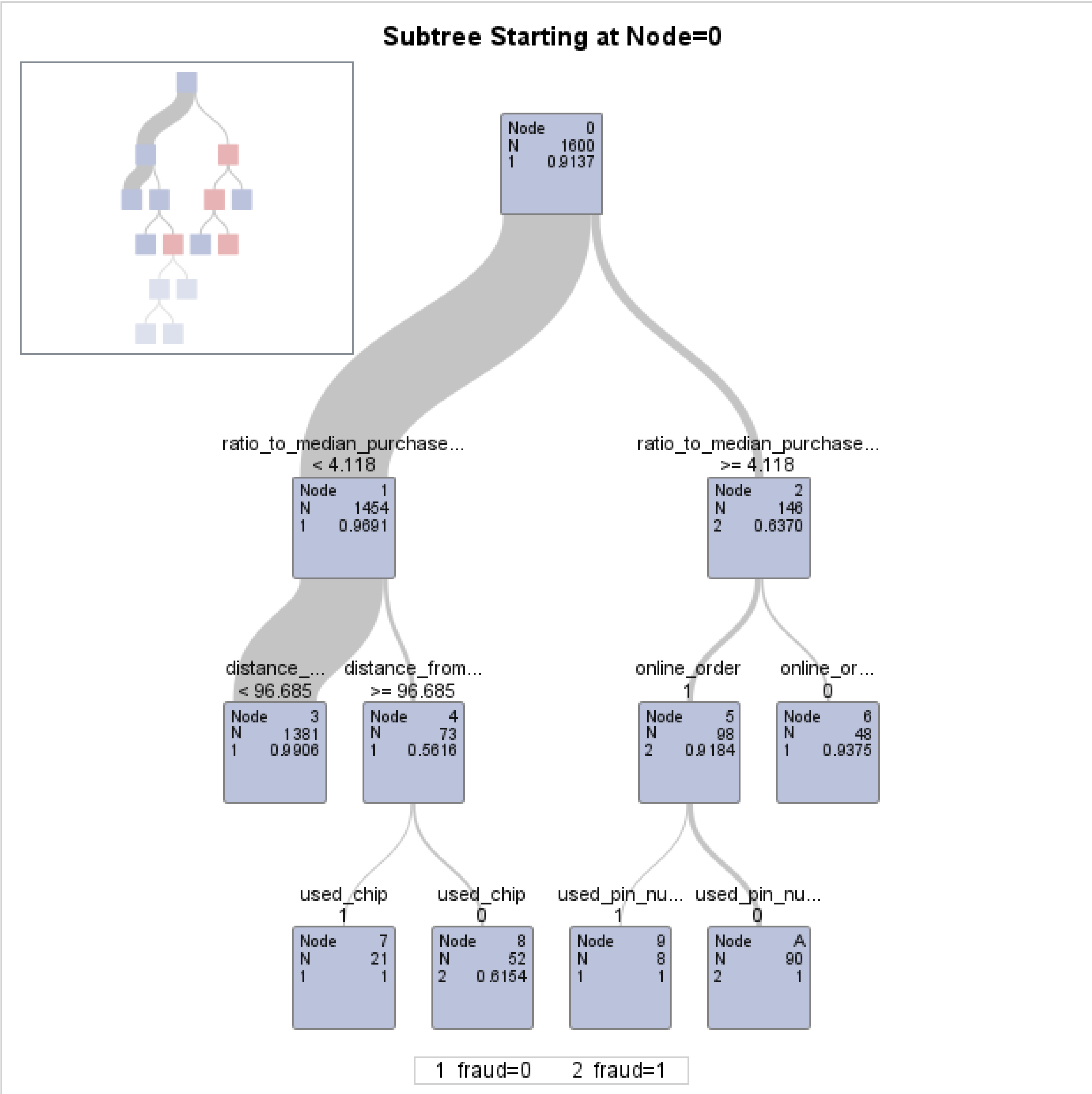
Number of Observations Read	1600
Number of Observations Used	1600

The HPSPLIT Procedure



The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 4.118
Node 1
N 1454
1 0.9691
ratio_to_median_purchase...
>= 4.118
Node 2
N 146
2 0.6370
distance_...
< 96.685
Node 3
N 1381
1 0.9906
distance_from...
>= 96.685
Node 4
N 73
1 0.5616
online_order
1
Node 5
N 98
2 0.9184
online_or...
0
Node 6
N 48
1 0.9375
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 52
2 0.6154
used_pin_nu...
1
Node 9
N 8
1 1
used_pin_nu...
0
Node A
N 90
2 1
1 fraud=0 2 fraud=1

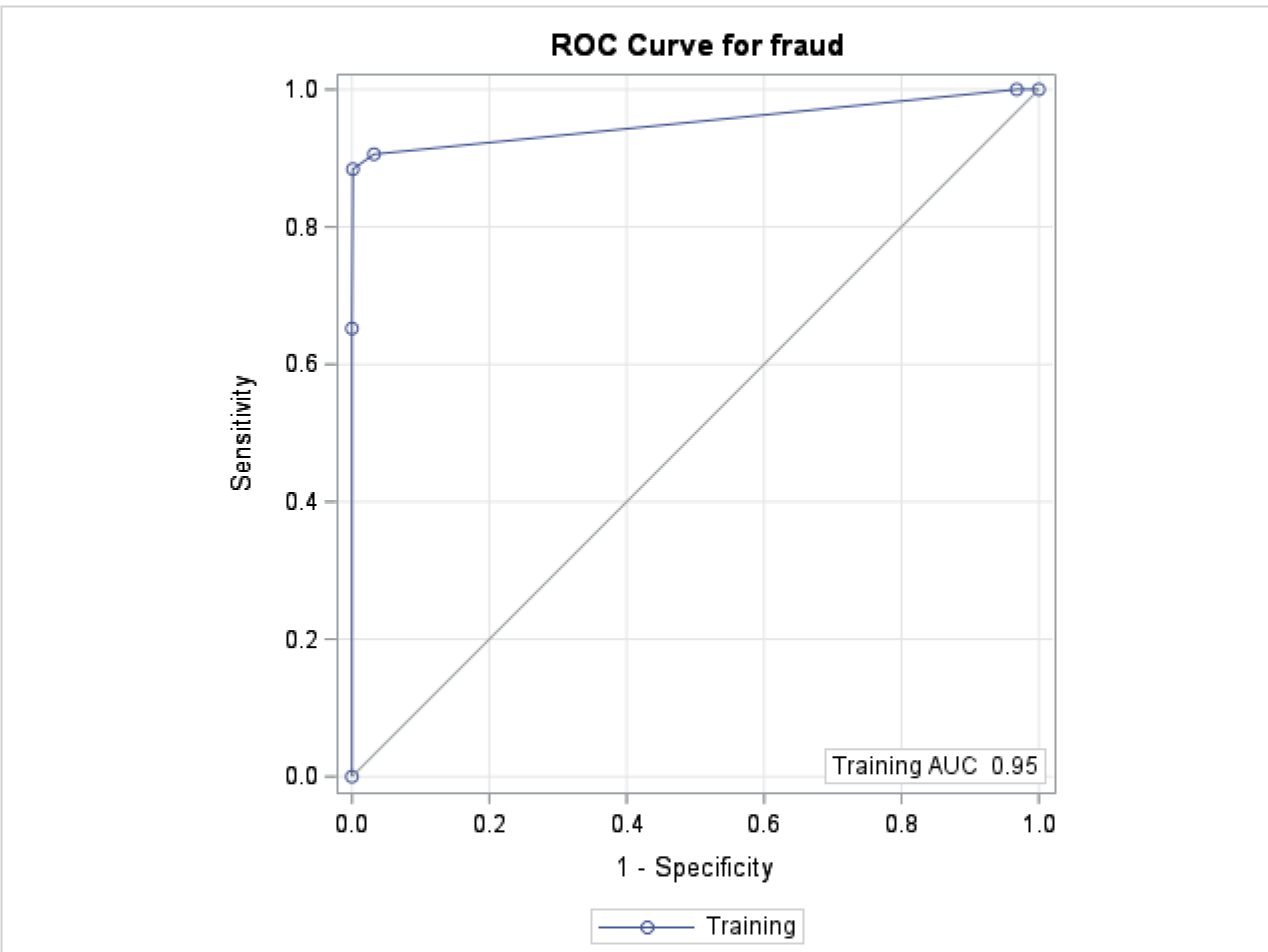
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1460	2	0.0014
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
8	0.0110	0.0113	0.8841	0.9986	0.0833	0.0220	35.1450	0.9524



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.4934	4.8713	2
used_chip	0.3410	3.3660	1

tp	fp	tn	fn	total
32	0	361	7	400

The SAS System

accuracy	misclassrate	sensitivity	FNR	specificity	FPR	precision	NPV	F1score
0.9825	0.0175	0.820513	0.179487	1	0	1	0.980978	0.901408

R Code

```
library(readr)
library(rpart)
library(rpart.plot)
library(dplyr)
library(partykit)
library(CHAID)

card_data =
read.csv("C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/card_transdata.csv
",
header=T, sep=",")

# Splitting data into 80% training and 20% testing sets.

set.seed(122470)
sample = sample(c(T,F), nrow(card_data),
replace=T, prob=c(0.8, 0.2))
train = card_data[sample,]
test = card_data[!sample,]

# Fitting pruned binary tree with Gini Splitting Criterion.

tree_gini = rpart(fraud~distance_from_home+distance_from_last_transaction
+ratio_to_median_purchase_price+repeat_retailer+used_chip+used_pin_number
+online_order, data=train, method="class", parms=list(split="Gini"),
maxdepth=7)

rpart.plot(tree_gini, type=3)

pred_values = predict(tree_gini, test)
test = cbind(test, pred_values)

tp = c()
```

```r
fp = c()
tn = c()
fn = c()

total = nrow(test)
for (i in 1:total) {
    tp[i] = ifelse(test$"1"[i]>0.5 & test$fraud[i]==1,1,0)
    fp[i] = ifelse(test$"1"[i]>0.5 & test$fraud[i]==0,1,0)
    tn[i] = ifelse(test$"1"[i]>0.5 & test$fraud[i]==0,1,0)
    fn[i] = ifelse(test$"1"[i]>0.5 & test$fraud[i]==1,1,0)
}

print(tp <- sum(tp))
print(fp <- sum(fp))
print(tn <- sum(tn))
print(fn <- sum(fn))
print(total)

print(accuracy <- (tp+tn)/total)
print(misclassrate <- (fp+fn)/total)
print(sensitivity <- tp/(tp+fn))
print(FNR <- fn/(tp+fn))
print(specificity <- tn/(fp+tn))
print(FPR <- fp/(fp+tn))
print(precision <- tp/(fp+fp))
print(NPV <- tn/(fn+tn))
print(F1score <- 2*tp/(2*tp+fn+fp))
```

```
> print(tp <- sum(tp))
[1] 35
> print(fp <- sum(fp))
[1] 1
> print(tn <- sum(tn))
[1] 1
> print(fn <- sum(fn))
[1] 35
> print(total)
[1] 402
>
> print(accuracy <- (tp+tn)/total)
[1] 0.08955224
> print(misclassrate <- (fp+fn)/total)
[1] 0.08955224
> print(sensitivity <- tp/(tp+fn))
[1] 0.5
> print(FNR <- fn/(tp+fn))
[1] 0.5
> print(specificity <- tn/(fp+tn))
[1] 0.5
> print(FPR <- fp/(fp+tn))
[1] 0.5
> print(precision <- tp/(fp+fp))
```

```
[1] 17.5
> print(NPV <- tn/(fn+tn))
[1] 0.02777778
> print(F1score <- 2*tp/(2*tp+fn+fp))
[1] 0.6603774
```

Python Code

```
# STAT 574 HW1 Problem 3

# Import necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.model_selection import train_test_split
# Importing the data

card_path = "C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/\
card_transdata.csv"
card_data = pd.read_csv(card_path)

X = card_data.iloc[:,0:7].values
y = card_data.iloc[:,7].values

# Splitting the data into 80% training and 20% testing sets

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,
                                                    random_state=122470)

# Fitting binary tree with Gini splitting criterion.

gini_tree = DecisionTreeClassifier(max_leaf_nodes=7, criterion="gini",
                                   random_state=380381)
gini_tree_fit = gini_tree.fit(X_train, y_train)

# (a) Computing confusion matrix for fitted Gini tree

y_pred = gini_tree_fit.predict_proba(X_test)

total = len(y_pred)
tpos = []
fpos = []
```

```python
tneg = []
fneg = []

for sub1, sub2 in zip(y_pred[::,1], y_test):
    tpos.append(1) if (sub1>0.5 and sub2==1) else tpos.append(0)
    fpos.append(1) if (sub1>0.5 and sub2==0) else fpos.append(0)
    tneg.append(1) if (sub1<0.5 and sub2==0) else tneg.append(0)
    fneg.append(1) if (sub1<0.5 and sub2==1) else fneg.append(0)
    tp = sum(tpos)
    fp = sum(fpos)
    tn = sum(tneg)
    fn = sum(fneg)

print('tp:', tp)
print('fp:',  fp)
print('tn:', tn)
print('fn:', fn)
# (b) Computing the predictive performance measures: accuracy, sensitivity,
# misclassification rate, specificity, False negative rate, false positive rate,
# precision, negative predictive value, and F1 score.

accuracy = (tp+tn)/total
misclassrate = (fp+fn)/total
sensitivity = tp/(tp+fn)
FNR = fn/(tp+fn)
specificity = tn/(fp+tn)
FPR = fp/(fp+tn)
precision = tp/(tp+fp)
NPV = tn/(fn+tn)
F1score = 2*tp/(2*tp+fn+fp)

print("accuracy:", accuracy)
print("misclassification rate:", misclassrate)
print("sensitivity:", sensitivity)
print("False Negative Rate:", FNR)
print("Specificity:", specificity)
print("False Positive Rate:", FPR)
print("Precision:", precision)
print("Negative Predictive Value:", NPV)
print("F1 score:", F1score)
```

tp: 40

fp: 1

tn: 356

fn: 3

accuracy: 0.99

misclassification rate: 0.01

sensitivity: 0.9302325581395349

False Negative Rate: 0.06976744186046512

Specificity: 0.9971988795518207

False Positive Rate: 0.0028011204481792717

Precision: 0.975609756097561

Negative Predictive Value: 0.9916434540389972

F1 score: 0.9523809523809523

Problem 4.

SAS Code

```
proc import out=card_data
datafile="C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/card_transdata.csv
"
dbms=csv replace;

/* Splitting the data into 80% training and 20% testing sets*/

proc surveyselect data=card_data rate=0.8 seed=122470
out=card_data outall method=srs;
run;

/*Gini-splitting and cost-complexity pruning*/

proc hpsplit data=card_data maxdepth=7;
    class repeat_retailer used_chip used_pin_number online_order fraud;
    model fraud(event="1")=distance_from_home distance_from_last_transaction
ratio_to_median_purchase_price repeat_retailer used_chip used_pin_number
online_order;
    grow gini;
    prune costcomplexity;
    partition rolevar=selected(train="1");
```

```
    output out=predicted;
    ID selected;
run;

/* (a) COMPUTING CONFUSION MATRICES AND PERFORMANCE MEASURES
FOR TESTING SET FOR A RANGE OF CUTOFFS*/
data test;
set predicted;
if(selected="0");
run;

data cutoffs;
set test;
do i=0 to 101;
tp=(P_fraud1 >= 0.01*i and fraud="1");
fp=(P_fraud1 >= 0.01*i and fraud="0");
tn=(P_fraud1 <  0.01*i and fraud="0");
fn=(P_fraud1 <  0.01*i and fraud="1");
output;
end;
run;

proc sql;
create table confusion as
select i, sum(tp) as tp, sum(fp) as fp, sum(tn) as tn,
sum(fn) as fn, count(*) as total
from cutoffs
group by i;
quit;

proc sql;
create table measures as
select i, (tp+tn)/total as accuracy, (fp+fn)/total as
misclassrate, tp/(tp+fn) as sensitivity, tn/(fp+tn) as specificity,
fp/(fp+tn) as oneminusspec
from confusion
group by i;
quit;


/*  (b) PLOTTING ROC CURVE*/
title 'The Receiver Operating Characteristic Curve';
proc gplot data=measures;
symbol v=square interpol=join;
plot sensitivity*oneminusspec/ vaxis=0 to 1 by 0.1 haxis=0 to 1 by 0.1;
```

```
label sensitivity="Sensitivity" oneminusspec="1-Specificity";
run;

/* (c) REPORTING MEASURES FOR THE POINT ON ROC CURVE CLOSEST
TO THE IDEAL POINT (0,1)*/
proc sql;
select accuracy, misclassrate, sensitivity, specificity,
sqrt(oneminusspec**2+(1-sensitivity)**2) as distance, i*0.01 as cutoff
from measures
having distance=min(distance);
quit;

/* (d) COMPUTING AREA UNDER THE ROC CURVE*/
proc sort data=measures;
by oneminusspec;
run;

data AUC;
set measures;
lagx=lag(oneminusspec);
lagy=lag(sensitivity);
if lagx=. then lagx=0;
if lagy=. then lagy=0;
trapezoid=(oneminusspec-lagx)*(sensitivity+lagy)/2;
AUC+trapezoid;
run;

proc print data=AUC (firstobs=102) noobs;
var AUC;
run;
```

The SAS System

The SURVEYSELECT Procedure

Selection Method	Simple Random Sampling

Input Data Set	HOSPITAL
Random Number Seed	479576
Sampling Rate	0.8
Sample Size	3047
Selection Probability	0.800158

Sampling Weight	0
Output Data Set	HOSPITAL

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

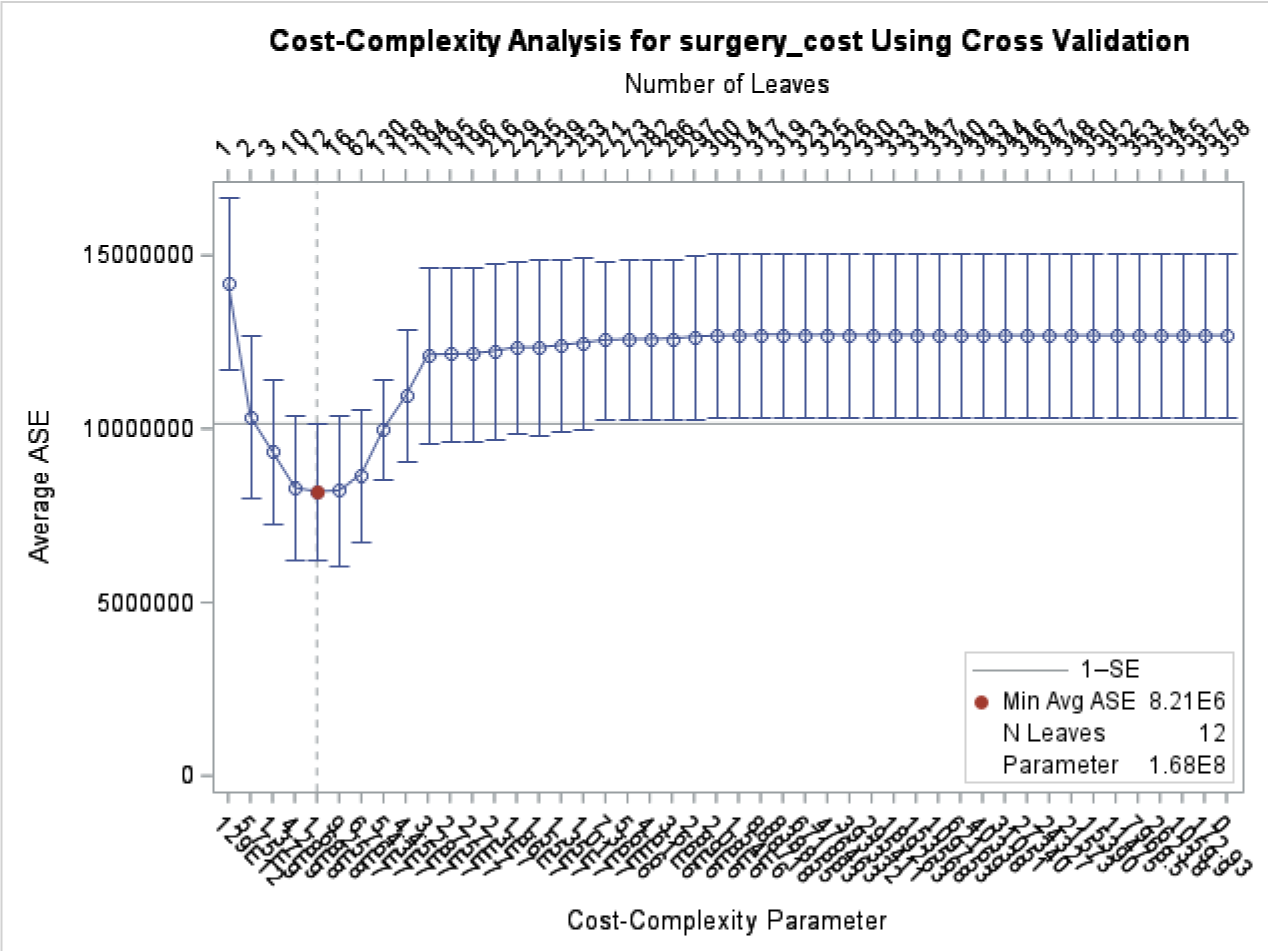
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	13

