Subject امیرمهدی انصاری پور (سوالات فصل ۹) 810198358
Date

1) سوال فقط

Occupation → Service : 000
→ Managment : 001
→ Sales : 010
→ Staff : 100

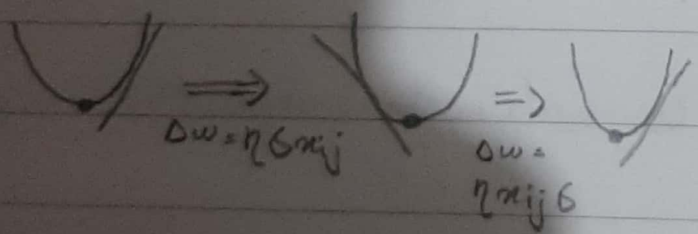
Indicator برای متغیر

occupation را خواسته (دیگر متغیرها نیازی نیست)

10) learning rate (یا η) بالا: باعث می‌شود که در حوالی

نقطه optimum، به چپ و راست این نقطه نوسان کنیم و هیچوقت به

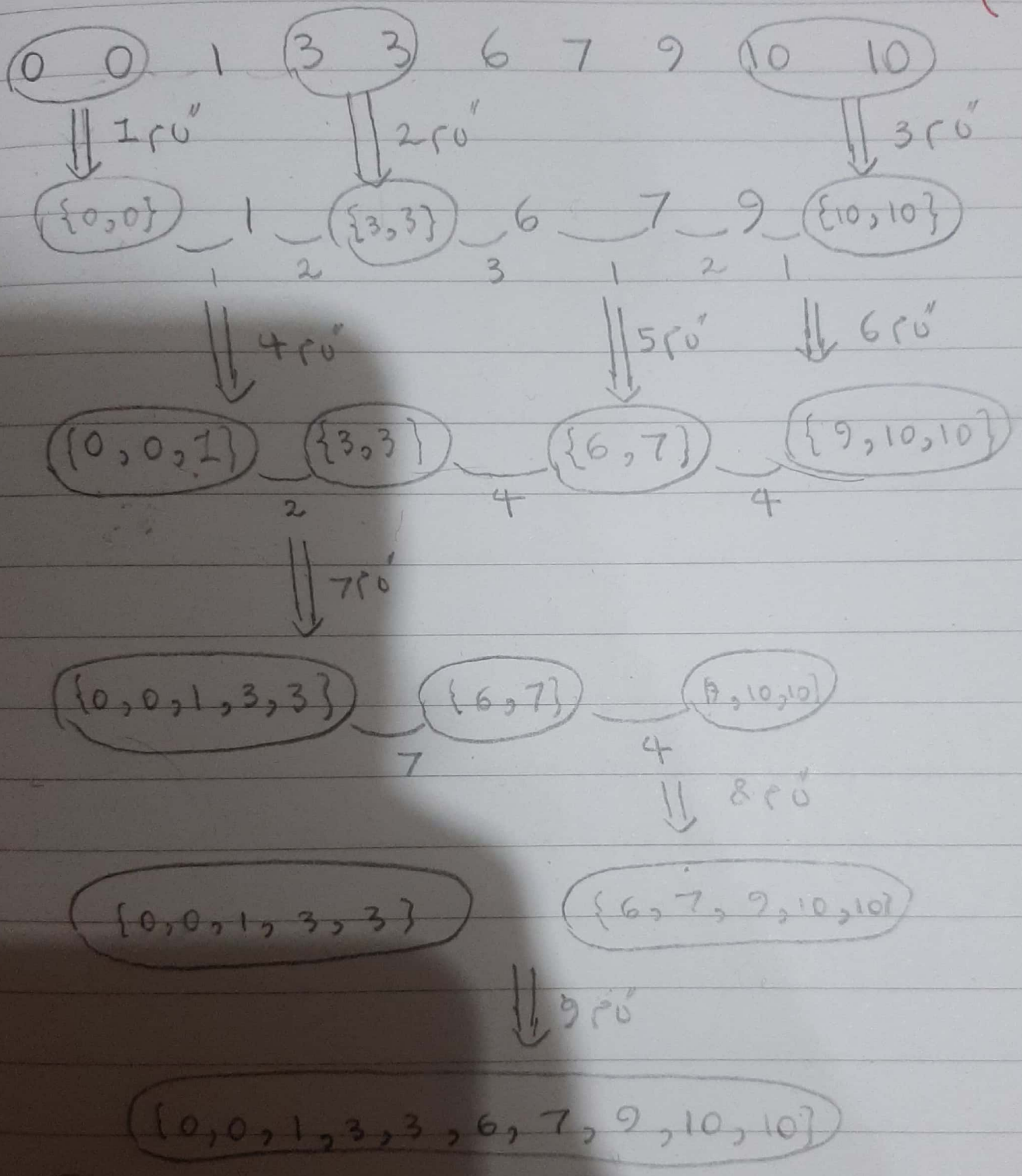
نقطه optimum converge نکنیم



8101983358 امیرعلی اسفاری پور (سوالات فصل 10)