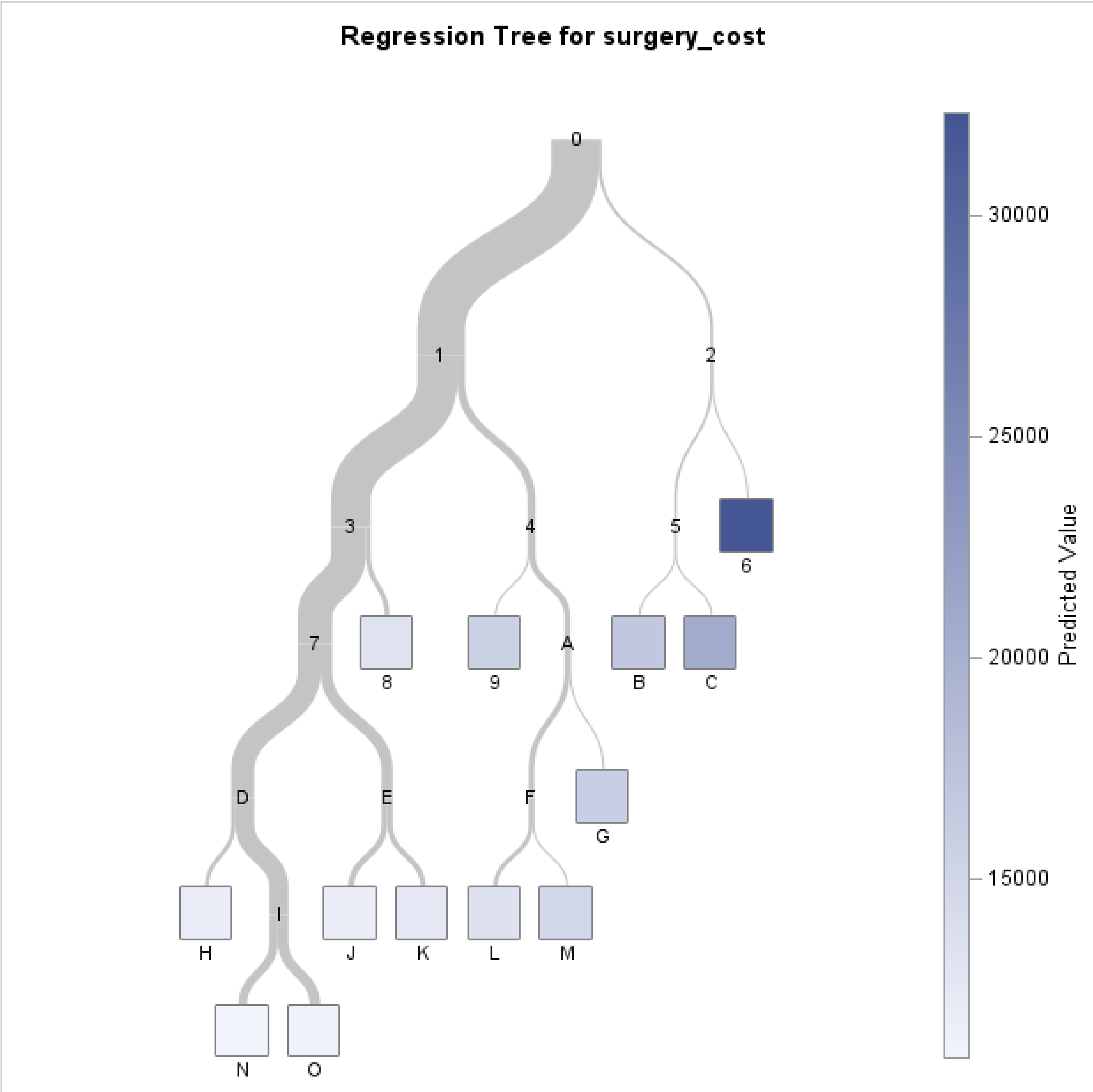
Number of Observations Read	3047
Number of Observations Used	3047

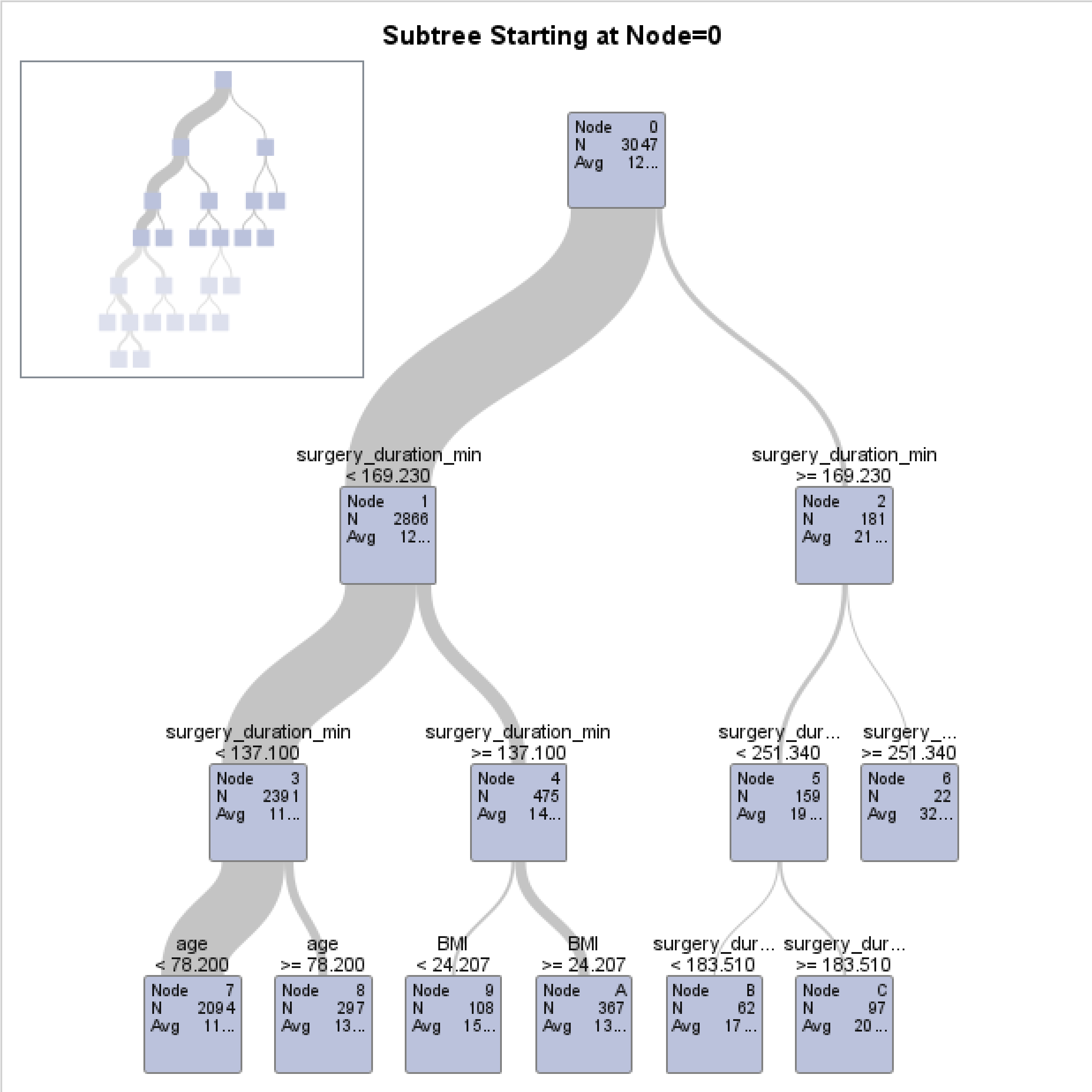
The HPSPLIT Procedure



The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_dur...
< 251.340
Node 5
N 159
Avg 19...
surgery_...
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
13	7242180	2.207E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	140554	6
age	0.2287	32141.5	4
BMI	0.1187	16676.8	1
ASA	0.0862	12112.9	1

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

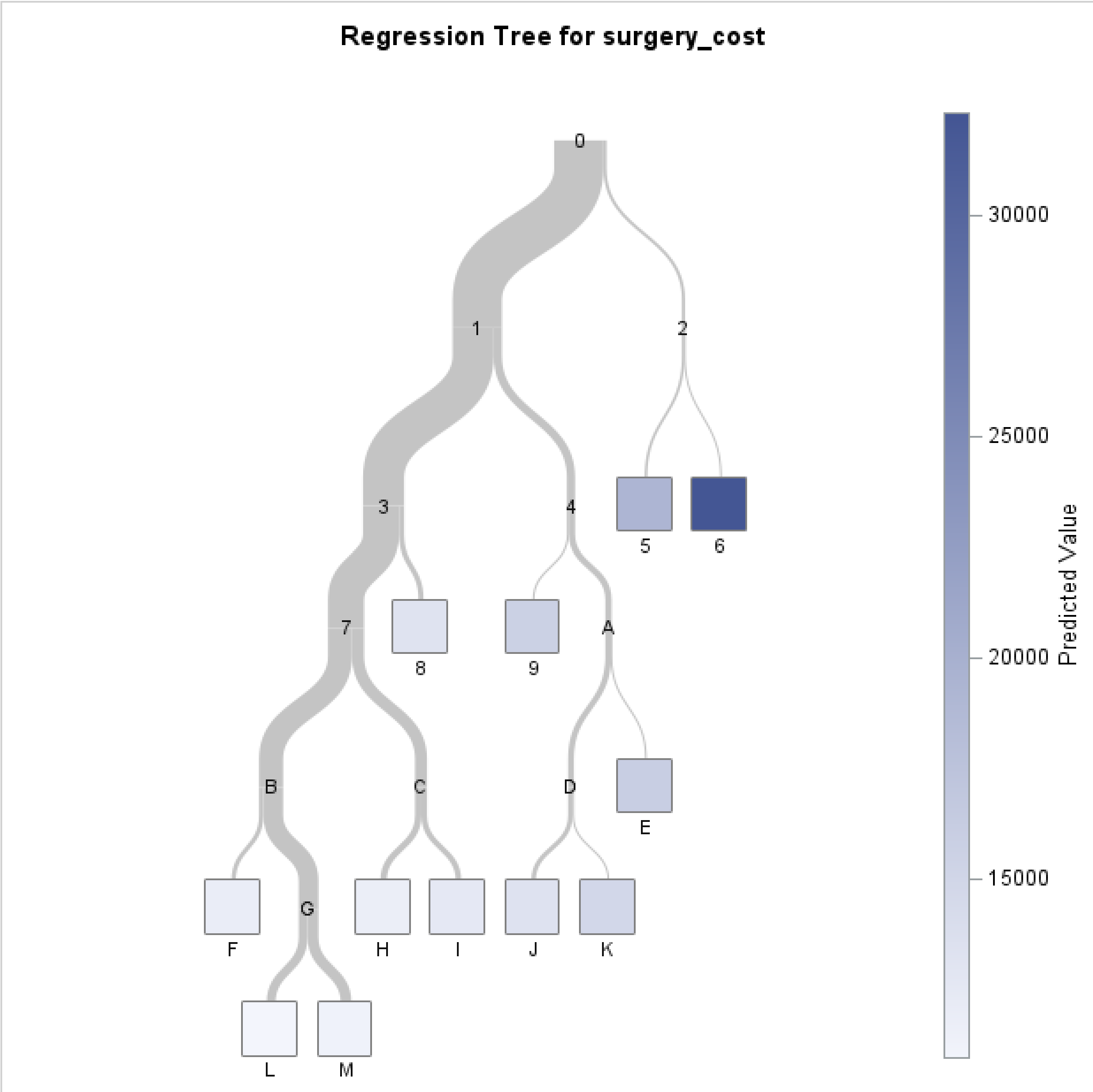
Model Information

Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	6
Number of Leaves Before Pruning	393
Number of Leaves After Pruning	12

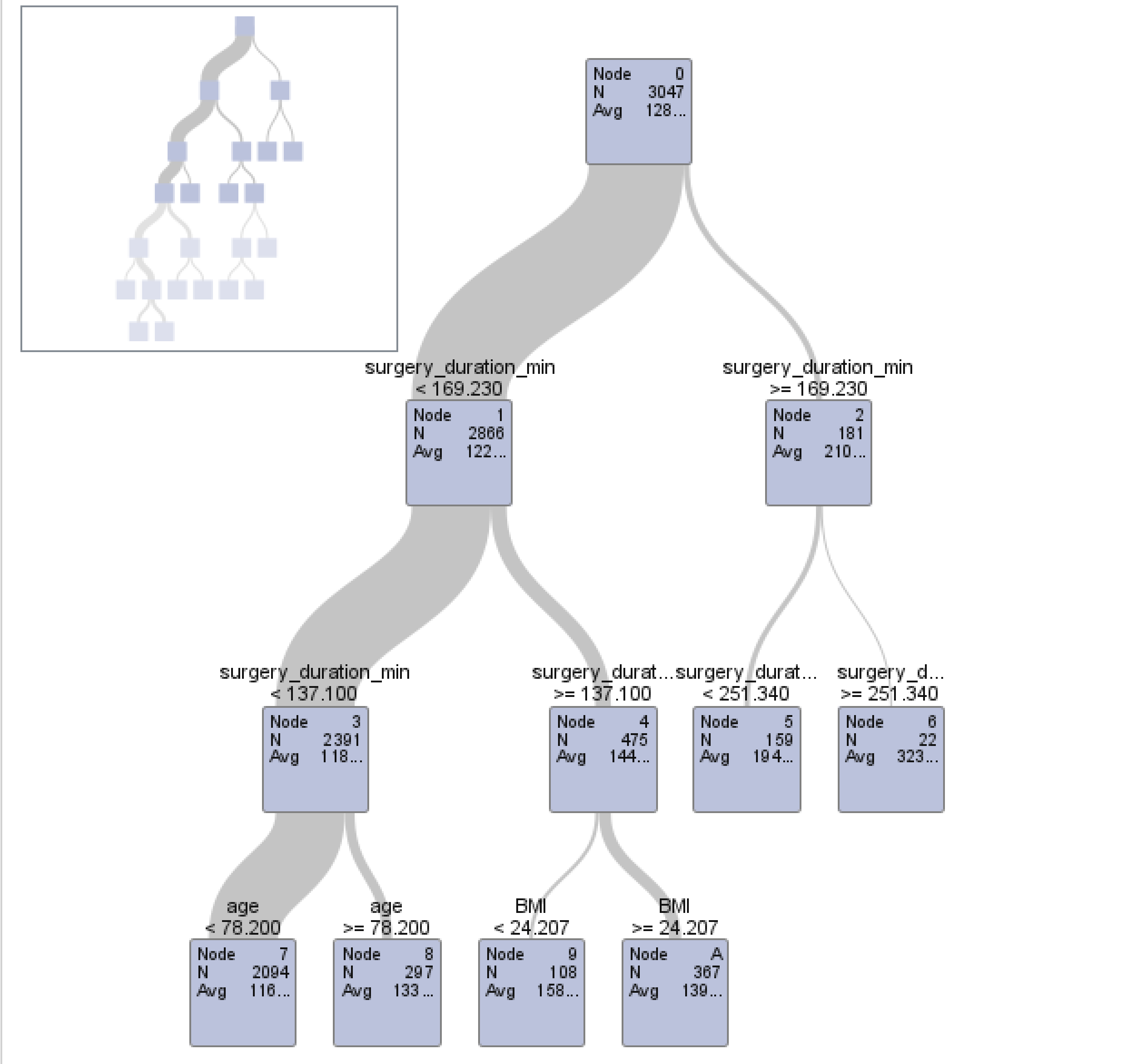
Number of Observations Read	3047
Number of Observations Used	3047

The SAS System

The HPSPLIT Procedure



Subtree Starting at Node=0



The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
12	7415512	2.26E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	138663	5
age	0.2318	32141.5	4
BMI	0.1203	16676.8	1
ASA	0.0874	12112.9	1

accuracy10	accuracy15	accuracy20
0.51117	0.660972	0.78318

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

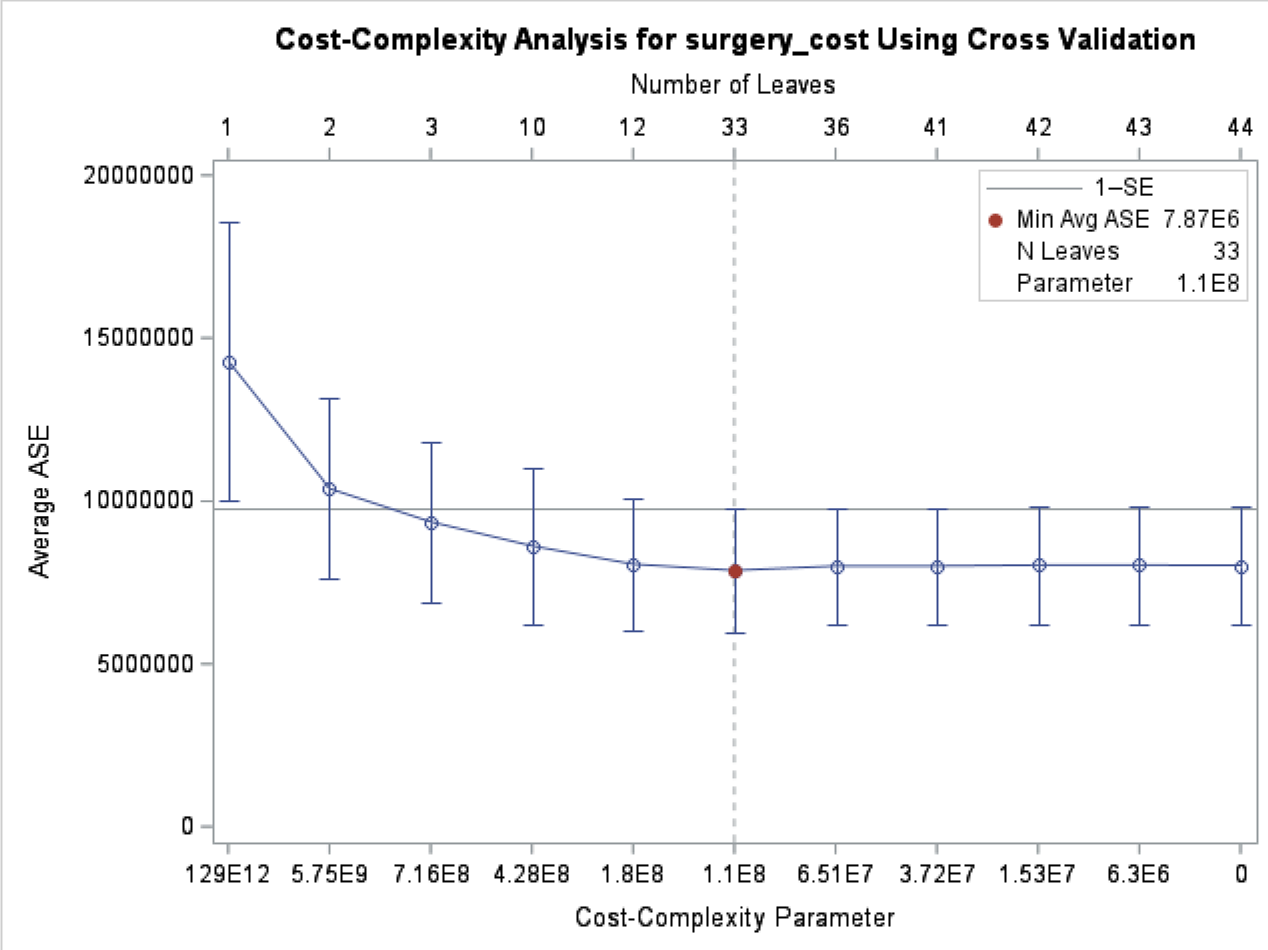
Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client