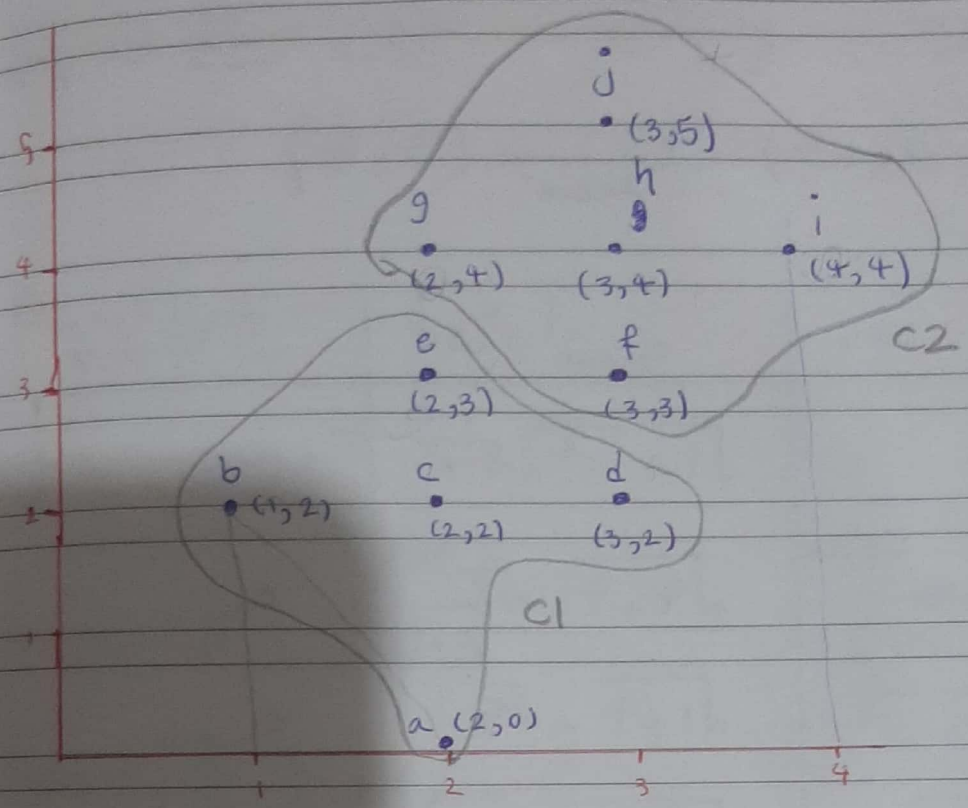
Subject Date

(4



(ادامه سوالات فصل ۱۰)

(6



مرکز نقل‌های اولیه را b و i قرار می‌دهیم.

	فاصله از b	فاصله از i	مرکز نقل انتخاب شده
a	$\sqrt{5}$	$\sqrt{16+4}=\sqrt{20}$	b
c	۱	$\sqrt{8}$	b
d	2	$\sqrt{4+1}=\sqrt{5}$	b
e	$\sqrt{2}$	$\sqrt{4+1}=\sqrt{5}$	b
f	$\sqrt{5}$	$\sqrt{2}$	i
g	$\sqrt{5}$	2	i
h	$\sqrt{8}$	۱	i
j	$\sqrt{13}$	$\sqrt{2}$	i

s.a.m

مراکز ثقل جدید :

$c_{1_{new}} = (2, 1.8)$ $c_{2new} = (3, 4)$

	فاصله از c_{1new}	فاصله از c_{2new}	مرکز ثقل انتخاب شده
(3,0) a	1.8	$\sqrt{1+16} = \sqrt{17}$	c_{1new}
(1,2) b	$\sqrt{1+(0.2)^2}$	$\sqrt{4+4} = \sqrt{8}$	c_{1new}
(2,2) c	0.2	$\sqrt{1+4} = \sqrt{5}$	c_{1new}
(3,2) d	$\sqrt{1+(0.2)^2}$	$\sqrt{4} = 2$	c_{1new}
(2,3) e	1.2	$\sqrt{1+1} = \sqrt{2} = 1.4$	c_{1new}
(3,3) f	$\sqrt{1+(1.2)^2}$	1	c_{2new}
(2,4) g	2.2	1	c_{2new}
(3,4) h	$\sqrt{1+(2.2)^2}$	0	c_{2new}
(4,4) i	$\sqrt{4+(2.2)^2}$	1	c_{2new}
j	$\sqrt{1+(3.2)^2}$	1	c_{2new}

s.a.m

0.14 / 0.20

همانطور که دیده شد در مرحله دوم خوشه‌بندی clustering

الگوریتم با خوشه‌های نیز K) میشود،

$C_1 = (a, b, c, d, e)$

$C_2 = (h, g, f, i, j)$

	step 1	step 2
BCV :	$\sqrt{13}$	$\sqrt{1 + (2.2)^2} = \sqrt{5.84}$
WCV :	$\sqrt{43 + 47} = \sqrt{90}$	$\sqrt{31.36 + 40} = \sqrt{71.36}$
BCV / WCV :	0.37	0.28

$WCV = \sum_i \sum_i d(p, m_i)^2$

همانطور که در مرحله دوم نسبت سال ذخیره شده نسبت ratio

کاهش می‌یابد

(سوالات فصل 12)

(4 دوباره جدول بالا رو بکشید:

outlook	Temper	Humidity	Windy	Play
S	H	H	F	No
S	H	H	T	No
R	C	N	T	No
S	M	H	F	No
R	M	H	T	No
O	H	H	F	Yes
R	M	H	F	Yes
R	C	N	F	Yes
O	C	N	T	Yes
S	C	N	F	Yes
R	M	N	F	Yes
S	M	N	T	Yes
O	M	H	T	Yes
O	H	N	F	Yes

s.a.m

F1 = { { outlook = S }, { Temperature = H },
{ " = R } { " = C }
{ " = O } { " = M }

{Hum = H }, {Windy = T }, { Play = Yes } }
{ " = N } { " = F } { Play = No }

F2 = { { outlook = S, Humidity = H }, (5
{ outlook = S, windy = F },
{ " = S, Play = No },
{ outlook = R, Temperature = M },
{ " = R, Humidity = N },
{ " = R, Windy = T },
{ " = R, Play = Yes },
{ " = O, Play = Yes },
{ Temper = H, Humity = H },
{ " = H, Windy = F },
{ " = C, Humidity = N },
{ " = C, Play = Yes },
{ Temper = M, Humty = H },
{ " = M, Windy = T },

s.a.m

{ Temper = M, windy = F },

{ " = M, Play = Yes } ,

{ Humid = H, windy = F },

{ " = H, " = T },

{ " = N, " = T },

{ " = N, " = F },

{ " = H, Play = No },

{ " = H, Play = Yes },

{ " = N, Play = Yes },

{ windy = T, Play = No },

{ " = T, Play = Yes },

{ " = F, Play = Yes } }

$F_3 = \{$ {outlook = S, Humidity = H, Play = No}, (6
{outlook = R, windy = F, Play = Yes},
{Temper = C, Humidity = N, Play = Yes},
{Humidity = N, windy = F, Play = Yes} $\}$

(7

outlook = S => Play = No sup = 3/14
X conf = 3/5 = 60%
↓
P(No|S)

outlook = R => Play = Yes sup = 3/14
X confidence = 3/5 X
↓
P(Yes|R)

outlook = O => Play = Yes sup = 4/14 ✓
✓ conf = 1 ✓
P(Yes|O)