Model Information

Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	32

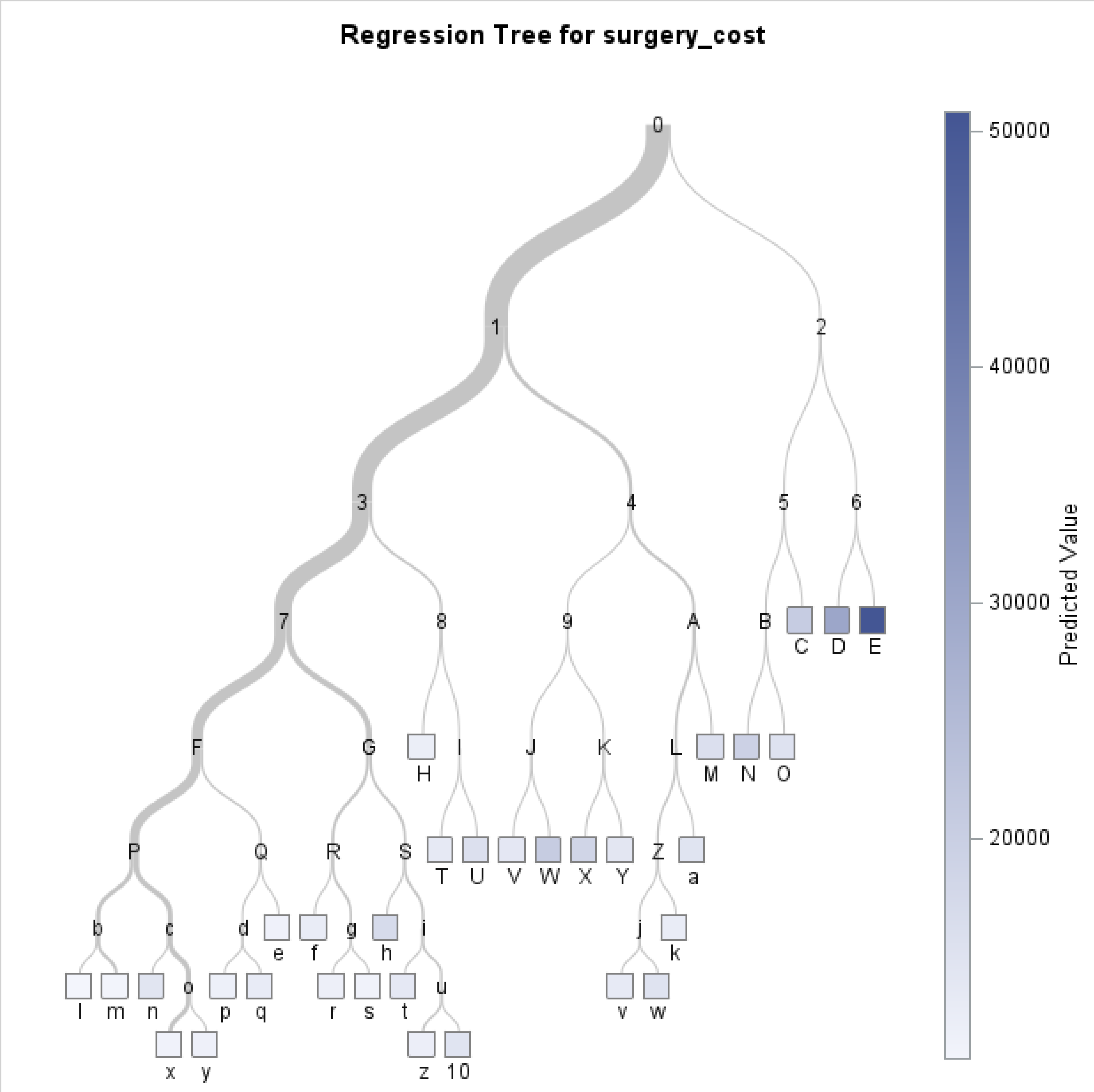
Number of Observations Read	3047
Number of Observations Used	3047

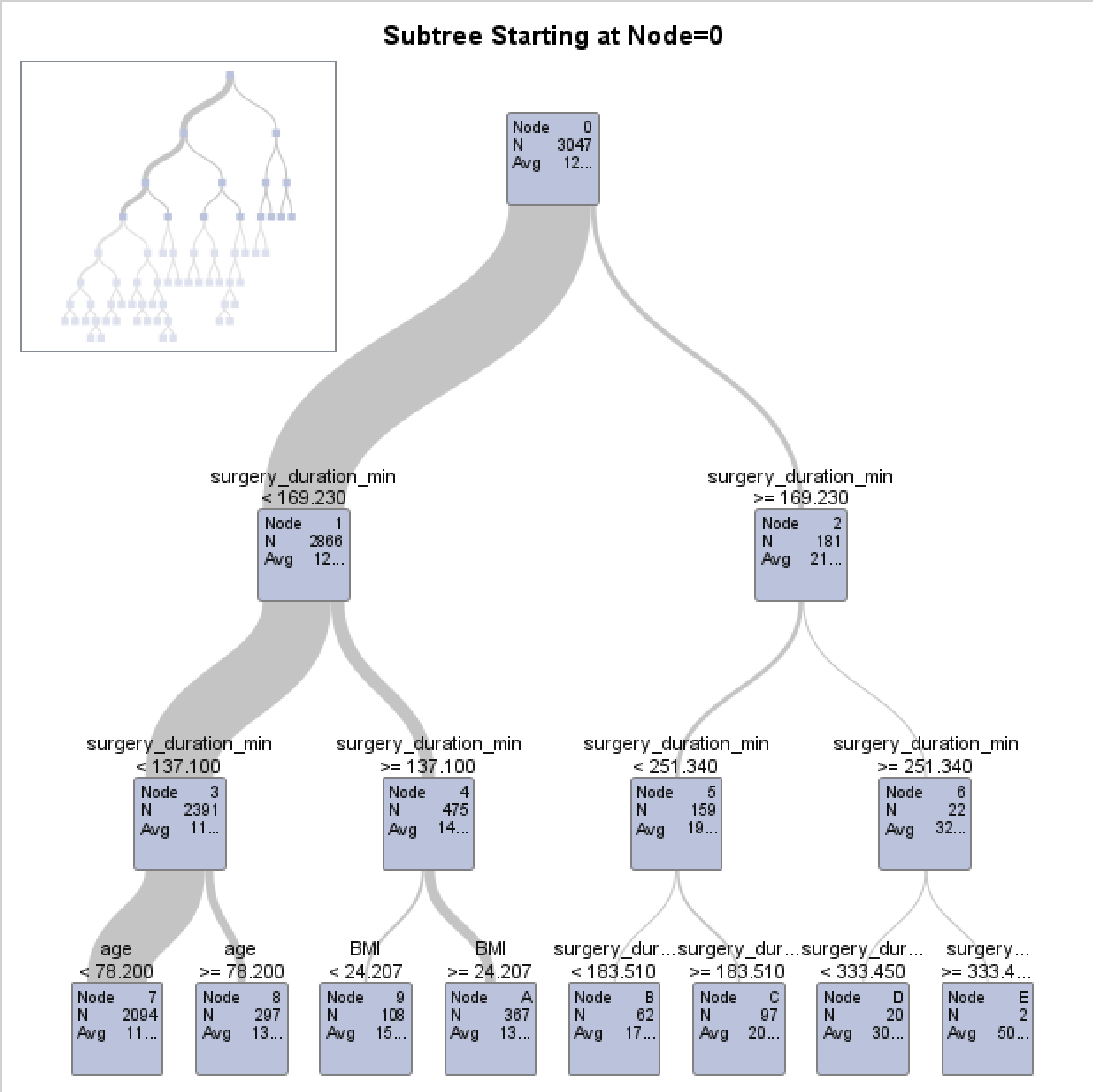
The HPSPLIT Procedure



The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0 N 3047 Avg 12...
surgery_duration_min < 169.230
Node 1 N 2866 Avg 12...
surgery_duration_min >= 169.230
Node 2 N 181 Avg 21...
surgery_duration_min < 137.100
Node 3 N 2391 Avg 11...
surgery_duration_min >= 137.100
Node 4 N 475 Avg 14...
surgery_duration_min < 251.340
Node 5 N 159 Avg 19...
surgery_duration_min >= 251.340
Node 6 N 22 Avg 32...
age < 78.200
Node 7 N 2094 Avg 11...
age >= 78.200
Node 8 N 297 Avg 13...
BMI < 24.207
Node 9 N 108 Avg 15...
BMI >= 24.207
Node A N 367 Avg 13...
surgery_dur... < 183.510
Node B N 62 Avg 17...
surgery_dur... >= 183.510
Node C N 97 Avg 20...
surgery_dur... < 333.450
Node D N 20 Avg 30...
surgery... >= 333.4...
Node E N 2 Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
32	6469173	1.971E10

Variable Importance

Variable	Training Relative	Training Importance	Count
surgery_duration_min	1.0000	144542	11
age	0.2588	37412.2	9
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.HOSPITAL	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

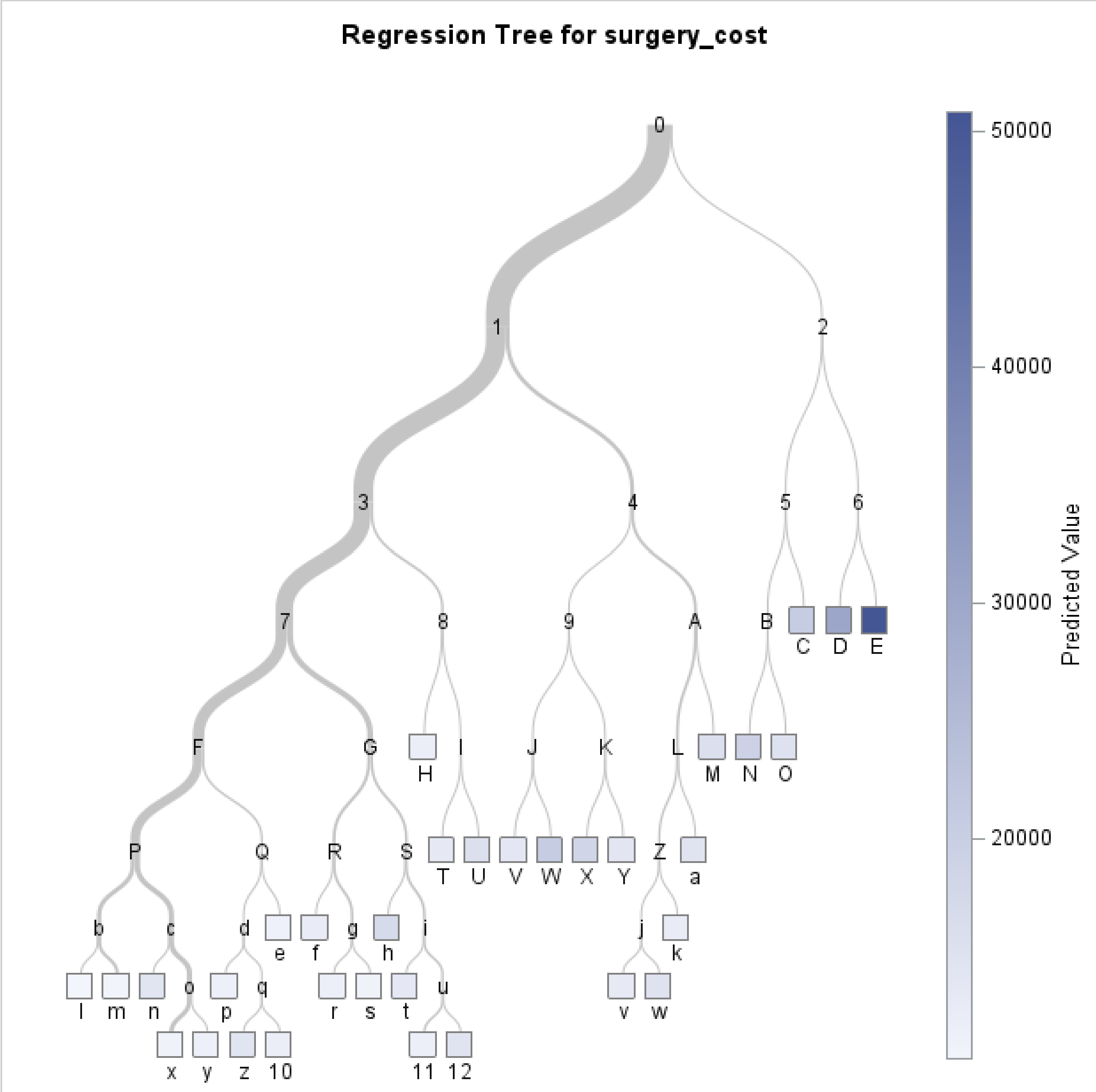
Model Information

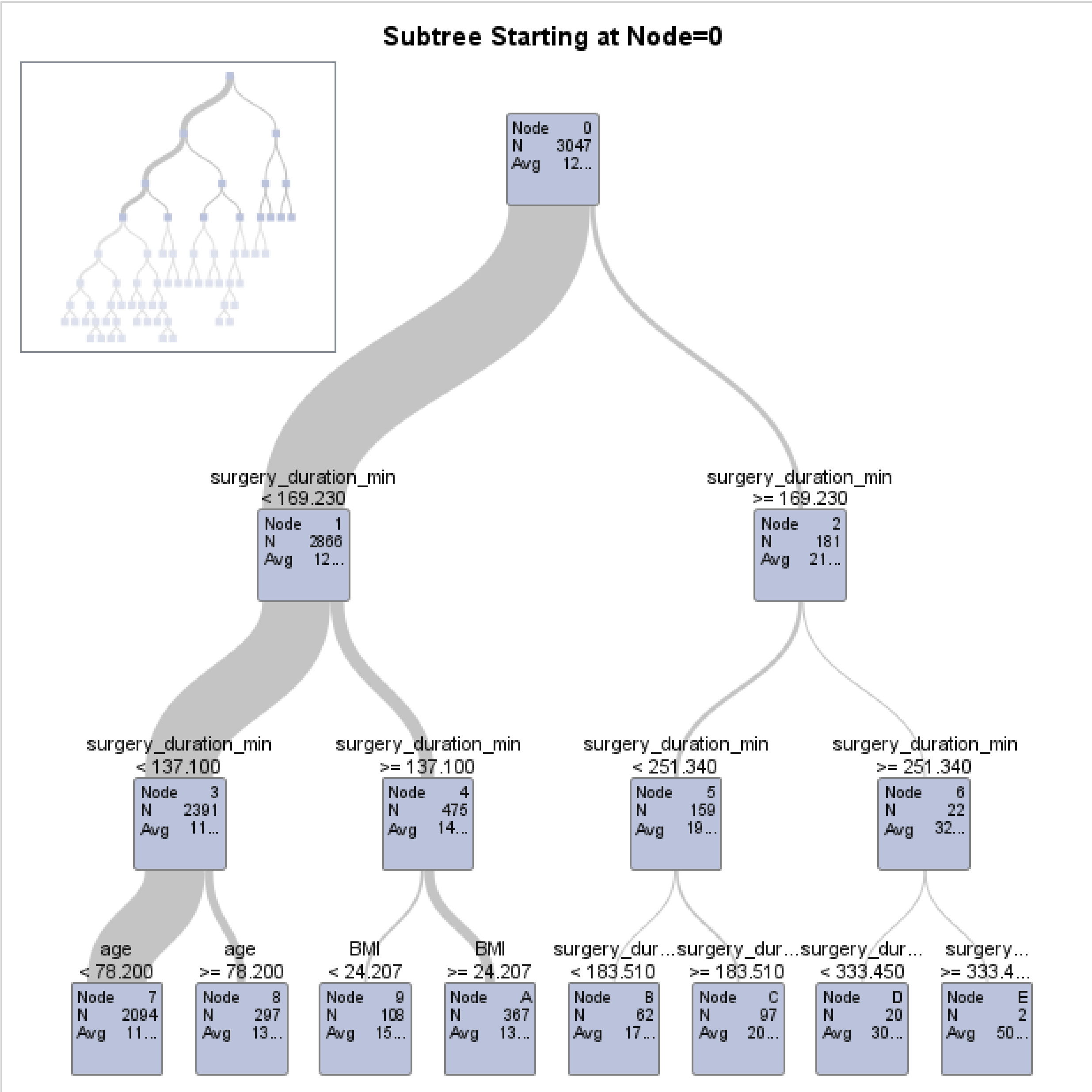
Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Number of Leaves
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	9
Tree Depth	8
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	33

Number of Observations Read	3047
Number of Observations Used	3047

The SAS System

The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 3047
Avg 12...
surgery_duration_min
< 169.230
Node 1
N 2866
Avg 12...
surgery_duration_min
>= 169.230
Node 2
N 181
Avg 21...
surgery_duration_min
< 137.100
Node 3
N 2391
Avg 11...
surgery_duration_min
>= 137.100
Node 4
N 475
Avg 14...
surgery_duration_min
< 251.340
Node 5
N 159
Avg 19...
surgery_duration_min
>= 251.340
Node 6
N 22
Avg 32...
age
< 78.200
Node 7
N 2094
Avg 11...
age
>= 78.200
Node 8
N 297
Avg 13...
BMI
< 24.207
Node 9
N 108
Avg 15...
BMI
>= 24.207
Node A
N 367
Avg 13...
surgery_dur...
< 183.510
Node B
N 62
Avg 17...
surgery_dur...
>= 183.510
Node C
N 97
Avg 20...
surgery_dur...
< 333.450
Node D
N 20
Avg 30...
surgery...
>= 333.4...
Node E
N 2
Avg 50...

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	RSS
33	6449015	1.965E10

Variable Importance

Variable	Training		Count
	Relative	Importance	
surgery_duration_min	1.0000	144542	11
age	0.2645	38224.2	10
BMI	0.1871	27046.7	5
gender	0.1261	18227.6	4
ASA	0.1010	14597.3	2

accuracy10	accuracy15	accuracy20
0.507227	0.670171	0.805519

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	CARD_DATA
Random Number Seed	122470
Sampling Rate	0.8
Sample Size	1600
Selection Probability	0.8
Sampling Weight	0
Output Data Set	CARD_DATA

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

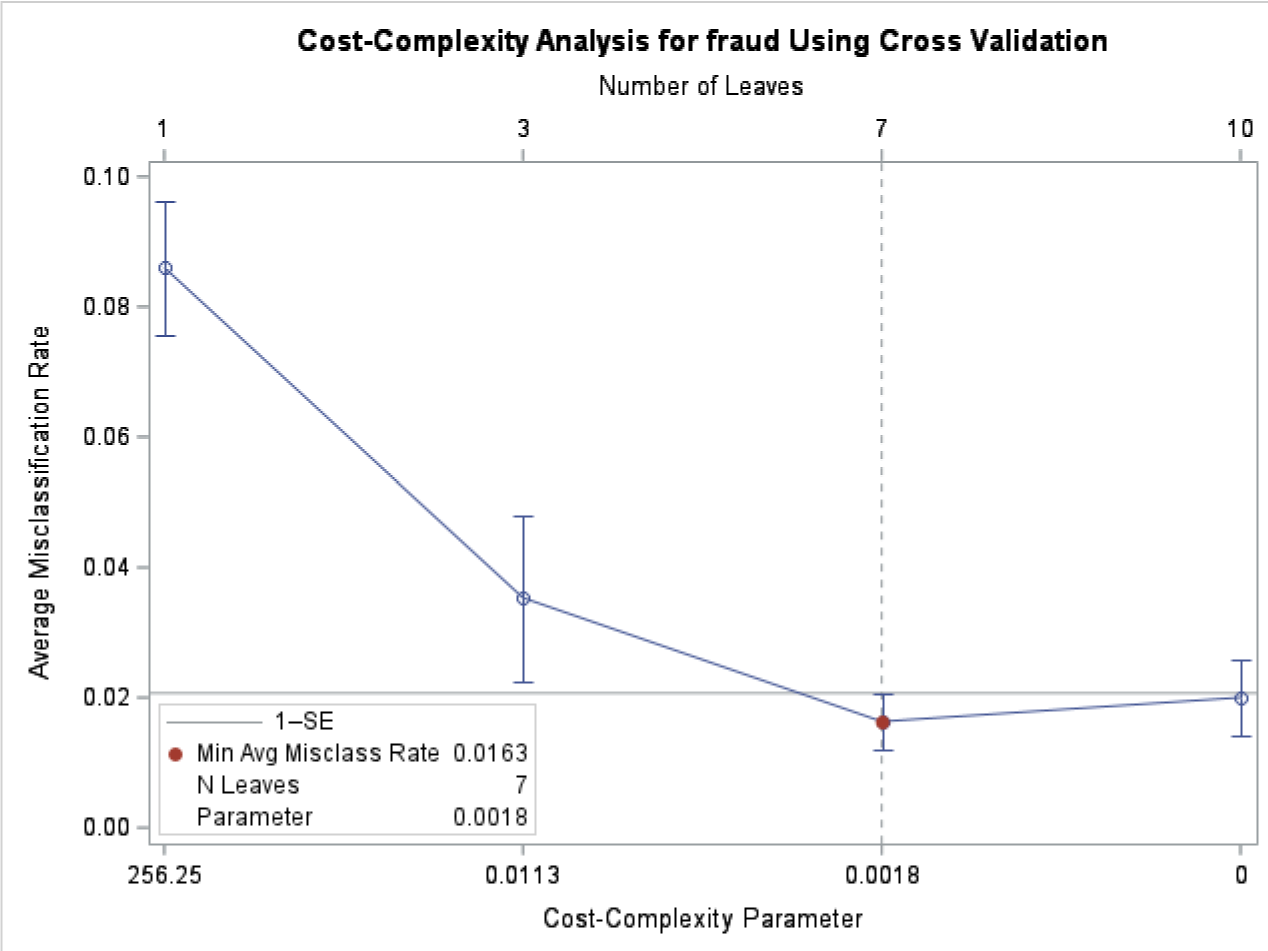
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