Temperature = C => Play = Yes sup = 3/14 ✓
✓ conf = 3/4 ✓
P(Yes|C)

s.a.m

Temperature = M => Play = Yes sup = 4/14
conf = 4/6 X

X

Humidity = H => Play = No sup = 4/14
conf = 4/7 X
P(No|H)

X

Humidity = H => Play = Yes sup = 3/14
conf = 3/7 X
P(Yes|H)

X

Humidity = N => Play = Yes sup = 6/14
conf = 6/7

✓

windy = T => Play = No sup = 3/14
conf = 3/6 X

X

windy = T => Play = Yes sup = 3/14
conf = 3/6 X

X

windy = F => Play = Yes sup = 6/14
conf = 6/8

✓

outlook = S, Humidity = H => Play = No (8

✓ sup = 3/14
confidence = 1

outlook = R, windy = F => Play = Yes

✓ sup = 3/14

P(Yes | out = R, windy = F) — conf = 1

Temper = C, Humidity = N => Play = Yes

✓ sup = 3/14
conf = 3/4

Humidity = N, windy = F => Play = Yes

✓ sup = 4/14
conf = 1

rule	sup x confidence
	(9
Humidity = N => Play = Yes	36/98
windy = F => Play = Yes	36/112
outlook = O => Play = Yes	4/14
Humidity = N, windy = F => Play = Yes	4/14
outlook = S, Humidity = H => Play = No	3/14
outlook = R, windy = F => Play = Yes	3/14
Temp = C => Play = Yes	9/56
Temp = C, Humidity = N => Play = Yes	9/56

$$J = P(x)\left[P(y|x)\ln\left(\frac{P(y|x)}{P(y)}\right) + (1-P(y|x))\ln\left(\frac{1-P(y|x)}{1-P(y)}\right)\right]$$ (1)

1) outlook = O => Play = Yes $\boxed{J = 0.75}$

$P(x) = 4/14$ $P(y|x) = 4/14$ $P(y) = 9/14$

2) outlook = S, Humidity = H => Play = No $\boxed{J = 0.22}$

$P(x) = 3/14$ $P(y|x) = 1$ $P(y) = 5/14$

3) Humidity = N, windy = F => Play = Yes $\boxed{J = 0.12}$

$P(x) = 4/14$ $P(y|x) = 1$ $P(y) = 9/14$

4) outlook = R, windy = F => Play = Yes $\boxed{J = 0.09}$

$P(x) = 3/14$ $P(y|x) = 1$ $P(y) = 9/14$

5) Humidity = N => Play = Yes $\boxed{J = 0.05}$

$P(x) = 1/2$ $P(y|x) = 6/7$ $P(y) = 9/14$

6) windy = F => Play = Yes $\boxed{J = 0.015}$

$P(x) = 8/14$ $P(y|x) = 6/8$ $P(y) = 9/14$

7) Temp = C => Play = Yes $\boxed{J = 0.007}$

$P(x) = 4/14$ $P(y|x) = 3/14$ $P(y) = 9/14$

s.a.m

8) Temp = C, Humidity = N => Play = Yes

$P(x) = 4/14$ $P(y|x) = 3/4$ $P(y) = 9/14$

$J = 0.007$

(12

پیش از برازش شبکه عصبی، ابتدا باید ورودی های را پردازش کنیم:

```
data_set$State = factor(data_set$State)
levels(data_set$State)
data_set$Area.Code = factor(data_set$Area.Code)
levels(data_set$Area.Code)
levels(data_set$Area.Code) = c("1", "2", "3") #need to collapse
levels(data_set$Area.Code)
data_set$Int.l.Plan = factor(data_set$Int.l.Plan)
data_set$VMail.Plan = factor(data_set$VMail.Plan)
data_set$Churn. = factor(data_set$Churn.)

##Indicator for categorical variables
newAreaCode = class.ind(data_set$Area.Code)
newdataSet = cbind(data_set, newAreaCode)[, -c(4, 25)]
newdataSet = newdataSet[, -2]  #Omit state attribute (independent from target)
colnames(newdataSet)[c(21, 22)] = c("AreaCode1", "AreaCode2")


newIntPlan = class.ind(data_set$Int.l.Plan)
newdataSet = cbind(newdataSet, newIntPlan)[, -c(4, 23)]
colnames(newdataSet)[22] = "Int.l.Plan"

newVmailPlan = class.ind(data_set$VMail.Plan)
newdataSet = cbind(newdataSet, newVmailPlan)[, -c(4, 23)]
colnames(newdataSet)[22] = "Vmail.Plan"

newChurn = class.ind(data_set$Churn.)
newdataSet = cbind(newdataSet, newChurn)[, -c(18, 23)]
colnames(newdataSet)[22] = "Churn."

newdataSet = newdataSet[, -c(1, 3)] # omit phone and X columns
newdataSet = newdataSet[, -c(16, 17)] # Omit AreaCode1 and 2
```

متغیر کیفی AreaCode و متغیر های باینری Vmail.plan Int.l.Plan را باید بصورت dummy دربیاوریم. همچنین متغیر هدف churn را نیز باید بصورت dummy تبدیل کنیم زیرا خروجی شبکه عصبی عددی بین صفر و یک است.

دقت شود که با تبدیل به dummy، ستون متغیر کیفی قبلی و یکی از ستون های dummy را حذف میکنیم. (k - 1 indicator کافیست)

سپس متغیر های کمی را normalize میکنیم. دقت شود که جفت متغیرهای با correlation = 1 را نباید به الگوریتم دهیم، و متغیرهای Mins و Charge با هم ارتباط مستقیم دارند. پس از میان هر جفت Day/Night/Eve/Intl Mins و