Model Information

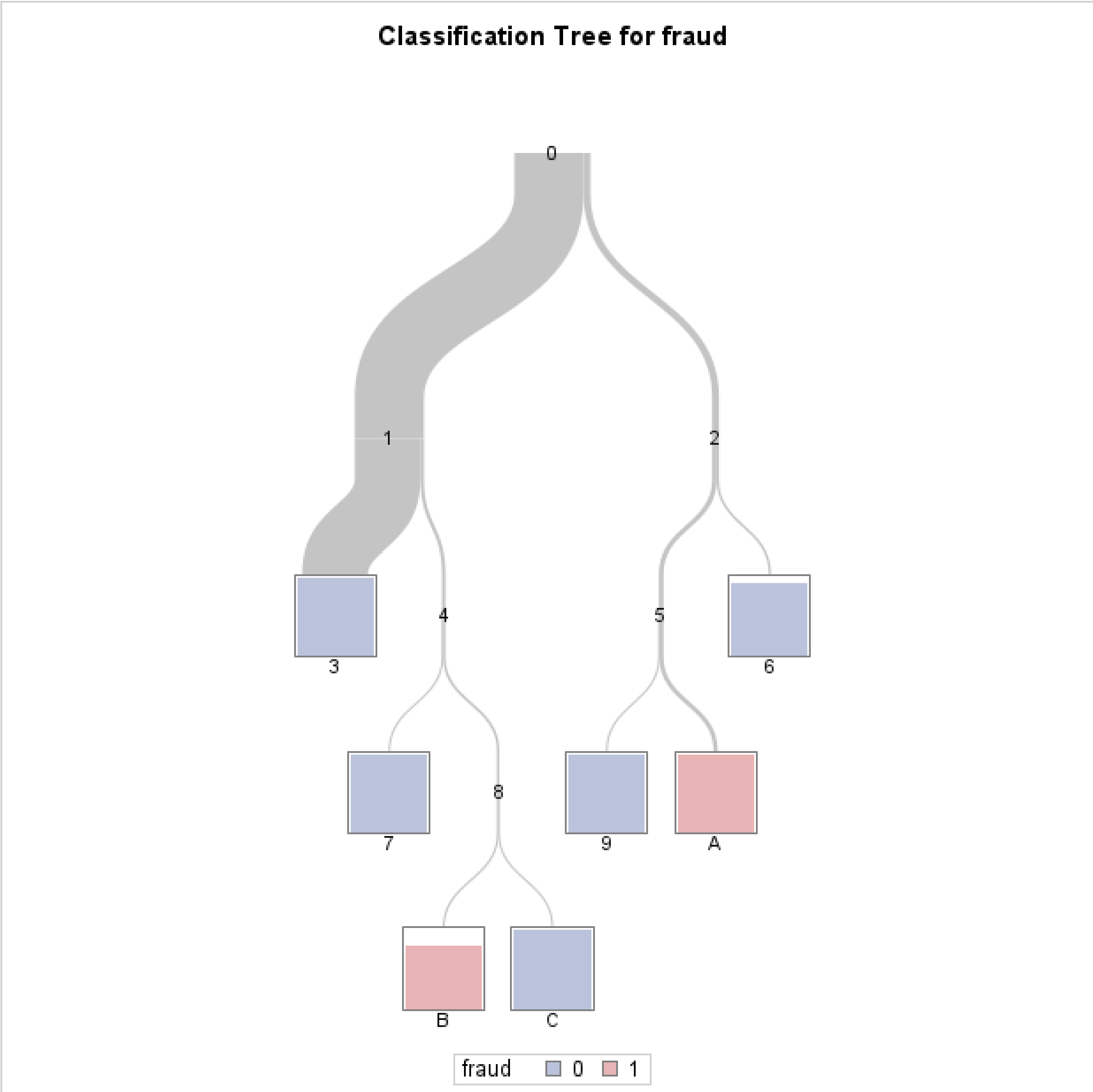
Split Criterion Used	Gini
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	12
Number of Leaves After Pruning	7
Model Event Level	1

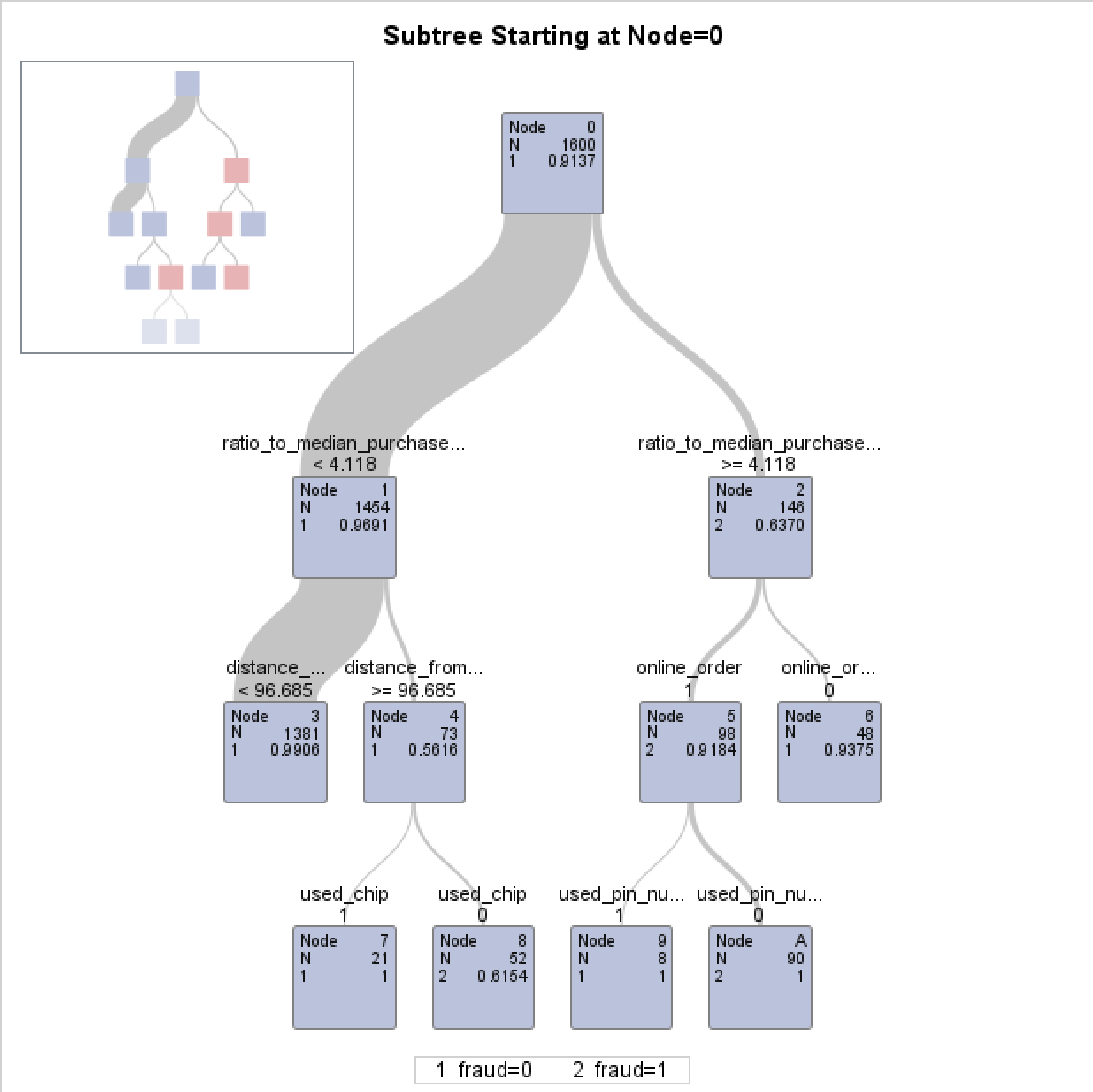
Number of Observations Read	1600
Number of Observations Used	1600

The HPSPLIT Procedure



The HPSPLIT Procedure




Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 4.118
Node 1
N 1454
1 0.9691
ratio_to_median_purchase...
>= 4.118
Node 2
N 146
2 0.6370
distance_...
< 96.685
Node 3
N 1381
1 0.9906
distance_from...
>= 96.685
Node 4
N 73
1 0.5616
online_order
1
Node 5
N 98
2 0.9184
online_or...
0
Node 6
N 48
1 0.9375
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 52
2 0.6154
used_pin_nu...
1
Node 9
N 8
1 1
used_pin_nu...
0
Node A
N 90
2 1
1 fraud=0 2 fraud=1

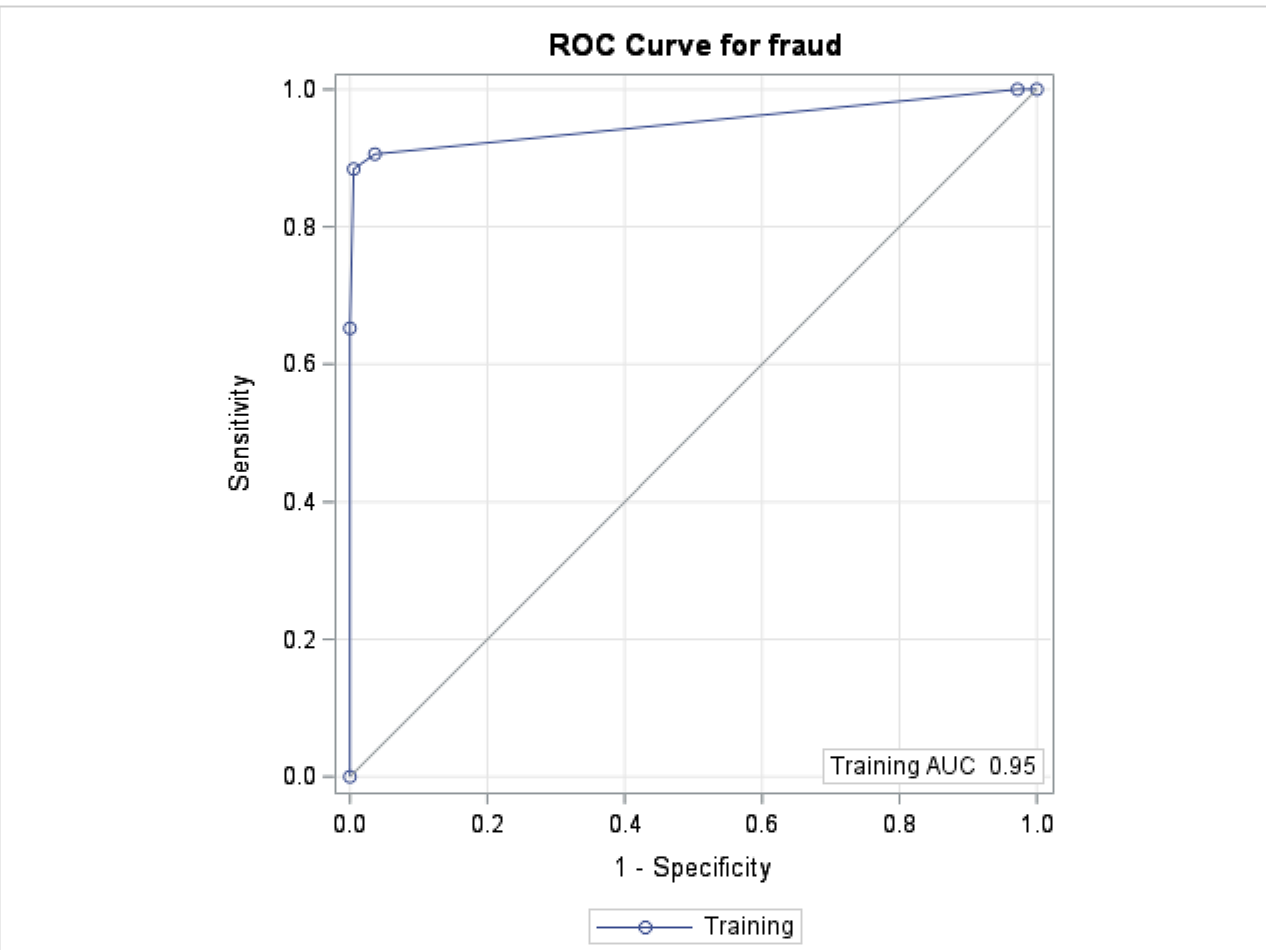
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1454	8	0.0055
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
7	0.0138	0.0150	0.8841	0.9945	0.0945	0.0276	44.1802	0.9514



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.3883	3.8333	1
used_chip	0.3410	3.3660	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	trueclassrate
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

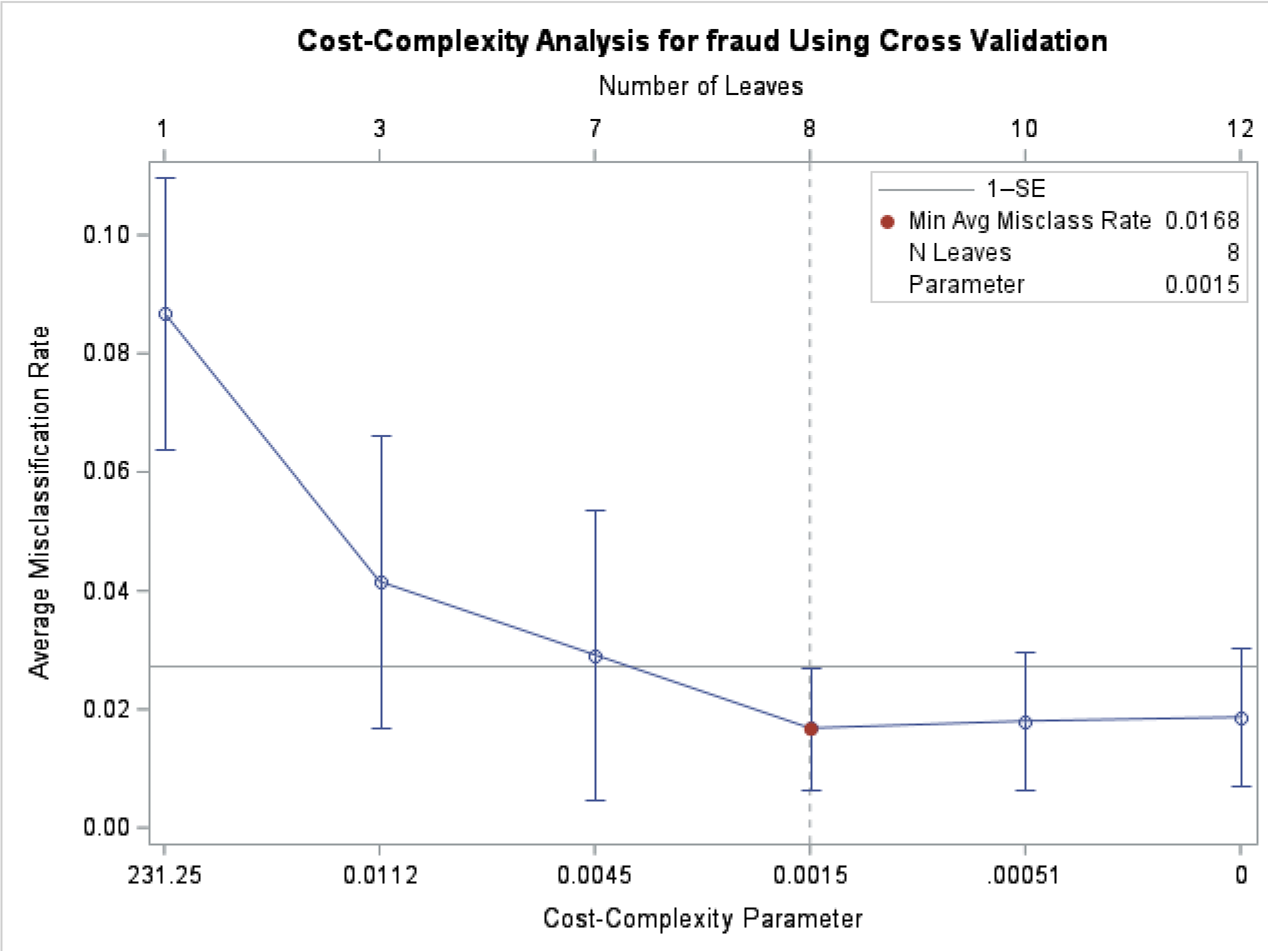
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED2	V9	Output	On Client

Model Information

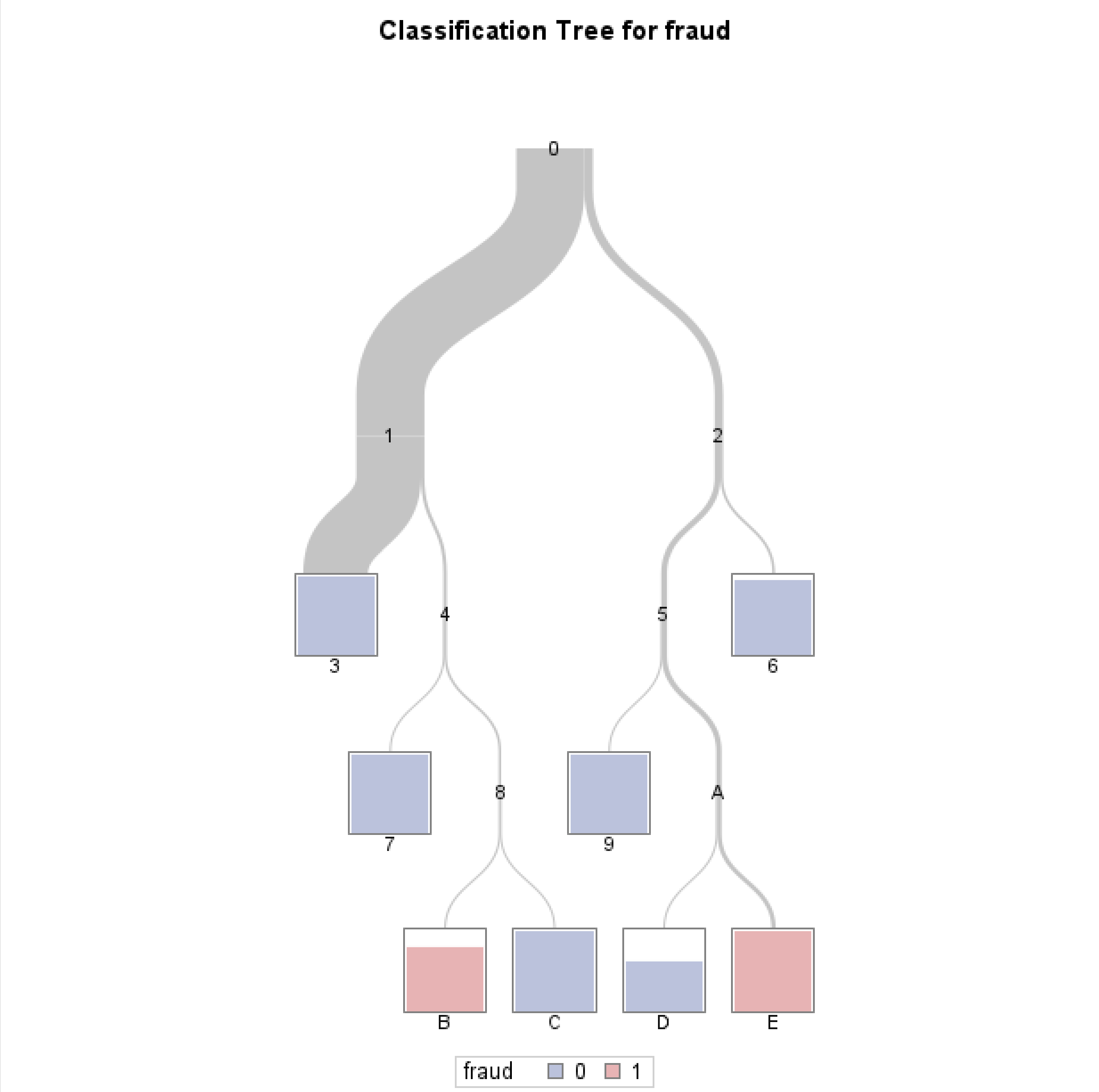
Split Criterion Used	Entropy
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	13
Number of Leaves After Pruning	8
Model Event Level	1

Number of Observations Read	1600
Number of Observations Used	1600

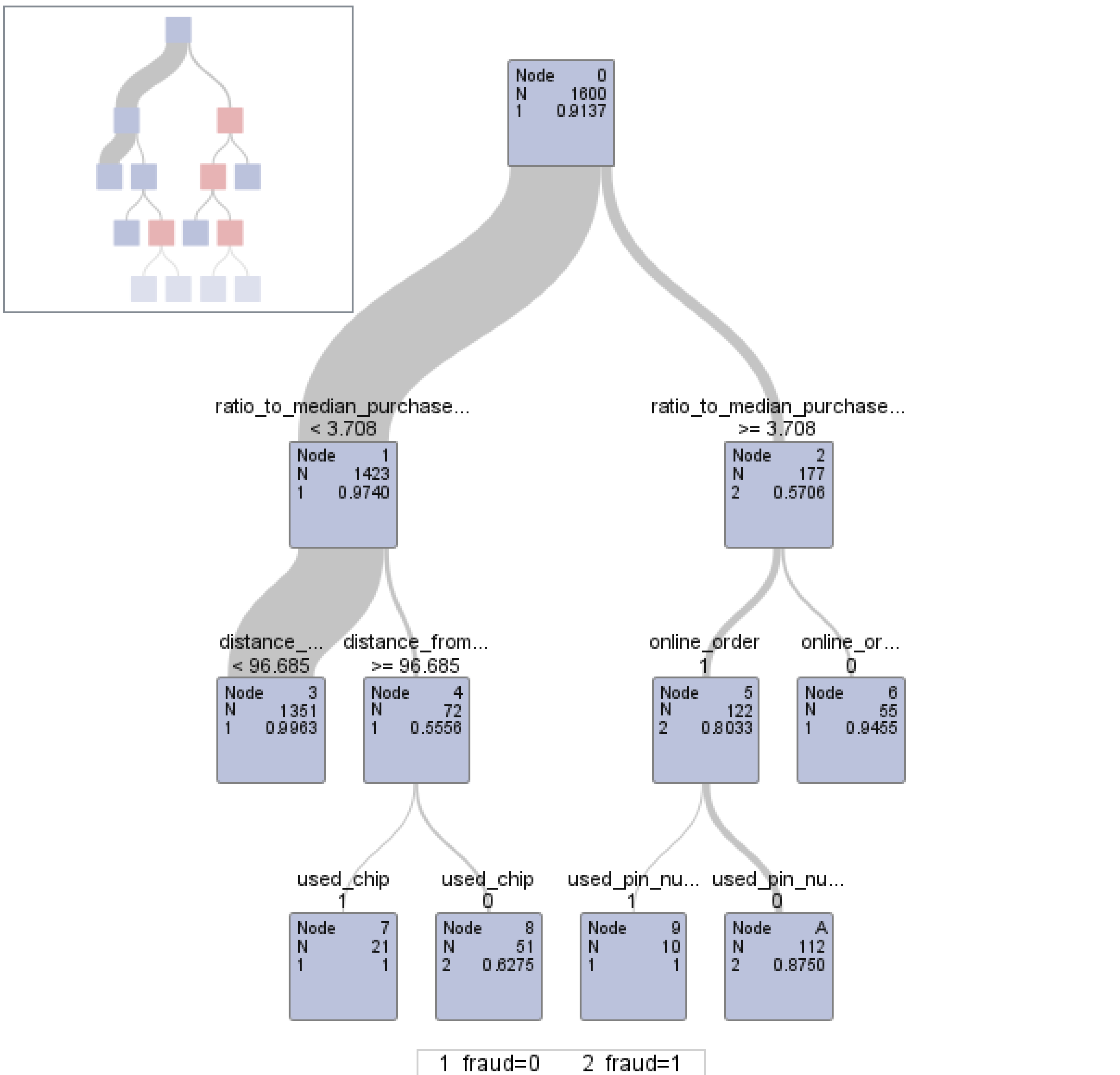
The HPSPLIT Procedure



The HPSPLIT Procedure



Subtree Starting at Node=0



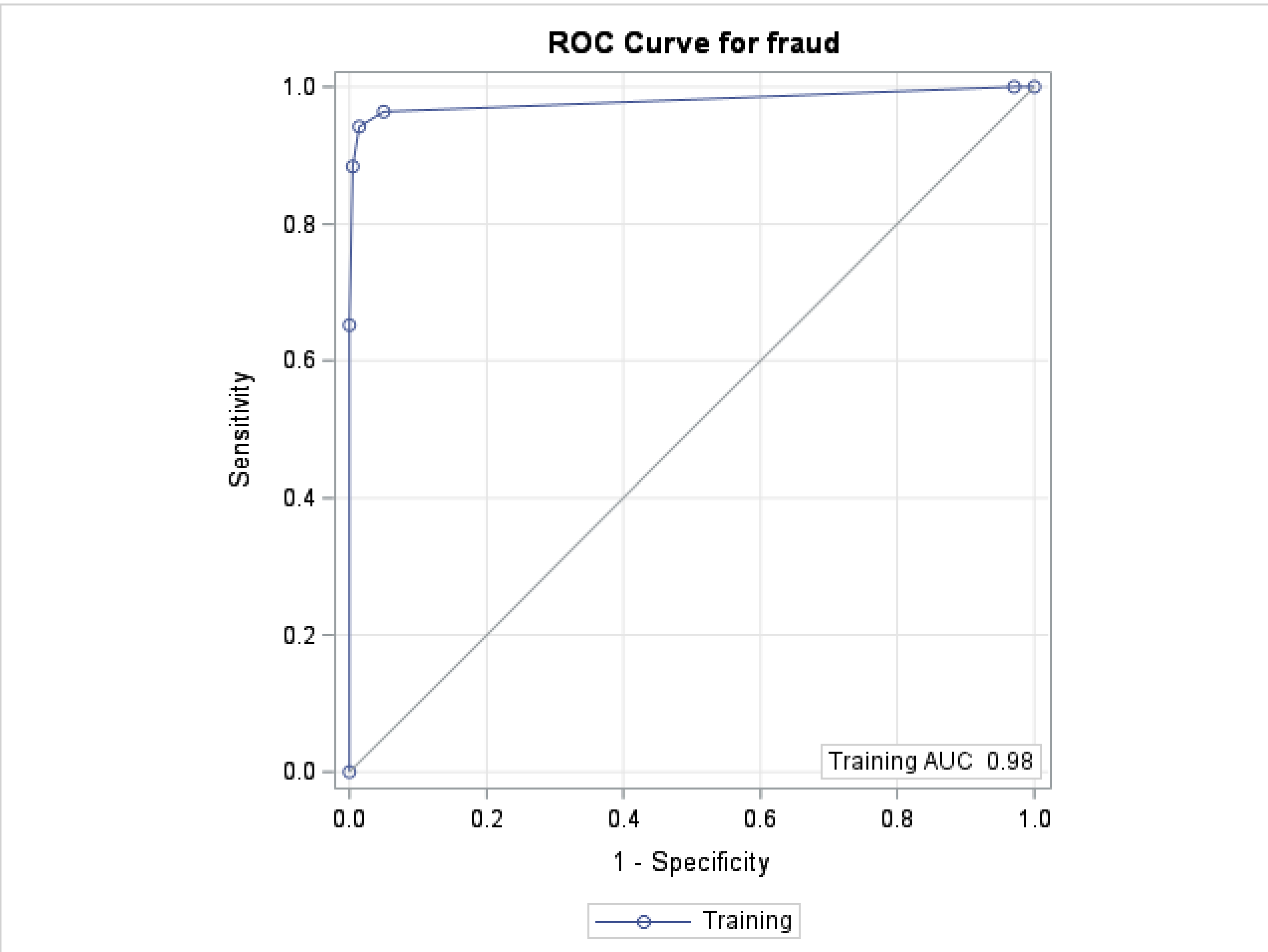
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1455	7	0.0048
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
8	0.0117	0.0144	0.8841	0.9952	0.0698	0.0233	37.3047	0.9797



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	10.3780	2
online_order	0.7137	7.4068	2
distance_from_home	0.4966	5.1534	1
used_pin_number	0.3613	3.7493	1
used_chip	0.3298	3.4223	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	trueclassrate
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

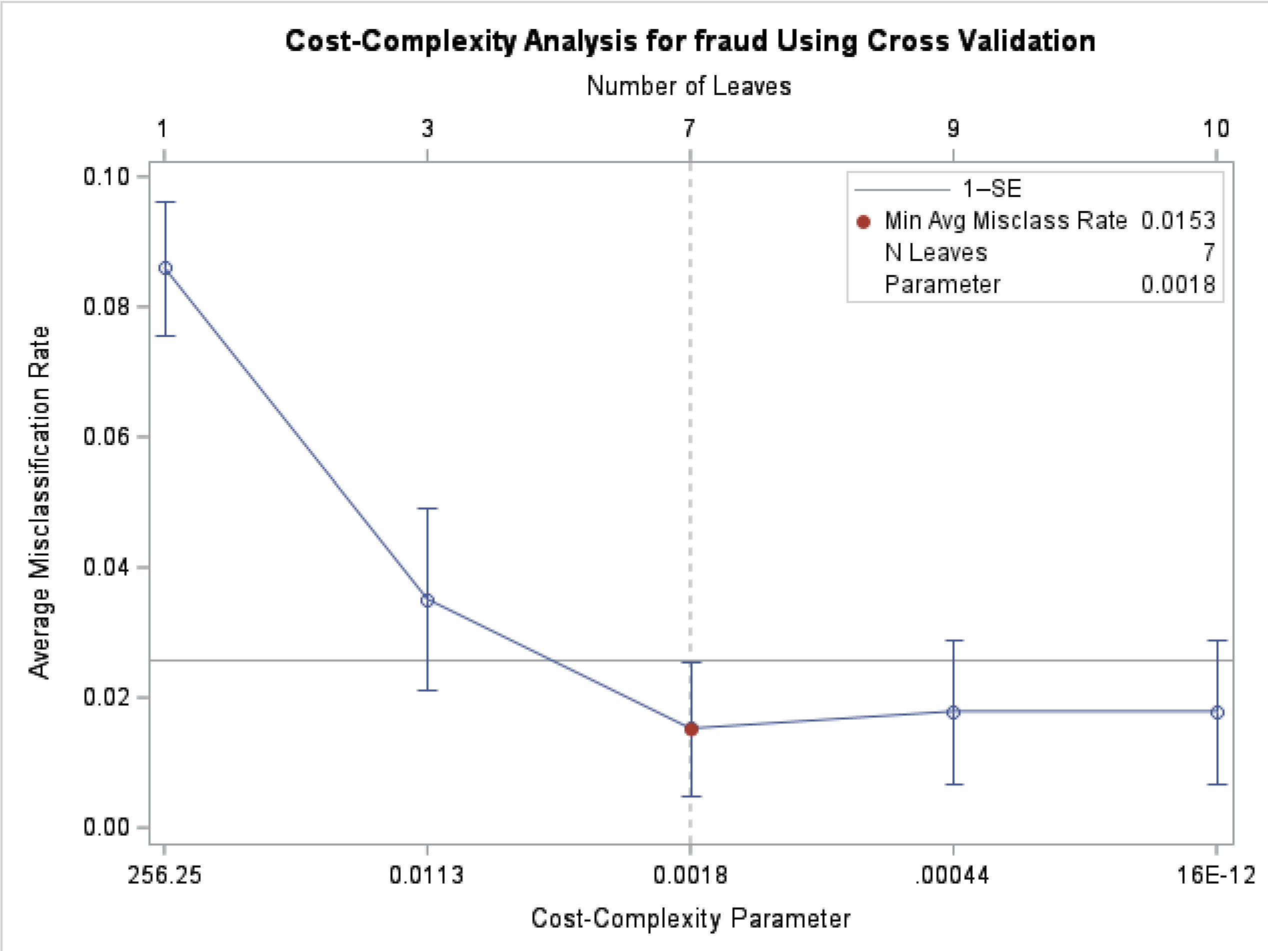
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED3	V9	Output	On Client

Model Information

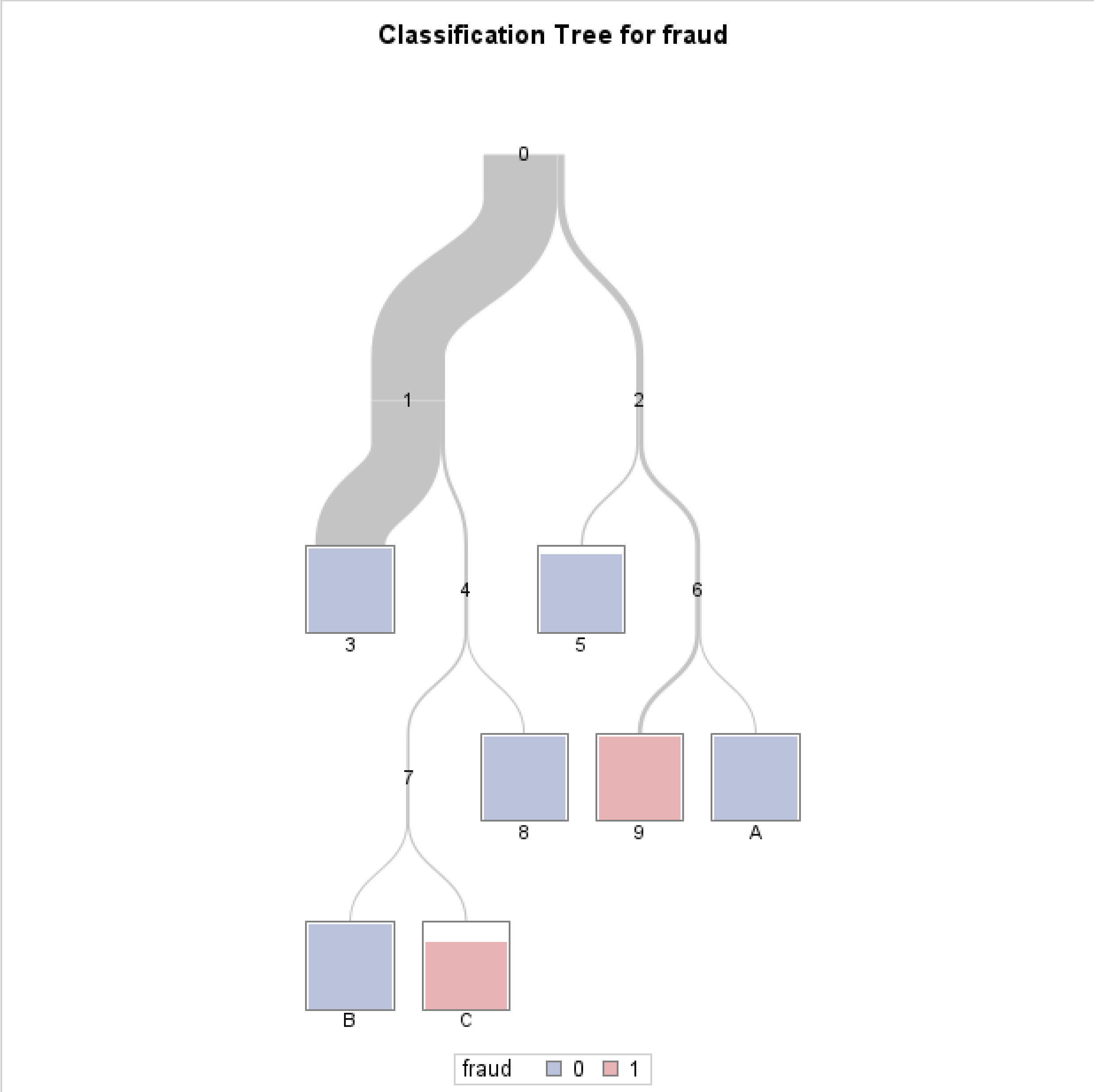
Split Criterion Used	CHAID
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	4
Maximum Tree Depth Achieved	4
Tree Depth	4
Number of Leaves Before Pruning	12
Number of Leaves After Pruning	7
Model Event Level	1