Charge یکی را انتخاب میکنیم.

```
normalize = function(a){
  return ((a - min(a)) / (max(a) - min(a)));
}

#colSums(is.na(newdataSet))

# Normalize numericals
(cor_res = cor.test(newdataSet$Day.Charge, newdataSet$Day.Mins, method = "pearson"))
(cor_res = cor.test(newdataSet$Eve.Charge, newdataSet$Eve.Mins, method = "pearson"))
(cor_res = cor.test(newdataSet$Night.Charge, newdataSet$Night.Mins, method = "pearson"))
(cor_res = cor.test(newdataSet$Intl.Charge, newdataSet$Intl.Mins, method = "pearson"))

newdataSet$Account.Length = normalize(newdataSet$Account.Length)
newdataSet$VMail.Message = normalize(newdataSet$VMail.Message)
newdataSet$Day.Mins = normalize(newdataSet$Day.Mins)
newdataSet$Day.Calls = normalize(newdataSet$Day.Calls)
newdataSet$Eve.Mins = normalize(newdataSet$Eve.Mins)
newdataSet$Eve.Calls = normalize(newdataSet$Eve.Calls)
newdataSet$Night.Mins = normalize(newdataSet$Night.Mins)
newdataSet$Night.Calls = normalize(newdataSet$Night.Calls)
newdataSet$Intl.Mins = normalize(newdataSet$Intl.Mins)
newdataSet$Intl.Calls = normalize(newdataSet$Intl.Calls)
newdataSet$CustServ.Calls = normalize(newdataSet$CustServ.Calls)

newdataSet = newdataSet[,-c(5, 8, 11, 14)] #Omit Day/Night/Eve/Int Charges

#sapply(c("Int.l.Plan", "VMail.Plan"),
#       function(i) newdataSet[,i] =  as.numeric(newdataSet[,i]))

str(newdataSet)
```

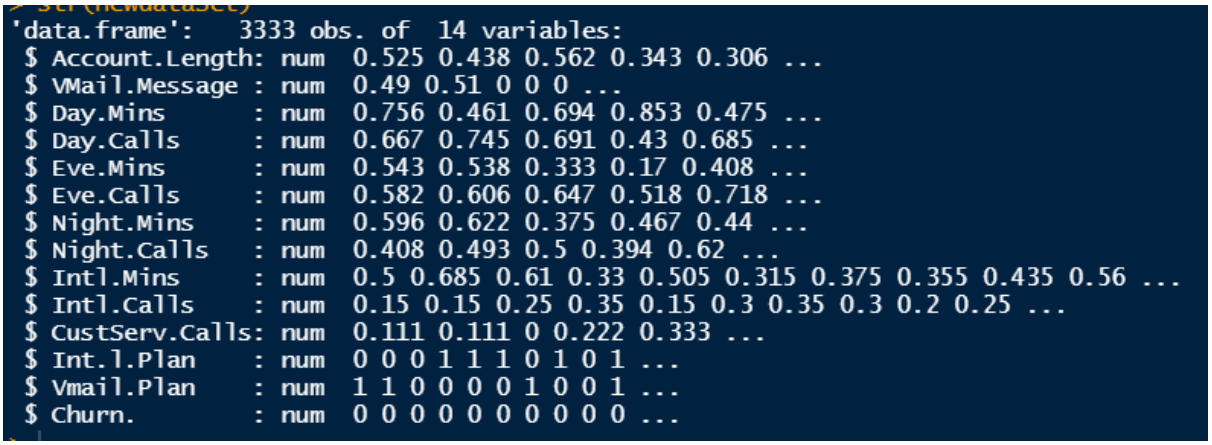
	Account.Length	VMail.Message	Day.Mins	Day.Calls	Eve.Mins	Eve.Calls	Night.Mins	Night.Calls
1	0.52479339	0.4901961	0.7557013	0.6666667	0.54275502	0.5823529	0.5957504	0.4084507
2	0.43801653	0.5098039	0.4606613	0.7454545	0.53753093	0.6058824	0.6218397	0.4929577
3	0.56198347	0.0000000	0.6938426	0.6909091	0.33324168	0.6470588	0.3749328	0.5000000
4	0.34297521	0.0000000	0.8534778	0.4303030	0.17019522	0.5176471	0.4671867	0.3943662
5	0.30578512	0.0000000	0.4751995	0.6848485	0.40775364	0.7176471	0.4402905	0.6197183
6	0.48347107	0.0000000	0.6368301	0.5939394	0.60654385	0.5941176	0.4860140	0.5985915
7	0.49586777	0.4705882	0.6220068	0.5333333	0.95820731	0.6352941	0.5094137	0.5985915
8	0.60330579	0.0000000	0.4475485	0.4787879	0.28347539	0.5529412	0.5072620	0.4436620
9	0.47933884	0.0000000	0.5259407	0.5878788	0.96673082	0.4705882	0.5180204	0.4014085
10	0.57851240	0.7254902	0.7371722	0.5090909	0.61039318	0.6529412	0.8154922	0.4507042
11	0.26446281	0.0000000	0.3680160	0.8303030	0.62826505	0.4882353	0.4991931	0.5492958
12	0.30165289	0.0000000	0.5350627	0.7696970	0.44927138	0.8705882	0.4647660	0.4295775
13	0.69008264	0.0000000	0.3671608	0.5818182	0.28842453	0.4176471	0.3171060	0.6690141
14	0.38842975	0.0000000	0.4464082	0.5333333	0.68078086	0.4411765	0.4548144	0.5774648
15	0.25206612	0.0000000	0.3440707	0.4242424	0.84465219	0.4470588	0.4835933	0.4647887
16	0.66115702	0.0000000	0.9489738	0.4060606	0.87379709	0.5705882	0.3695535	0.6690141
17	0.34710744	0.5294118	0.5598632	0.8424242	0.77233984	0.5294118	0.1777838	0.2957746
18	0.38016529	0.0000000	0.5436146	0.6909091	0.59994501	0.6529412	0.2861754	0.6197183
19	0.30991736	0.6470588	0.5407640	0.4000000	0.58509761	0.3823529	0.3832706	0.5281690
20	0.29752066	0.0000000	0.6396807	0.5454545	0.43854825	0.5176471	0.4561592	0.2887324

همانطور که دیده می شود، متغیر های عددی normalize شده اند.

CustServ.Calls	AreaCode1	AreaCode2	Int.l.Plan	Vmail.Plan	Churn.
0.1111111	0	1	0	1	0
0.1111111	0	1	0	1	0
0.0000000	0	1	0	0	0
0.2222222	1	0	1	0	0
0.3333333	0	1	1	0	0
0.0000000	0	0	1	0	0
0.3333333	0	0	0	1	0
0.0000000	0	1	1	0	0
0.1111111	1	0	0	0	0
0.0000000	0	1	1	1	0
0.4444444	0	1	0	0	1
0.0000000	0	1	0	0	0
0.1111111	1	0	0	0	0
0.3333333	0	0	0	0	0
0.4444444	0	1	0	0	0
0.4444444	0	1	0	0	1
0.1111111	1	0	0	1	0
0.3333333	0	0	0	0	0
0.1111111	0	0	0	1	0
0.1111111	0	1	0	0	0
0.0000000	0	1	0	0	0
0.5555556	1	0	0	0	1
0.0000000	0	1	0	0	0
0.2222222	0	1	0	0	0
0.0000000	0	0	0	0	0
0.3333333	0	1	0	0	0

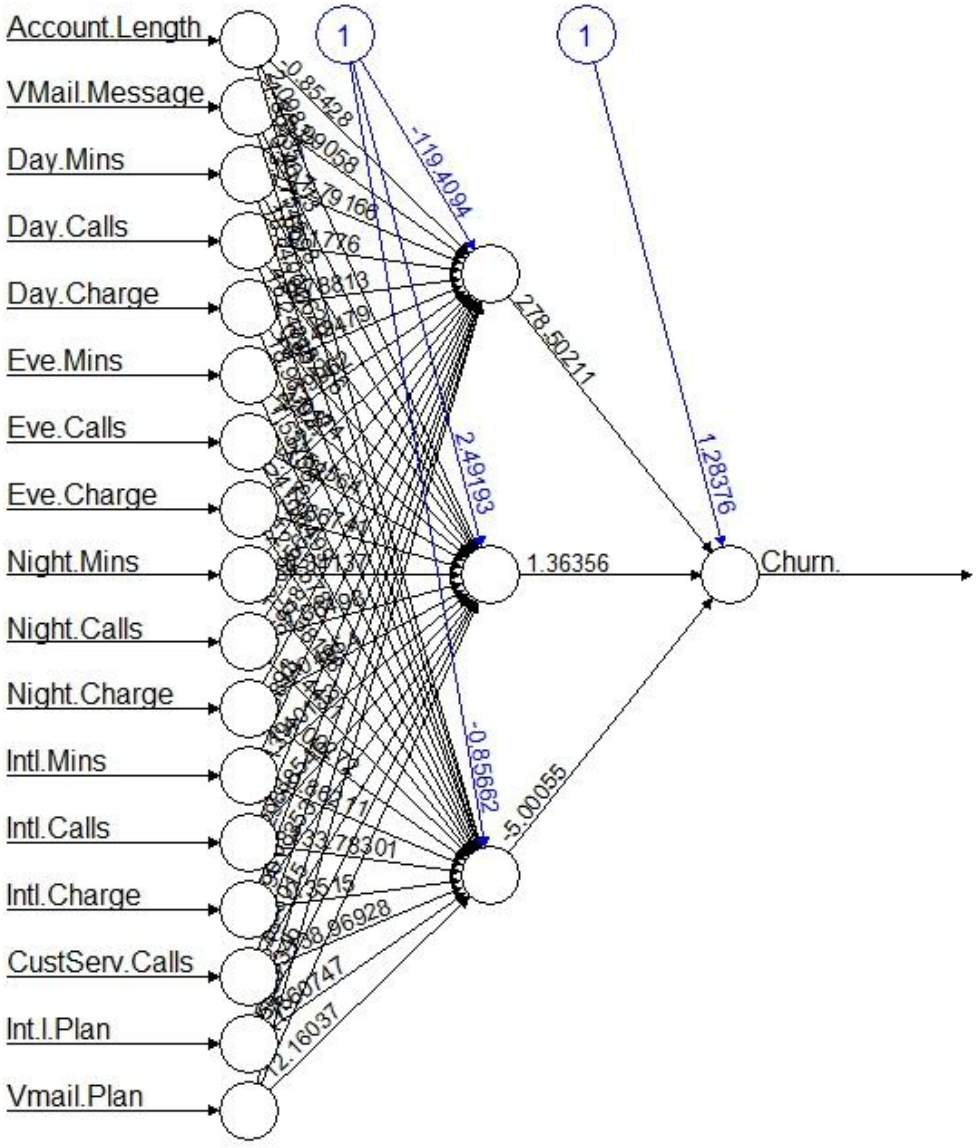
همانطور که دیده می شود، متغیرهای دودویی و کیفی نیز برایشان dummy variable تعریف شده.

با ران کردن str(newdataSet داریم:



تمامی متغیرهای ورودی شبکه عصبی numeric شده اند (factor نداریم)
دقت شود متغیر AreaCode از Churn مستقل است (از بخش EDA میدانیم) پس ورودی شبکه عصبی نمی باشد.

حال formula شبکه عصبی را تشکیل می دهیم و شبکه عصبی را برازش میدهیم:
(churn ستون 14 ام dataset می باشد)


Account.Length
VMail.Message
Day.Mins
Day.Calls
Day.Charge
Eve.Mins
Eve.Calls
Eve.Charge
Night.Mins
Night.Calls
Night.Charge
Intl.Mins
Intl.Calls
Intl.Charge
CustServ.Calls
Int.l.Plan
Vmail.Plan
1
1
-119.4094
278.50211
1.28376
2.49193
1.36356
-0.85662
-5.00055
Churn.

```r
right = paste0(colnames(newdataSet)[-18] , collapse = "+")
formula = paste0(colnames(newdataSet)[18] , "~" , right)
formula = as.formula(formula)

net.dat <- neuralnet(formula,
                     data = newdataSet,
                     rep = 1,
                     lifesign = "full",
                     hidden = 3,
                     linear.output=FALSE)

(net.dat$net.result[[1]])

(mean((newdataSet[,18] - net.dat$net.result[[1]])^2))
```

(Top Level)

Terminal | Background Jobs

1.2.2 · ~/

```
73000 min thresh: 0.0119897818073295
74000 min thresh: 0.0118719942281751
75000 min thresh: 0.0118719942281751
76000 min thresh: 0.0118719942281751
77000 min thresh: 0.0117035314431611
78000 min thresh: 0.0117035314431611
79000 min thresh: 0.0117035314431611
80000 min thresh: 0.0110702574498792
81000 min thresh: 0.0110702574498792
82000 min thresh: 0.0110702574498792
83000 min thresh: 0.0110702574498792
84000 min thresh: 0.0110702574498792
85000 min thresh: 0.0110702574498792
86000 min thresh: 0.0104628971337711
87000 min thresh: 0.0104628971337711
88000 min thresh: 0.0104628971337711
89000 min thresh: 0.0104628971337711
90000 min thresh: 0.0101807237208479
91000 min thresh: 0.0101807237208479
92000 min thresh: 0.0101807237208479
93000 min thresh: 0.0101148213264655
93613 error: 128.55263        time: 1.88 mins
```