Number of Observations Read	1600
Number of Observations Used	1600

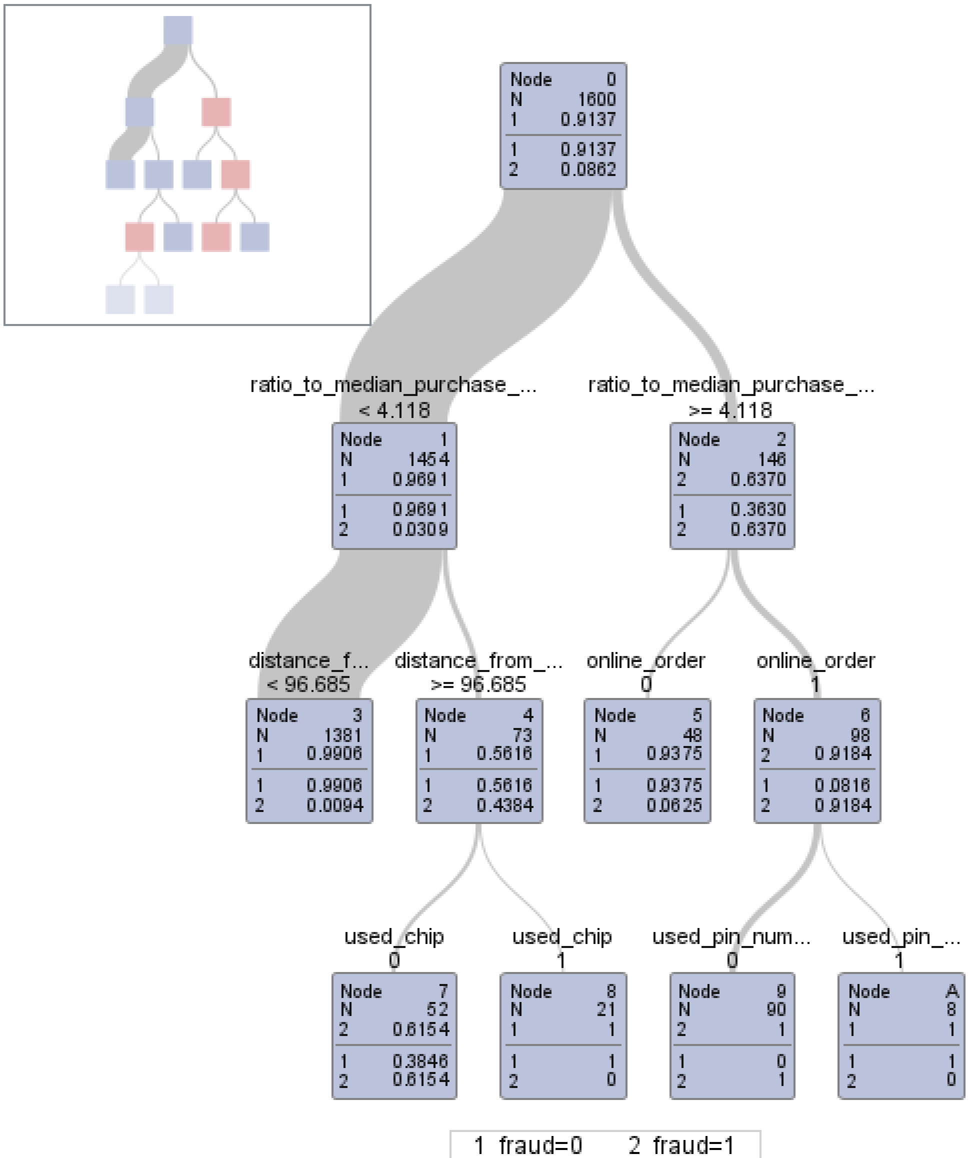
The HPSPLIT Procedure



The HPSPLIT Procedure



Subtree Starting at Node=0



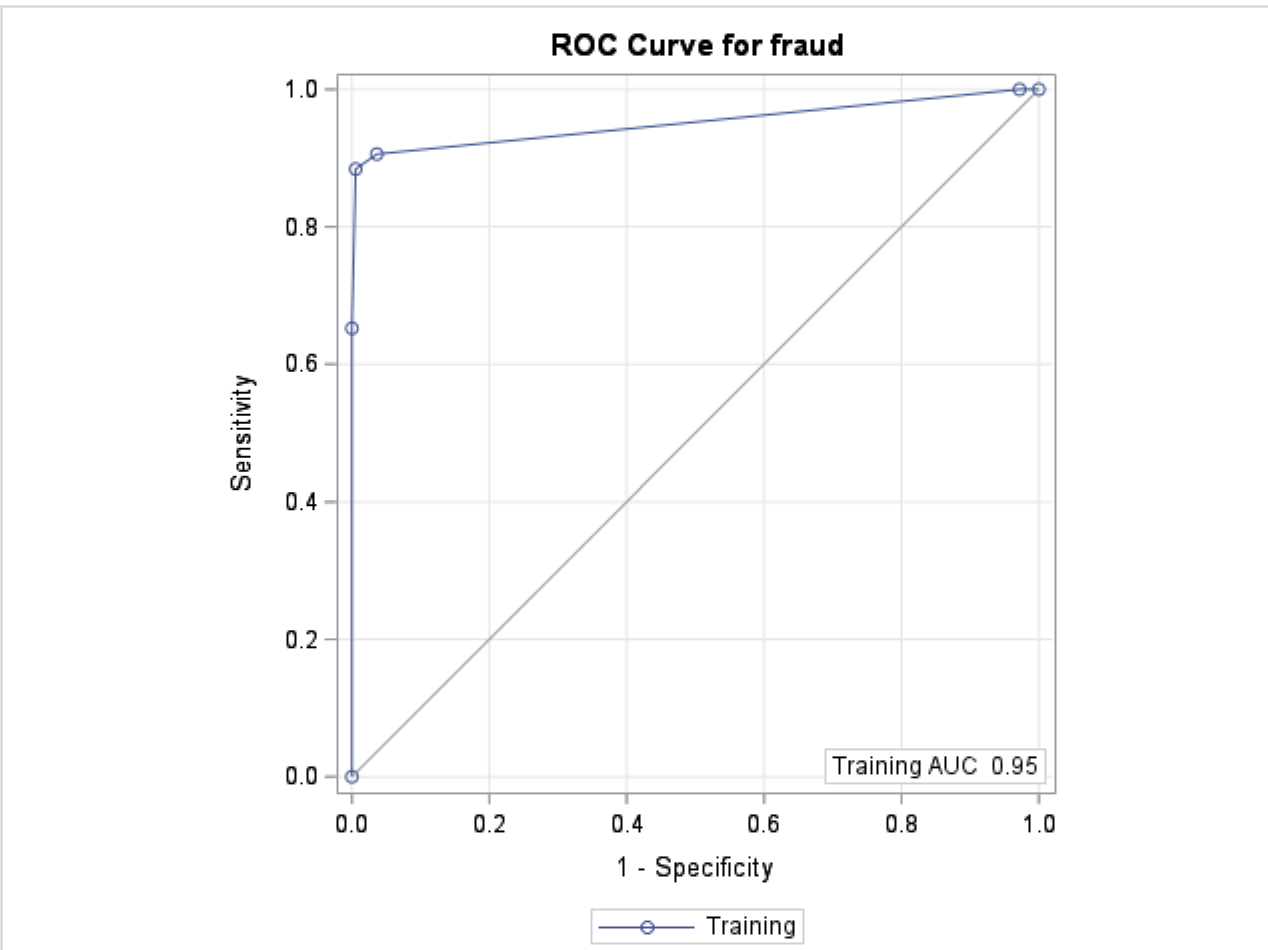
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1454	8	0.0055
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
7	0.0138	0.0150	0.8841	0.9945	0.0945	0.0276	44.1802	0.9514



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.3883	3.8333	1
used_chip	0.3410	3.3660	1

cutoff	trueclassrate
0.01	0.9025
0.02	0.9025
0.03	0.9025
0.04	0.9025
0.05	0.9025
0.06	0.9025
0.07	0.9025
0.08	0.9025
0.09	0.9025
0.1	0.9025
0.11	0.9025
0.12	0.9025
0.13	0.9025
0.14	0.9025
0.15	0.9025
0.16	0.9025
0.17	0.9025
0.18	0.9025
0.19	0.9025
0.2	0.9025
0.21	0.9025
0.22	0.9025
0.23	0.9025
0.24	0.9025
0.25	0.9025
0.26	0.9025
0.27	0.9025

cutoff	trueclassrate
0.28	0.9025
0.29	0.9025
0.3	0.9025
0.31	0.9025
0.32	0.9025
0.33	0.9025
0.34	0.9025
0.35	0.9025
0.36	0.9025
0.37	0.9025
0.38	0.9025
0.39	0.9025
0.4	0.9025
0.41	0.9025
0.42	0.9025
0.43	0.9025
0.44	0.9025
0.45	0.9025
0.46	0.9025
0.47	0.9025
0.48	0.9025
0.49	0.9025
0.5	0.9025
0.51	0.9025
0.52	0.9025
0.53	0.9025
0.54	0.9025
0.55	0.9025
0.56	0.9025

cutoff	trueclassrate
0.57	0.9025
0.58	0.9025
0.59	0.9025
0.6	0.9025
0.61	0.9025
0.62	0.9025
0.63	0.9025
0.64	0.9025
0.65	0.9025
0.66	0.9025
0.67	0.9025
0.68	0.9025
0.69	0.9025
0.7	0.9025
0.71	0.9025
0.72	0.9025
0.73	0.9025
0.74	0.9025
0.75	0.9025
0.76	0.9025
0.77	0.9025
0.78	0.9025
0.79	0.9025
0.8	0.9025
0.81	0.9025
0.82	0.9025
0.83	0.9025
0.84	0.9025
0.85	0.9025

cutoff	trueclassrate
0.86	0.9025
0.87	0.9025
0.88	0.9025
0.89	0.9025
0.9	0.9025
0.91	0.9025
0.92	0.9025
0.93	0.9025
0.94	0.9025
0.95	0.9025
0.96	0.9025
0.97	0.9025
0.98	0.9025
0.99	0.9025

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	CARD_DATA
Random Number Seed	122470
Sampling Rate	0.8
Sample Size	1600
Selection Probability	0.8
Sampling Weight	0
Output Data Set	CARD_DATA

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

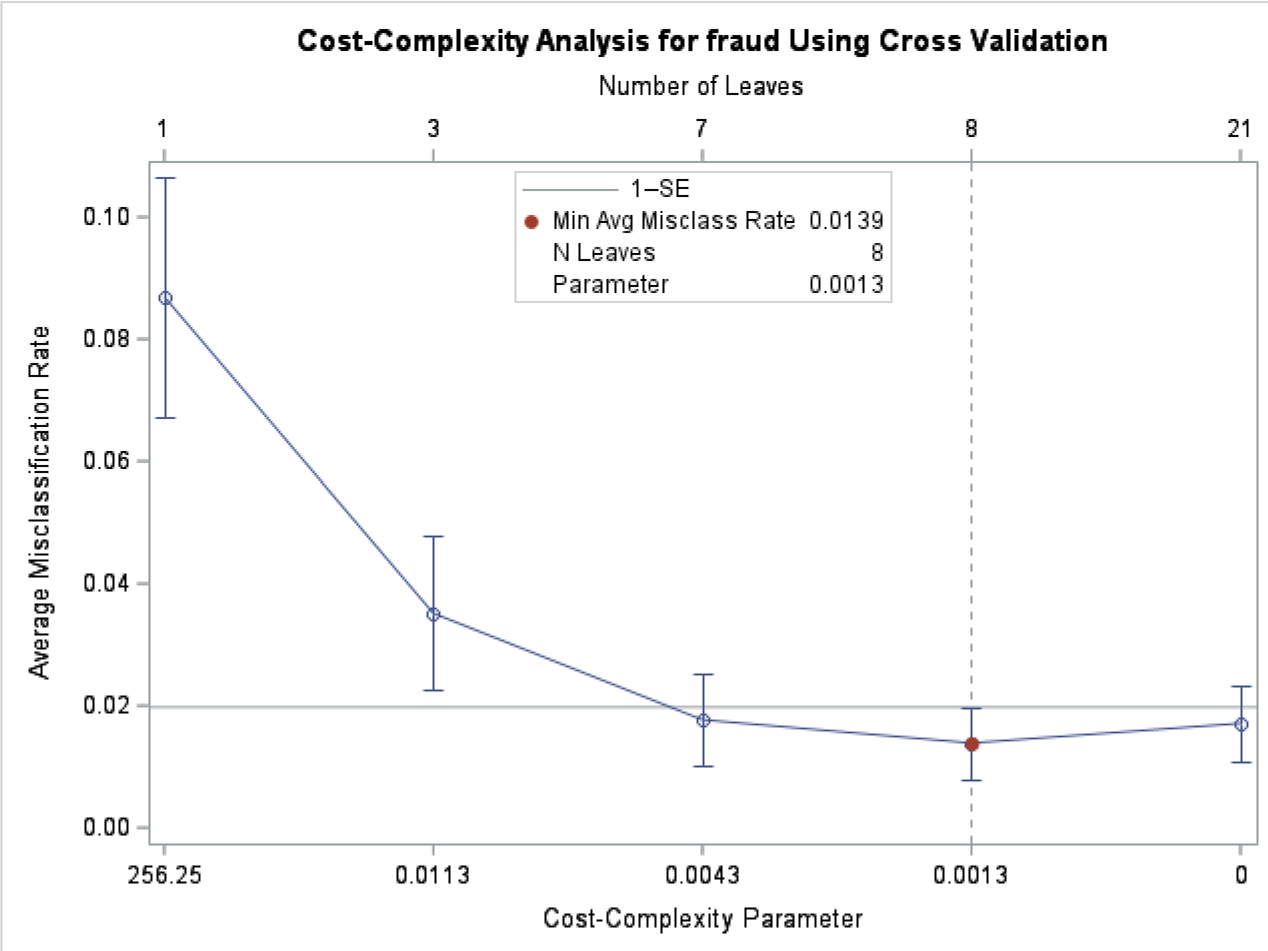
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

Model Information

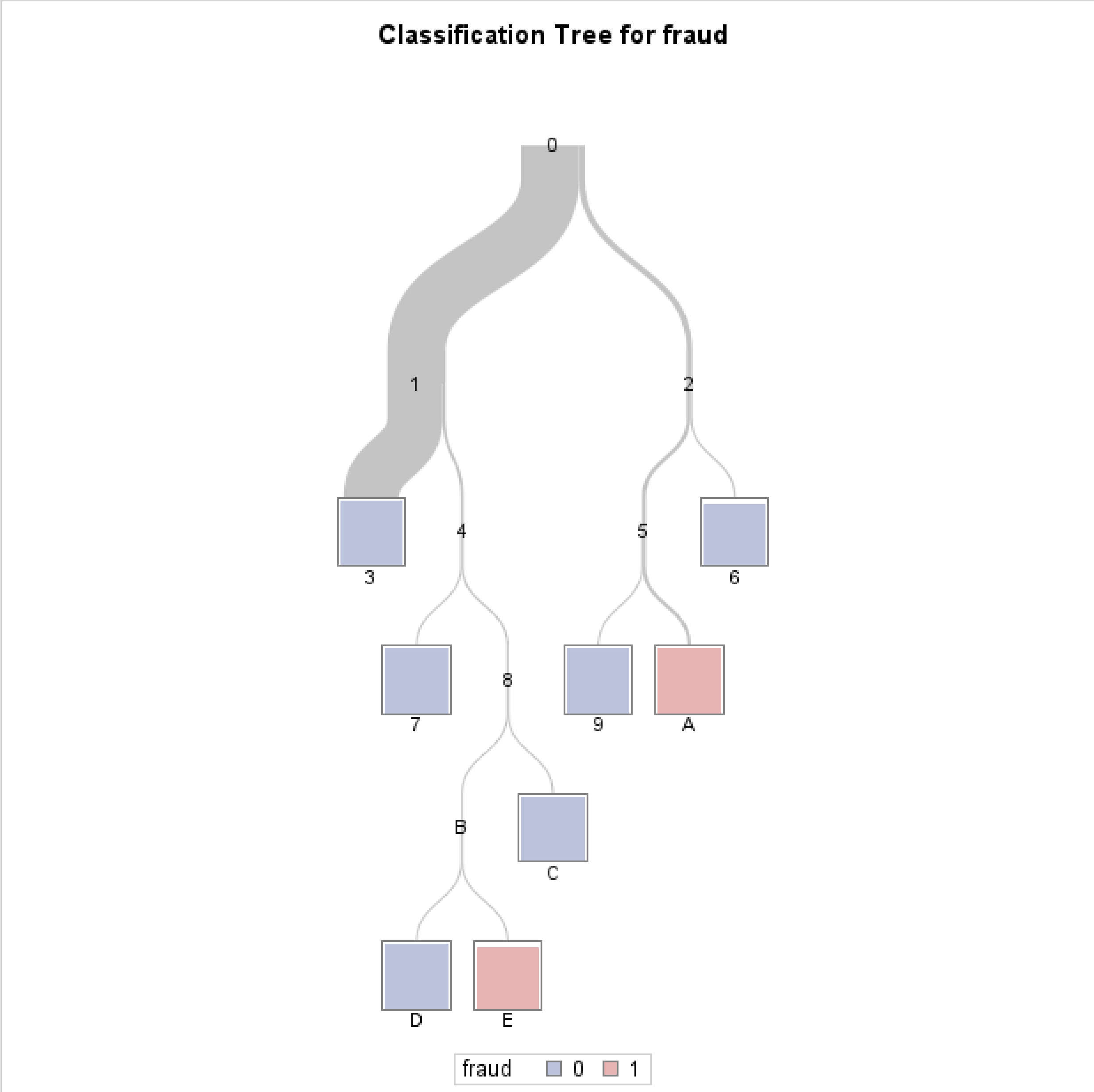
Split Criterion Used	Gini
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	7
Maximum Tree Depth Achieved	7
Tree Depth	5
Number of Leaves Before Pruning	26
Number of Leaves After Pruning	8
Model Event Level	1

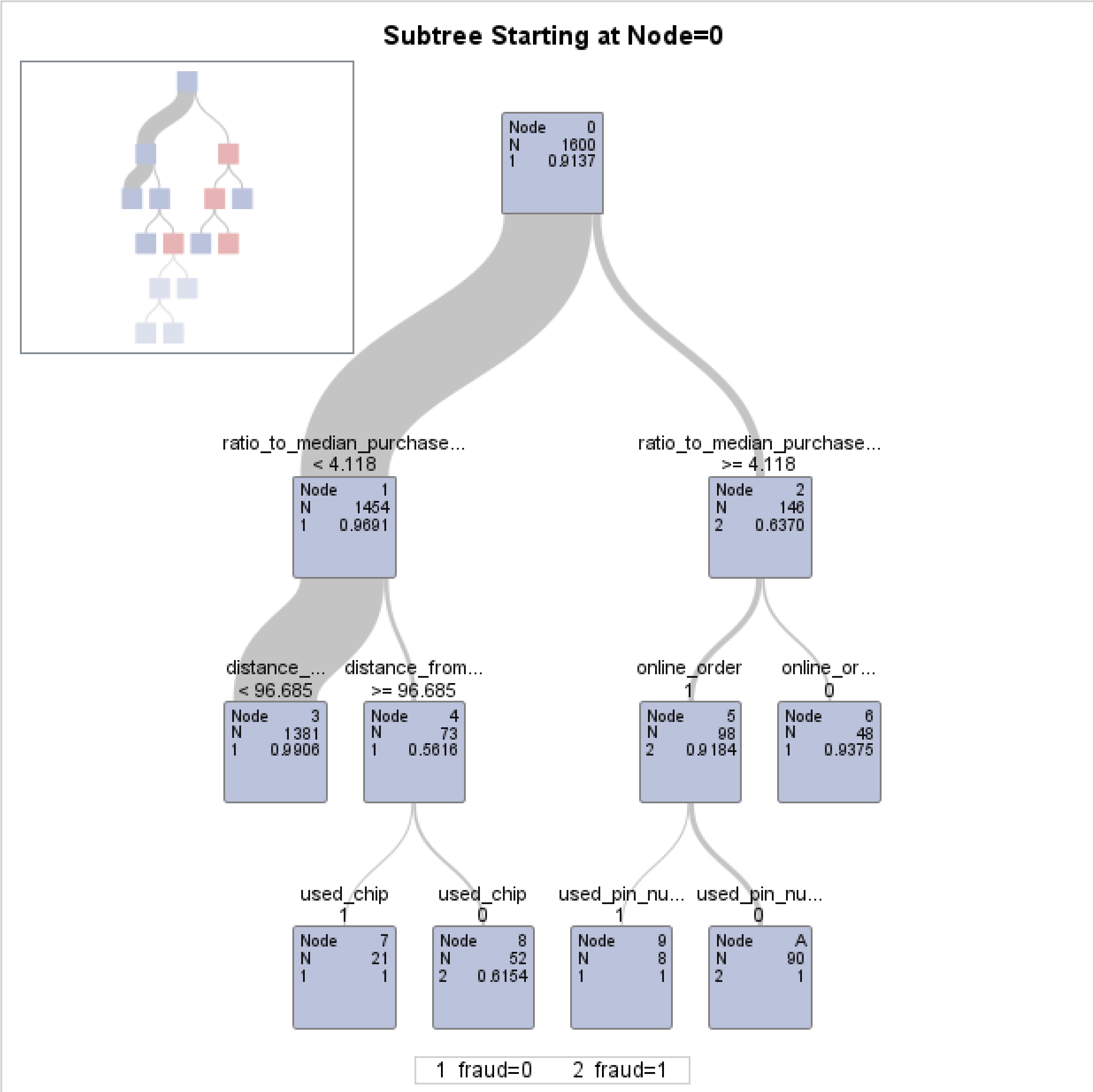
Number of Observations Read	1600
Number of Observations Used	1600

The HPSPLIT Procedure



The HPSPLIT Procedure





Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 4.118
Node 1
N 1454
1 0.9691
ratio_to_median_purchase...
>= 4.118
Node 2
N 146
2 0.6370
distance_...
< 96.685
Node 3
N 1381
1 0.9906
distance_from...
>= 96.685
Node 4
N 73
1 0.5616
online_order
1
Node 5
N 98
2 0.9184
online_or...
0
Node 6
N 48
1 0.9375
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 52
2 0.6154
used_pin_nu...
1
Node 9
N 8
1 1
used_pin_nu...
0
Node A
N 90
2 1
1 fraud=0 2 fraud=1

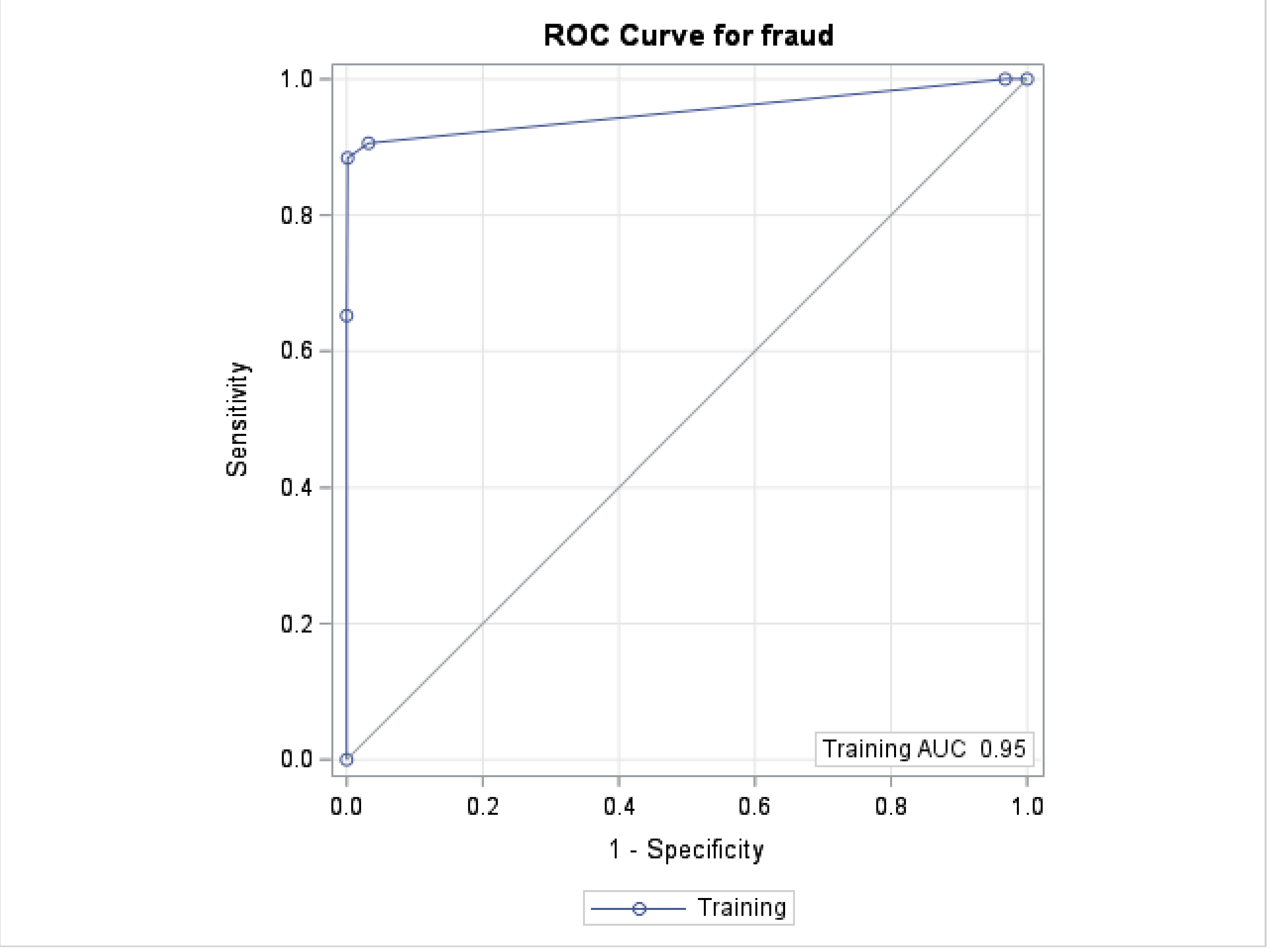
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1460	2	0.0014
1	16	122	0.1159

Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
8	0.0110	0.0113	0.8841	0.9986	0.0833	0.0220	35.1450	0.9524



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.4934	4.8713	2
used_chip	0.3410	3.3660	1

tp	fp	tn	fn	total
32	0	361	7	400

The SAS System

accuracy	misclassrate	sensitivity	FNR	specificity	FPR	precision	NPV	F1score
0.9825	0.0175	0.820513	0.179487	1	0	1	0.980978	0.901408

The SURVEYSELECT Procedure

Selection Method Simple Random Sampling

Input Data Set	CARD_DATA
Random Number Seed	122470
Sampling Rate	0.8
Sample Size	1600
Selection Probability	0.8
Sampling Weight	0
Output Data Set	CARD_DATA

The HPSPLIT Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

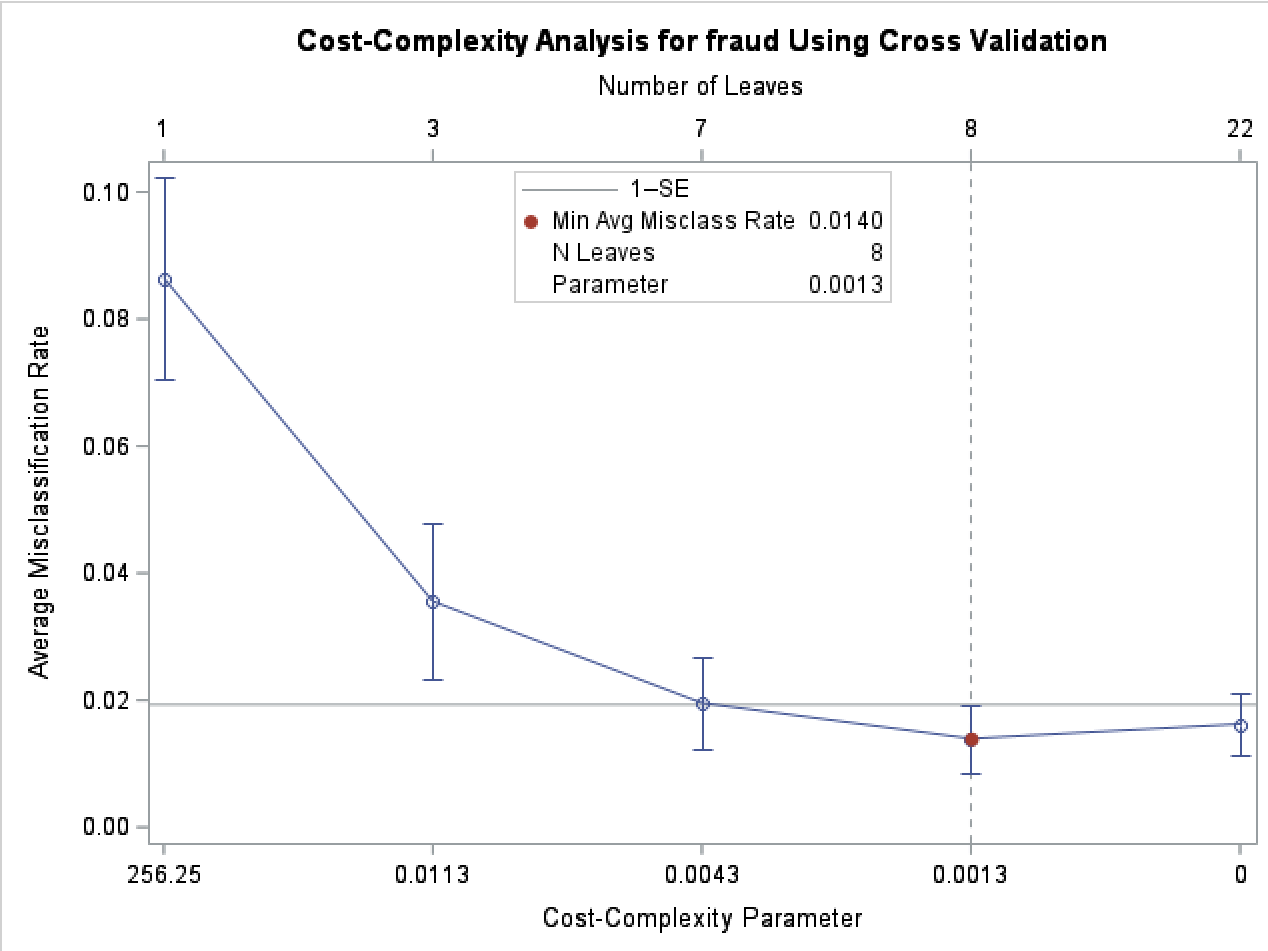
Data	Engine	Role	Path
WORK.CARD_DATA	V9	Input	On Client
WORK.PREDICTED	V9	Output	On Client

Model Information

Split Criterion Used	Gini
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	7
Maximum Tree Depth Achieved	7
Tree Depth	5
Number of Leaves Before Pruning	26
Number of Leaves After Pruning	8
Model Event Level	1

Number of Observations Read	1600
Number of Observations Used	1600

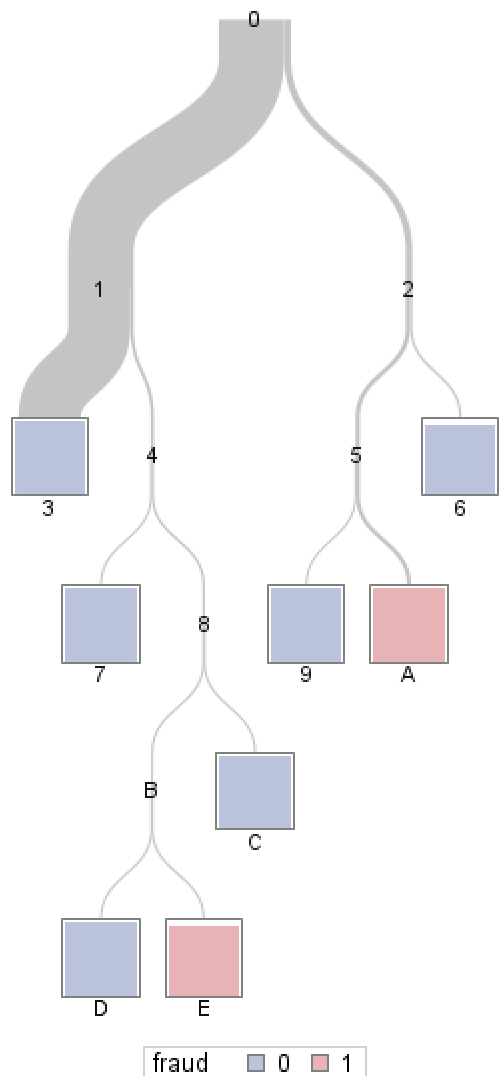
The HPSPLIT Procedure

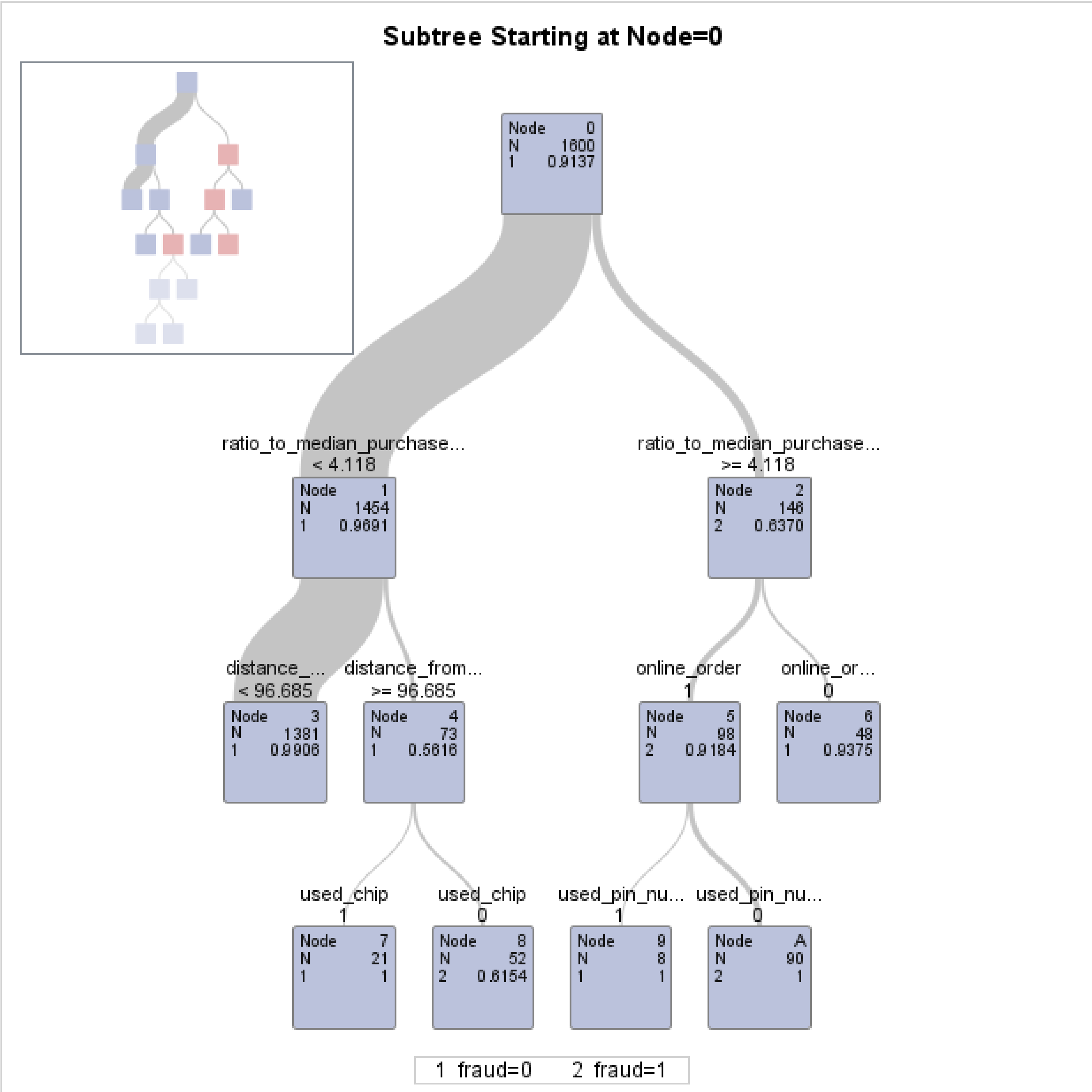


The SAS System

The HPSPLIT Procedure

Classification Tree for fraud




Subtree Starting at Node=0
Node 0
N 1600
1 0.9137
ratio_to_median_purchase...
< 4.118
Node 1
N 1454
1 0.9691
ratio_to_median_purchase...
>= 4.118
Node 2
N 146
2 0.6370
distance_...
< 96.685
Node 3
N 1381
1 0.9906
distance_from...
>= 96.685
Node 4
N 73
1 0.5616
online_order
1
Node 5
N 98
2 0.9184
online_or...
0
Node 6
N 48
1 0.9375
used_chip
1
Node 7
N 21
1 1
used_chip
0
Node 8
N 52
2 0.6154
used_pin_nu...
1
Node 9
N 8
1 1
used_pin_nu...
0
Node A
N 90
2 1
1 fraud=0
2 fraud=1

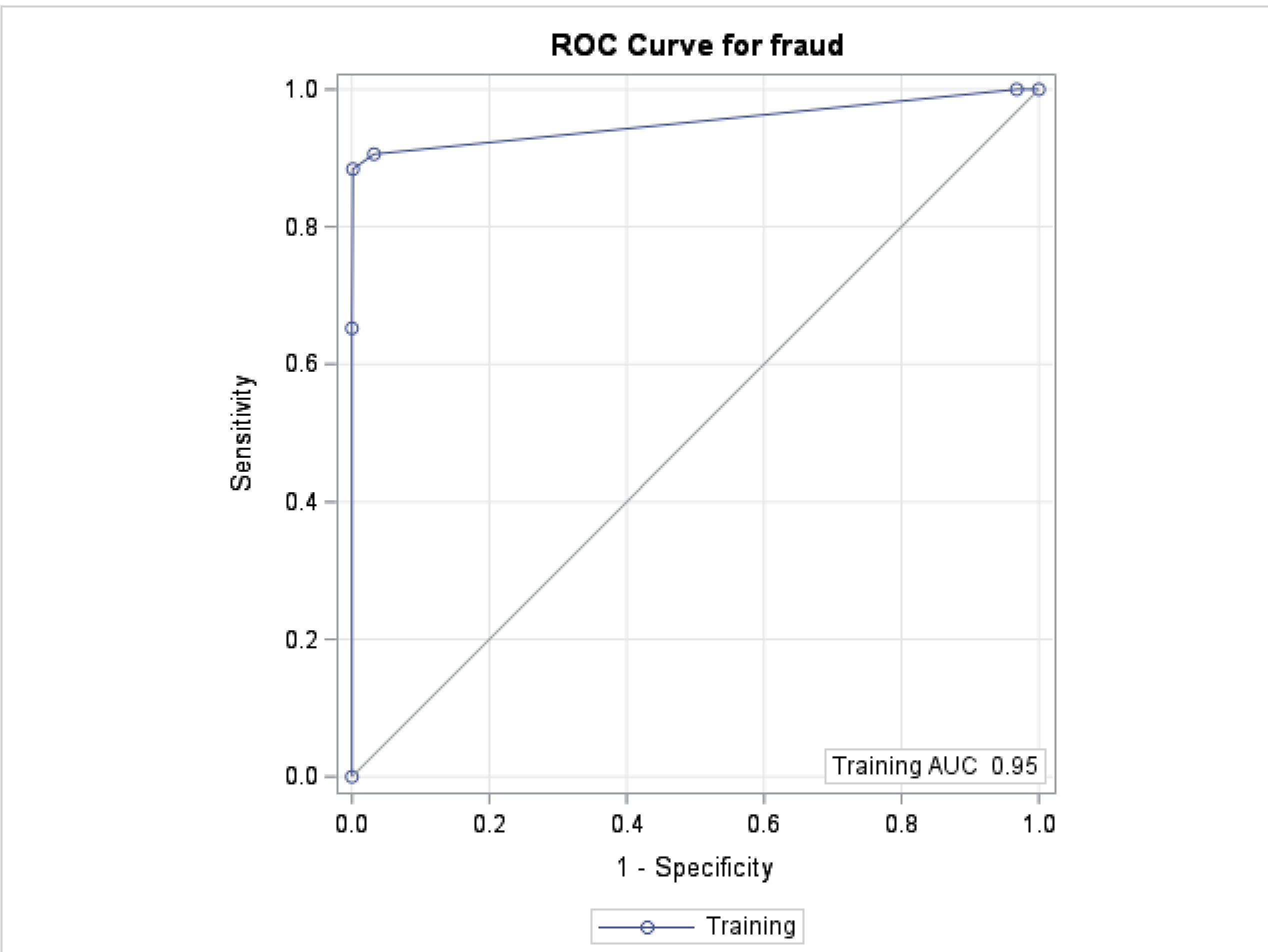
The HPSPLIT Procedure

Model-Based Confusion Matrix

Actual	Predicted 0	Predicted 1	Error Rate
0	1460	2	0.0014
1	16	122	0.1159

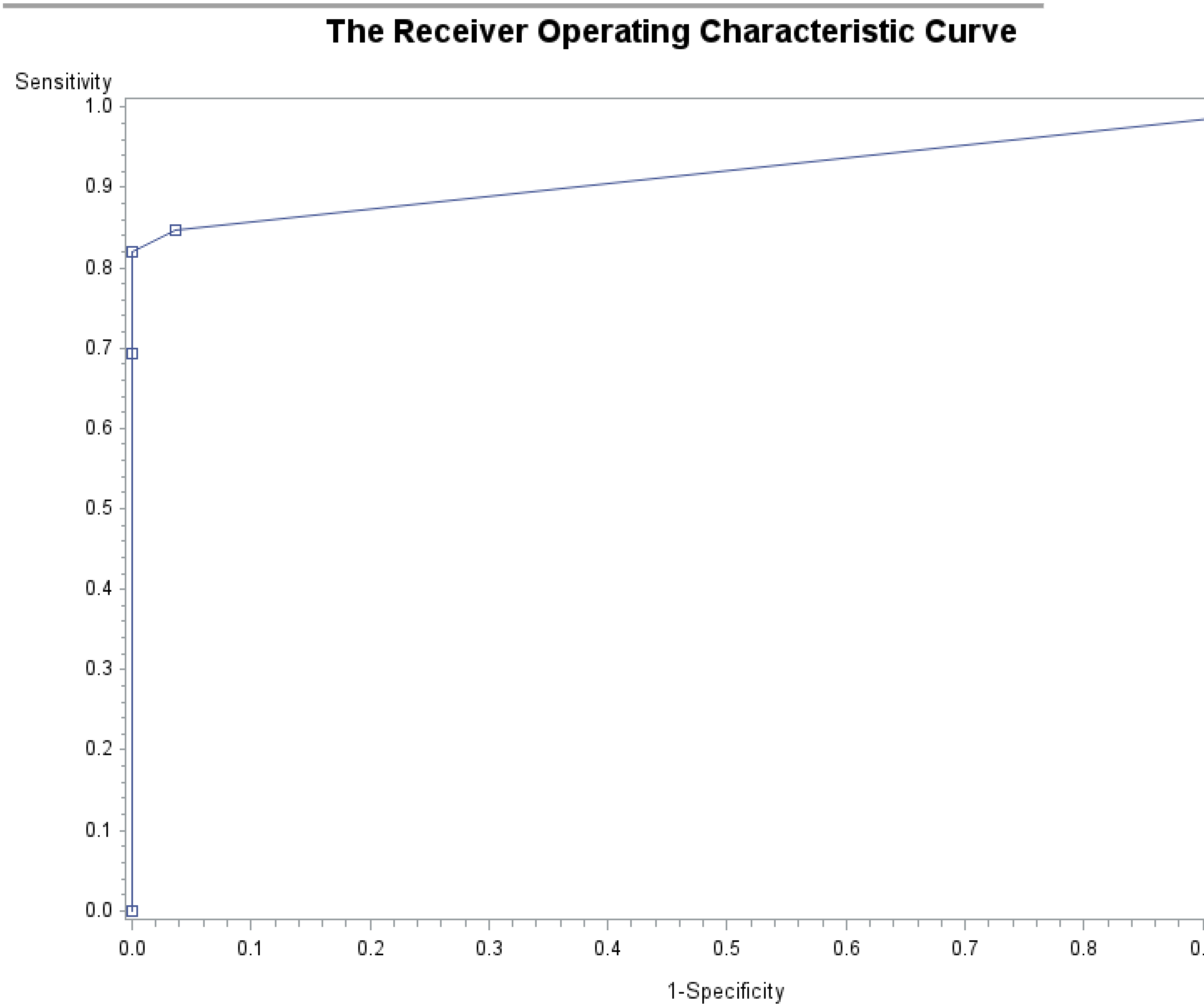
Model-Based Fit Statistics for Selected Tree