وزن های نهایی شبکه عصبی:

```
> (mean((newdataSet[,18] - net.dat$net.result[[1]])^2))
[1] 0.07713929
> net.dat$weights
[[1]]
[[1]][[1]]
                [,1]         [,2]         [,3]
 [1,] -1.194094e+02  2.49193189   -0.8566233
 [2,] -8.542754e-01  2.09832320   -1.8633495
 [3,]  5.990579e+00  0.46713143   13.2754515
 [4,]  4.779166e+01  1.60979619   18.8400379
 [5,]  1.776029e-01  2.03627731   18.2483324
 [6,]  7.881294e-01  0.86275530   18.9639683
 [7,]  2.242479e+01  3.01740227    1.5375802
 [8,] -2.199618e+00  1.51563657  -41.9649863
 [9,]  8.225797e-01  2.66740570    2.9257766
[10,]  3.472843e+01  0.34137250   -0.3818816
[11,] -9.561536e-02  1.86498384   -1.4431023
[12,] -3.597765e-01 -0.04953662    1.0027151
[13,]  1.568930e+00  1.40112541    0.8621068
[14,]  1.394041e-03  1.38545800  133.7830126
[15,]  1.332976e+00  2.08352543    0.3514997
[16,] -7.900786e-02  2.98015175 -138.9692805
[17,]  1.735475e+00  1.25344809    7.6074704
[18,] -2.765839e+01 -0.21780039   12.1603713

[[1]][[2]]
          [,1]
[1,]   1.283760
[2,] 278.502106
[3,]   1.363564
[4,]  -5.000547
```

در آخر برای confusion matrix و accuracy داریم:

(قطر اصلی ماتریس میشود تعداد پیش بینی های درست انجام شده توسط شبکه، و accuracy میشود جمع قطر اصلی به کل رکوردها (3333))

```
86  tb = table(newdataSet$Churn.,round(net.dat$net.result[[1]]))
87  tb
88  sum(diag(tb)/sum(tb))
89
90
91
92
93
94
95
96
97
```

86:61 (Top Level)

Console Terminal Background Jobs

R 4.2.2 · ~/

```
> plot(net.dat)
> tb = table(newdataSet$Churn.,round(net.dat$net.result[[1]]))
> tb

      0    1
  0 2848    2
  1  280  203
> sum(diag(tb)/sum(tb))
[1] 0.9153915
>
```

13) با توجه به sensitivity analysis می توان دریافت که اثرگذارترین متغیرها عبارتند از:

Vmail.Plan Intl.Plan
Account.Length
DayMins
CustServiceCall
…

(12

باید پیش از ران کردن الگوریتم، متغیرها را standardize کنیم تا معیار فاصله میان رکوردها به سمت ویژگی خاصی متمایل نشود.

```
install.packages("cluster")

library(cluster)
data_set = read.csv(file.choose(), header = T)
data_set = data_set[, -c(1, 16)] #Omit name & rating
data_set = na.omit(data_set)

data_set$mfr = factor(data_set$mfr)
levels(data_set$mfr)
table(data_set$mfr)

data_set$type = factor(data_set$type)
levels(data_set$type)
table(data_set$type)

data_set = data_set[, -c(1)] # Omit type

newdataSet = scale(newdataSet)

kc = kmeans(newdataSet , centers = 4)
clusplot( newdataSet , kc$cluster , color = TRUE , shade = TRUE  )

kc$centers
```

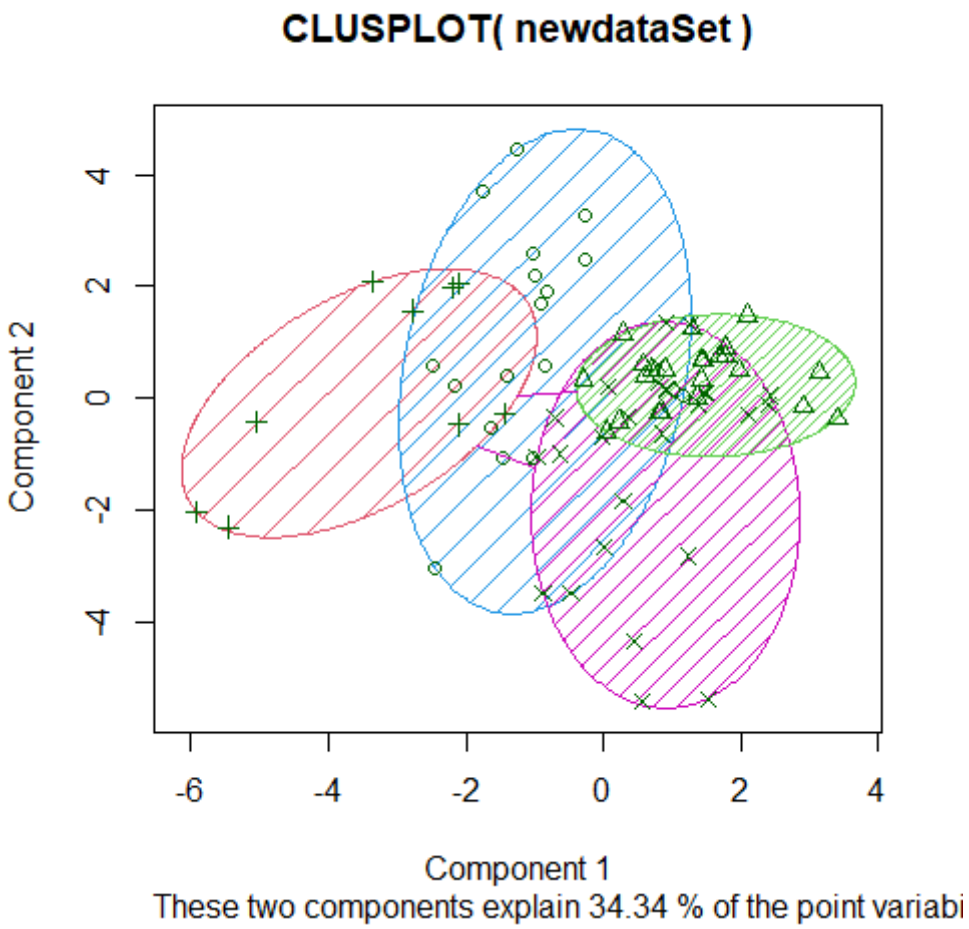
```
> data_set$mfr = factor(data_set$mfr)
> levels(data_set$mfr)
[1] "A" "G" "K" "N" "P" "Q" "R"
> table(data_set$mfr)

 A  G  K  N  P  Q  R
 1 22 23  6  9  8  8
>
> data_set$type = factor(data_set$type)
> levels(data_set$type)
[1] "C" "H"
> table(data_set$type)

 C  H
74  3
>
```