N Leaves	ASE	Mis-class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
8	0.0110	0.0113	0.8841	0.9986	0.0833	0.0220	35.1450	0.9524



Variable Importance

Variable	Training		Count
	Relative	Importance	
ratio_to_median_purchase_price	1.0000	9.8722	1
online_order	0.7782	7.6823	2
distance_from_home	0.5117	5.0512	1
used_pin_number	0.4934	4.8713	2
used_chip	0.3410	3.3660	1


The Receiver Operating Characteristic Curve
Sensitivity
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1-Specificity

The Receiver Operating Characteristic Curve

accuracy	misclassrate	sensitivity	specificity	distance	cutoff
0.9525	0.0475	0.846154	0.963989	0.158005	0.01
0.9525	0.0475	0.846154	0.963989	0.158005	0.02
0.9525	0.0475	0.846154	0.963989	0.158005	0.03
0.9525	0.0475	0.846154	0.963989	0.158005	0.04
0.9525	0.0475	0.846154	0.963989	0.158005	0.05
0.9525	0.0475	0.846154	0.963989	0.158005	0.06

The Receiver Operating Characteristic Curve

AUC

0.90507

R Code

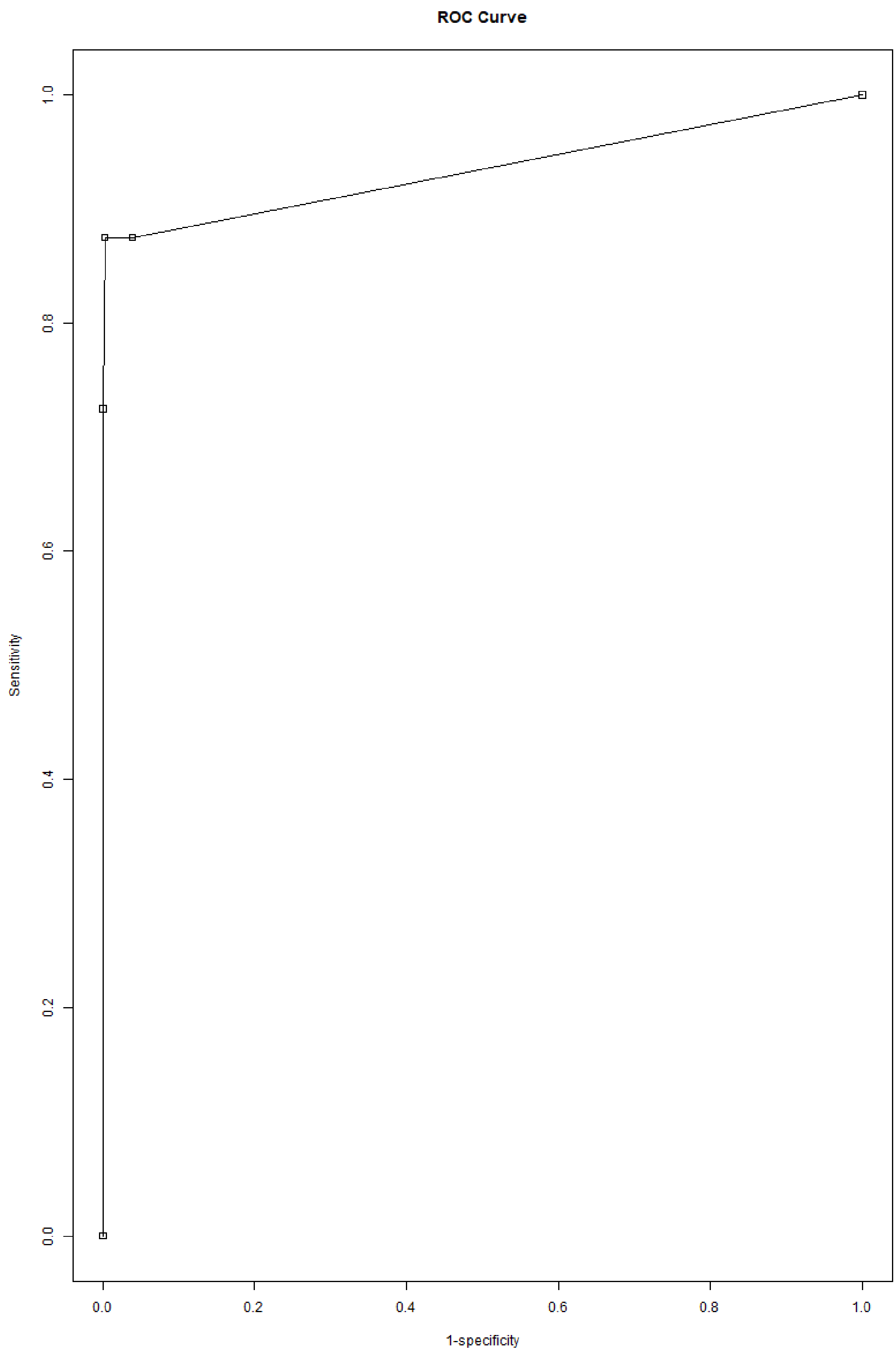
```
install.packages("Hmisc")
library(readr)
library(rpart)
library(rpart.plot)
library(dplyr)
library(partykit)
library(CHAID)
library(Hmisc)

card_data =
read.csv("C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/card_transdata.csv
",
header=T, sep=",")

# Splitting data into 80% training and 20% testing sets.

set.seed(122470)
sample = sample(c(T,F), nrow(card_data),
replace=T, prob=c(0.8, 0.2))
train = card_data[sample,]
test = card_data[!sample,]

# Fitting pruned binary tree with Gini Splitting Criterion.

tree_gini = rpart(fraud~distance_from_home+distance_from_last_transaction
+ratio_to_median_purchase_price+repeat_retailer+used_chip+used_pin_number
+online_order, data=train, method="class", parms=list(split="Gini"),
maxdepth=7)

# Computing confusion matrices and performance measures for testing set
# for a range of cutoffs.

pred_values = predict(tree_gini, test)
test = cbind(test, pred_values)
```

```
tpos = matrix(NA, nrow=nrow(test), ncol=102)
fpos = matrix(NA, nrow=nrow(test), ncol=102)
tneg = matrix(NA, nrow=nrow(test), ncol=102)
fneg = matrix(NA, nrow=nrow(test), ncol=102)

for (i in 0:101) {
    tpos[,i+1] = ifelse(test$fraud=="1" & test$"1">=0.01*i,1,0)
    fpos[,i+1] = ifelse(test$fraud=="0" & test$"1">=0.01*i,1,0)
    tneg[,i+1] = ifelse(test$fraud=="0" & test$"1"<0.01*i,1,0)
    fneg[,i+1] = ifelse(test$fraud=="1" & test$"1"<0.01*i,1,0)
}

tp = c()
fp = c()
tn = c()
fn = c()
accuracy = c()
misclassrate = c()
sensitivity = c()
specificity = c()
oneminusspec = c()
cutoff = c()

for (i in 1:102) {
    tp[i] = sum(tpos[,i])
    fp[i] = sum(fpos[,i])
    tn[i] = sum(tneg[,i])
    fn[i] = sum(fneg[,i])
    total = nrow(test)
    accuracy[i] = (tp[i]+tn[i])/total
    misclassrate[i] = (fp[i]+fn[i])/total
    sensitivity[i] = tp[i]/(tp[i]+fn[i])
    specificity[i] = tn[i]/(fp[i]+tn[i])
    oneminusspec[i] = fp[i]/(fp[i]+tn[i])
    cutoff[i] = 0.01*(i-1)
}

# Plotting ROC Curve

plot(oneminusspec, sensitivity, type="l", lty=1, main="ROC Curve",
xlab="1-specificity", ylab="Sensitivity")
points(oneminusspec, sensitivity, pch=0)

# Reporting measures for the point on the ROC Curve closest to
# the ideal point (0,1).
```

```r
distance = c()
for (i in 1:102) {
    distance[i] = sqrt(oneminusspec[i]^2+(1-sensitivity[i])^2)
}

measures = cbind(accuracy, misclassrate, sensitivity, specificity,
distance, cutoff)
min_dist = min(distance)
print(measures[which(measures[,5]==min_dist),])

# Computing the area under the ROC Curve

sensitivity = sensitivity[order(sensitivity)]
oneminusspec = oneminusspec[order(oneminusspec)]

lagx = Lag(oneminusspec, shift=1)
lagy = Lag(sensitivity, shift=1)
lagx[is.na(lagx)] = 0
lagy[is.na(lagy)] = 0
trapezoid = (oneminusspec-lagx)*(sensitivity+lagy)/2
print(AUC <- sum(trapezoid))
```

```
accuracy misclassrate sensitivity specificity  distance cutoff
 [1,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.08
 [2,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.09
 [3,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.10
 [4,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.11
 [5,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.12
 [6,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.13
 [7,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.14
 [8,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.15
 [9,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.16
[10,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.17
[11,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.18
[12,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.19
[13,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.20
[14,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.21
[15,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.22
[16,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.23
[17,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.24
[18,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.25
[19,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.26
[20,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.27
[21,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.28
[22,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.29
[23,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.30
[24,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.31
[25,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.32
[26,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.33
[27,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.34
[28,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.35
[29,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.36
[30,] 0.9850746   0.01492537       0.875   0.9972376 0.1250305    0.37
```

[31,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.38
[32,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.39
[33,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.40
[34,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.41
[35,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.42
[36,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.43
[37,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.44
[38,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.45
[39,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.46
[40,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.47
[41,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.48
[42,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.49
[43,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.50
[44,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.51
[45,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.52
[46,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.53
[47,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.54
[48,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.55
[49,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.56
[50,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.57
[51,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.58
[52,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.59
[53,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.60
[54,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.61
[55,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.62
[56,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.63
[57,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.64
[58,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.65
[59,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.66
[60,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.67
[61,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.68
[62,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.69
[63,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.70
[64,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.71
[65,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.72
[66,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.73
[67,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.74
[68,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.75
[69,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.76
[70,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.77
[71,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.78
[72,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.79
[73,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.80
[74,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.81
[75,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.82
[76,]	0.9850746	0.01492537	0.875	0.9972376	0.1250305	0.83


ROC Curve
Sensitivity
1-specificity
0.0
0.2
0.4
0.6
0.8
1.0

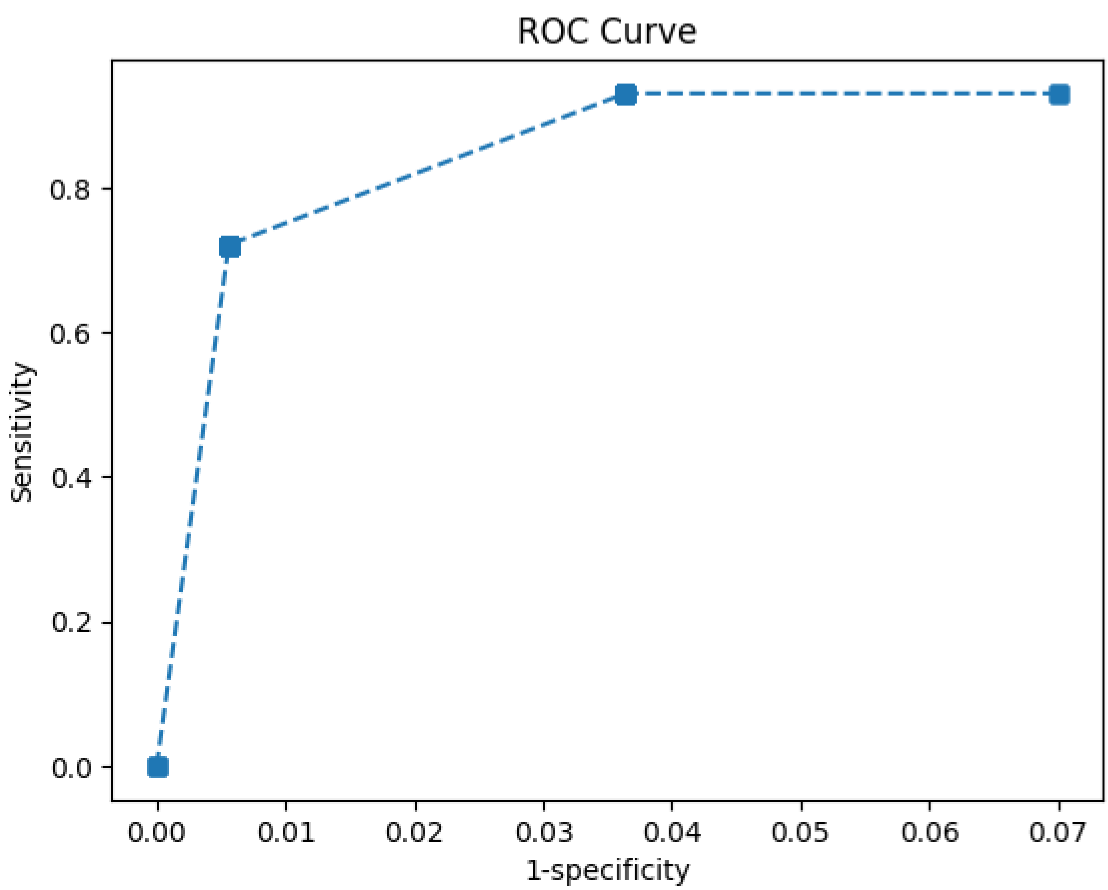
0.9348757

Python Code

```
# STAT 574 HW1 Problem 4

# Import necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.model_selection import train_test_split
# Importing the data

card_path = "C:/Users/coryg/OneDrive/Desktop/STAT_574_Data_Mining/\
card_transdata.csv"
card_data = pd.read_csv(card_path)

X = card_data.iloc[:,0:7].values
y = card_data.iloc[:,7].values

# Splitting the data into 80% training and 20% testing sets

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,
                                                    random_state=122470)

# Fitting binary tree with Gini splitting criterion.

gini_tree = DecisionTreeClassifier(max_leaf_nodes=4, criterion="gini",
                                   random_state=380381)
gini_tree_fit = gini_tree.fit(X_train, y_train)
# (a) Compute prediction accuracy, misclassification rate, sensitivity,
# specficity for a range of cutoffs between 0.01 and 0.99.

y_pred = gini_tree_fit.predict_proba(X_test)
total = len(y_pred)

cutoff = []
accuracy = []
misclassrate = []
sensitivity = []
specificity = []
oneminusspec = []
```

```python
distance = []

for i in range(99):
    tp=0
    fp=0
    tn=0
    fn=0
    cutoff.append(0.01*(i+1))
    for sub1, sub2 in zip(y_pred[::,1], y_test):
        tp_ind=1 if (sub1>0.01*(i+1) and sub2==1) else 0
        fp_ind=1 if (sub1>0.01*(i+1) and sub2==0) else 0
        tn_ind=1 if (sub1<0.01*(i+1) and sub2==0) else 0
        fn_ind=1 if (sub1<0.01*(i+1) and sub2==1) else 0
        tp+=tp_ind
        fp+=fp_ind
        tn+=tn_ind
        fn+=fn_ind
    accuracy_i = (tp+tn)/total
    misclassrate_i = (fp+fn)/total
    sensitivity_i = tp/(tp+fn)
    specificity_i = tn/(fp+tn)
    oneminusspec_i = fp/(fp+tn)
    distance_i = np.sqrt(pow(oneminusspec_i,2)+pow(1-sensitivity_i,2))

    accuracy.append(accuracy_i)
    misclassrate.append(misclassrate_i)
    sensitivity.append(sensitivity_i)
    specificity.append(specificity_i)
    oneminusspec.append(oneminusspec_i)
    distance.append(distance_i)
# (b) Construct a ROC Curve.

plt.plot(oneminusspec, sensitivity, linestyle='--', marker='s')
plt.title('ROC Curve')
plt.xlabel('1-specificity')
plt.ylabel('Sensitivity')
# (c) Compute the minimal distance between the ROC Curve and the ideal
# point (0,1) and output accuracy, misclassification rate, sensitivity,
# specificity, and cutoff that corresponds to the minimal distance.

df = pd.DataFrame({'accuracy':accuracy, 'misclassrate':misclassrate,
'sensitivity':sensitivity,
                   'specificity':specificity, 'oneminusspec':oneminusspec,
'distance':distance,
                   'cutoff':cutoff})
```

```
min_distance = min(distance)
optimal = df[df['distance'] == min_distance]
print(optimal)
# (d) Compute the area under the ROC Curve

df = df.sort_values('oneminusspec', ascending=True)
df['lagx'] = df['oneminusspec'].shift(1)
df['lagy'] = df['sensitivity'].shift(1)
df['lagx'] = np.nan_to_num(df['lagx'], nan=0)
df['lagy'] = np.nan_to_num(df['lagy'], nan=0)
df['trapezoid'] = ((df['oneminusspec']-
df['lagx'])*(df['sensitivity']+df['lagy']))/2
AUC = 1 - sum(df['trapezoid'])
print(AUC)
```



Minimal distances between ROC Curve and ideal point (0,1), along with accuracy, misclassification rate, sensitivity, specificity, and cutoff that corresponds to the minimal distance:

```
   accuracy  misclassrate  sensitivity  specificity  oneminusspec  distance  \
7      0.96          0.04     0.930233     0.963585      0.036415  0.078699
```

8	0.96	0.04	0.930233	0.963585	0.036415	0.078699
9	0.96	0.04	0.930233	0.963585	0.036415	0.078699
10	0.96	0.04	0.930233	0.963585	0.036415	0.078699
11	0.96	0.04	0.930233	0.963585	0.036415	0.078699
12	0.96	0.04	0.930233	0.963585	0.036415	0.078699
13	0.96	0.04	0.930233	0.963585	0.036415	0.078699
14	0.96	0.04	0.930233	0.963585	0.036415	0.078699
15	0.96	0.04	0.930233	0.963585	0.036415	0.078699
16	0.96	0.04	0.930233	0.963585	0.036415	0.078699
17	0.96	0.04	0.930233	0.963585	0.036415	0.078699
18	0.96	0.04	0.930233	0.963585	0.036415	0.078699
19	0.96	0.04	0.930233	0.963585	0.036415	0.078699
20	0.96	0.04	0.930233	0.963585	0.036415	0.078699
21	0.96	0.04	0.930233	0.963585	0.036415	0.078699
22	0.96	0.04	0.930233	0.963585	0.036415	0.078699
23	0.96	0.04	0.930233	0.963585	0.036415	0.078699
24	0.96	0.04	0.930233	0.963585	0.036415	0.078699
25	0.96	0.04	0.930233	0.963585	0.036415	0.078699
26	0.96	0.04	0.930233	0.963585	0.036415	0.078699
27	0.96	0.04	0.930233	0.963585	0.036415	0.078699
28	0.96	0.04	0.930233	0.963585	0.036415	0.078699
29	0.96	0.04	0.930233	0.963585	0.036415	0.078699
30	0.96	0.04	0.930233	0.963585	0.036415	0.078699
31	0.96	0.04	0.930233	0.963585	0.036415	0.078699
32	0.96	0.04	0.930233	0.963585	0.036415	0.078699
33	0.96	0.04	0.930233	0.963585	0.036415	0.078699
34	0.96	0.04	0.930233	0.963585	0.036415	0.078699
35	0.96	0.04	0.930233	0.963585	0.036415	0.078699

36	0.96	0.04	0.930233	0.963585	0.036415	0.078699
37	0.96	0.04	0.930233	0.963585	0.036415	0.078699
38	0.96	0.04	0.930233	0.963585	0.036415	0.078699
39	0.96	0.04	0.930233	0.963585	0.036415	0.078699
40	0.96	0.04	0.930233	0.963585	0.036415	0.078699
41	0.96	0.04	0.930233	0.963585	0.036415	0.078699
42	0.96	0.04	0.930233	0.963585	0.036415	0.078699
43	0.96	0.04	0.930233	0.963585	0.036415	0.078699
44	0.96	0.04	0.930233	0.963585	0.036415	0.078699
45	0.96	0.04	0.930233	0.963585	0.036415	0.078699

	cutoff
7	0.08
8	0.09
9	0.10
10	0.11
11	0.12
12	0.13
13	0.14
14	0.15
15	0.16
16	0.17
17	0.18
18	0.19
19	0.20
20	0.21
21	0.22
22	0.23

23	0.24
24	0.25
25	0.26
26	0.27
27	0.28
28	0.29
29	0.30
30	0.31
31	0.32
32	0.33
33	0.34
34	0.35
35	0.36
36	0.37
37	0.38
38	0.39
39	0.40
40	0.41
41	0.42
42	0.43
43	0.44
44	0.45
45	0.46

AUC

0.9412741840922415