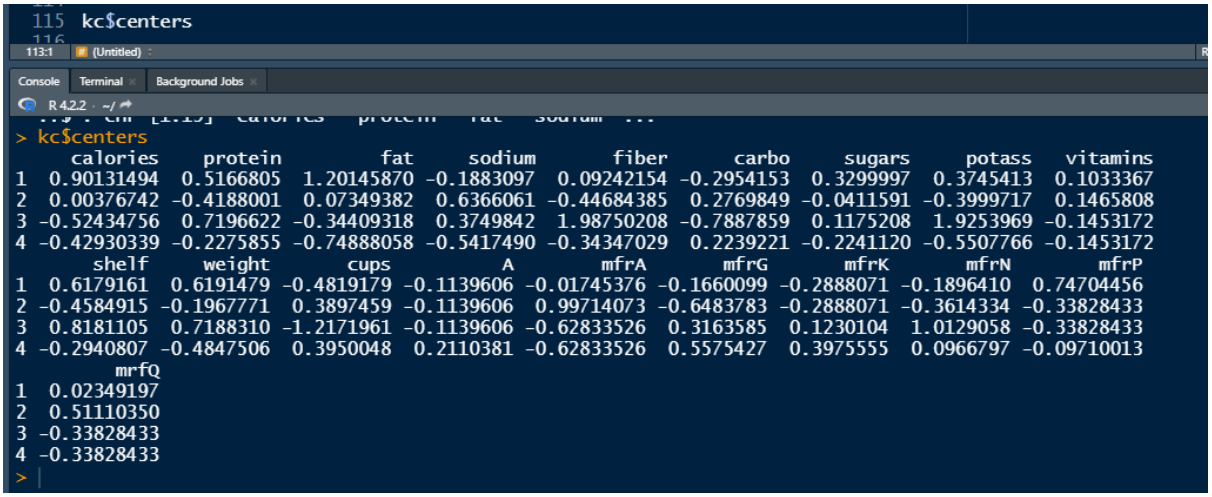
دقت شود که متغیر type را حذف کردیم زیرا باعث bias الگوریتم به رکوردهای با type == C خواهد شد.

سپس با ران کردن الگوریتم داریم:



(13

با اجرای دستور kc$centers، میتوان میانگین هر خوشه را بدست آورد و حدود ویژگی های رکوردهای هر خوشه را یافت



(10

در ابتدا متغیر های name و rating را حذف میکنیم که در الگوریتم نباید وارد شوند. سپس متغیر type را هم حذف میکنیم زیرا همانطور که قبلا گفته شد باعث bias الگوریتم میشود. با ران گرفتن دستور table میتوان متوجه شد که ستون های potass carbo هم دارای -1 هستند و نباید در imputation بیایند.

```
data_set = read.csv(file.choose(), header = T)
data_set = data_set[, -c(1, 3)] ## omit name and type
data_set = data_set[, -c(14)] #omit rating
table(data_set["potass"])
table(data_set["carbo"])
table(data_set["sugars"])

data_set$sugars[58] = NA ## based on the exercise
right = paste0(colnames(data_set)[-c(8, 1, 7, 9)] , collapse = "+") #we should not
#consider the target rating
formula = paste0(colnames(data_set)[8] , "~" , right)
formula = as.formula(formula)

model1 = lm( formula , data = data_set)
Sugars_missInd = which(is.na(data_set)[,8] == T)

Imputed_Sugars = predict(model1 , newdata = data_set[Sugars_missInd,c(2, 3, 4, 5, 6,
                                                      10, 11, 12, 13)])
Imputed_Sugars
data_set[Sugars_missInd,8] = Imputed_Sugars

135:1 (Untitled)
```

```
Console Terminal Background Jobs
R 4.2.2 · ~/
> #consider the target rating
> formula = paste0(colnames(data_set)[8] , "~" , right)
> formula = as.formula(formula)
>
> model1 = lm( formula , data = data_set)
> Sugars_missInd = which(is.na(data_set)[,8] == T)
>
> Imputed_Sugars = predict(model1 , newdata = data_set[Sugars_missInd,c(2, 3, 4, 5, 6,
+                                                       10, 11, 12, 13)])
> Imputed_Sugars
      58
3.616061
> data_set[Sugars_missInd,8] = Imputed_Sugars
```

```
potass
 -1  15  20  25  30  35  40  45  50  55  60  65  70  80  85  90  95 100 105 110 115 120 125 130 135 140
  2   1   1   4   4   5   4   4   1   3   3   1   1   1   1   5   4   3   2   5   1   3   1   1   1   2
160 170 190 200 230 240 260 280 320 330
  2   2   2   1   1   1   1   1   1   1
> table(data_set["carbo"])
carbo
 -1   5   7   8   9  10 10.5  11 11.5  12  13 13.5  14  15  16  17  18  19  20  21  22  23
  1   1   1   2   1   2   2   5   1   7   8   1   7   8   7   6   3   1   3   7   2   1
```

سپس مقدار sugar ردیف 58، (Quaker Oatmeal.) را که برابر مقدار نامعتبر -1 است، NA میکنیم. در آخر مدل رگرسیون را برای پیش بینی ستون sugars

برازش میدهیم و فقط سطر 58 آنرا(Imputed_Sugars) در dataset اصلی قرار میدهیم.

57	Q	100	4	1	135	2.0	14.0	6.000000	110	25	3	1.00	0.50
58	Q	100	5	2	0	2.7	-1.0	3.616061	110	0	1	1.00	0.67
59	K	120	3	1	210	5.0	14.0	12.000000	240	25	2	1.33	0.75

همانطور که دیده میشود، در سطر 58 و ستون sugar مقدار imputed را داریم.
(مقدار carbo در این سطر -1 است که آن را در سوال بعدی جایگذاری میکنیم)

(11
همانند سوال قبل عمل میکنیم منتها این بار sugars را در فرمول می آوریم زیرا داده گمشده ندارد.(در سوال قبل جاگذاری شد)

```
140
141 data_set$carbo[58] = NA
142 right =  paste0(colnames(data_set)[-c(1, 7, 9)] , collapse = "+") #we should not
143 #consider the target rating
144 formula = paste0(colnames(data_set)[7] , "~" , right)
145 formula = as.formula(formula)
146
```

```
Console  Terminal  Background Jobs
R 4.2.2 · ~/
     5     6     7     8     9
     5     7     4     5     4
    11    12    13    14    15
     5     7     4     3     2
> data_set$carbo[58] = NA
> right =  paste0(colnames(data_set)[-c(1, 7, 9)] , collapse = "+") #we should not
> #consider the target rating
> formula = paste0(colnames(data_set)[7] , "~" , right)
> formula = as.formula(formula)
> formula
carbo ~ calories + protein + fat + sodium + fiber + sugars +
    vitamins + shelf + weight + cups
```

سپس برای این متغیر نیز رگرسیون را برازش میدهیم و مقدار Imputed_Carbo برابر 12.43 را در سطر 58 ام قرار میدهیم.

```
model1 = lm( formula , data = data_set)
Sugars_missInd = which(is.na(data_set)[,8] == T)

Imputed_Sugars = predict(model1 , newdata = data_set[Sugars_missInd,c(2, 3, 4, 5, 6,
                                                    10, 11, 12, 13)])
Imputed_Sugars
data_set[Sugars_missInd,8] = Imputed_Sugars

data_set$carbo[58] = NA
right =  paste0(colnames(data_set)[-c(1, 7, 9)] , collapse = "+") #we should not
#consider the target rating
formula = paste0(colnames(data_set)[7] , "~" , right)
formula = as.formula(formula)
model1 = lm( formula , data = data_set)
Carbo_missInd = which(is.na(data_set)[,7] == T)
Carbo_missInd
Imputed_Carbo = predict(model1 , newdata = data_set[Carbo_missInd,c(2, 3, 4, 5, 6, 8,
                                                                    10, 11, 12, 13)])
Imputed_Carbo
data_set[Carbo_missInd,7] = Imputed_Carbo
```

144:54 (Untitled)

Console | Terminal | Background Jobs

R 4.2.2 · ~/

```
> #consider the target rating
> formula = paste0(colnames(data_set)[7] , "~" , right)
> formula = as.formula(formula)
> model1 = lm( formula , data = data_set)
> Carbo_missInd = which(is.na(data_set)[,7] == T)
> Carbo_missInd
[1] 58
> Imputed_Carbo = predict(model1 , newdata = data_set[Carbo_missInd,c(2, 3, 4, 5, 6, 8,
+                                                                     10, 11, 12, 13)])
> Imputed_Carbo
      58
12.43121
> data_set[Carbo_missInd,7] = Imputed_Carbo
>
```

57	Q	100	4	1	135	2.0	14.00000	6.000000	110	25	3	1.00	0.50
58	Q	100	5	2	0	2.7	12.43121	3.616061	110	0	1	1.00	0.67
59	K	120	3	1	210	5.0	14.00000	12.000000	240	25	2	1.33	0.75

مقادیر carbo و sugars در سطر 58 ام جاگذاری شدند.

1)

متغیر های State, AreaCode و Phone را حذف میکنیم زیرا از بخش EDA به یاد داریم که از churn مستقل اند.

سپس متغیرهای correlated را نیز حذف میکنیم (گفته شد که از هر دو جفت mins و charge مربوط به یک نوع call با هم correlation = 1 دارند، پس یکی را حذف میکنیم)

فرمول بدست آمده برای درخت CART بصورت زیر است:

```
156 data_set = read.csv(file.choose(), stringsAsFactors = T, header = T)
157 library(rpart)
158 library(rpart.plot)
159
160
161 right = paste0(colnames(data_set)[-c(22, 1, 2, 4, 5, 11, 14, 17, 20)] , collapse = "+") #Omit
162 #consider the target rating
163 formula = paste0(colnames(data_set)[22] , "~" , right)
164 formula = as.formula(formula)
165 formula
166
167 model3 = rpart(formula , data = data_set)
168 predicted = predict(model3 , type = "class" )
169 (tab = table(data_set$Churn. , predicted))
170 TN = tab[1,1] ; FP = tab[1,2] ; FN = tab[2,1] ; TP = tab[2,2]
171
172 (FPP = FP / (FP + TP))
173 (FNP = FN / (FN + TN))
174 (ErrorRate = (FN + FP) / (FN + FP + TN + TP))
175 (Accuracy = 1 - ErrorRate)
176 (Sensitivity = TP / (TP + FN))
177 (Specifity = TN / (TN + FP))
178
179
159:1 (Untitled)

Console Terminal Background Jobs
R 4.2.2 · ~/

> right = paste0(colnames(data_set)[-c(22, 1, 2, 4, 5, 11, 14, 17, 20)] , collapse = "+") #Omit sta
de, phone and charges
> #consider the target rating
> formula = paste0(colnames(data_set)[22] , "~" , right)
> formula = as.formula(formula)
> formula
Churn. ~ Account.Length + Int.l.Plan + VMail.Plan + VMail.Message +
    Day.Mins + Day.Calls + Eve.Mins + Eve.Calls + Night.Mins +
    Night.Calls + Intl.Mins + Intl.Calls + CustServ.Calls
> model3 = rpart(formula , data = data_set)
```

حال الگوریتم CART را ران کرده و ماتریس confusion را بدست می آوریم:

```
> model3 = rpart(formula , data = data_set)
> predicted = predict(model3 , type = "class" )
> (tab = table(data_set$Churn. , predicted))
        predicted
         False. True.
  False.   2815    35
  True.     122   361
> TN = tab[1,1] ; FP = tab[1,2] ; FN = tab[2,1] ; TP = tab[2,2]
> (FPP = FP / (FP + TP))
[1] 0.08838384
> (FNP = FN / (FN + TN))
[1] 0.04153899
> (ErrorRate = (FN + FP) / (FN + FP + TN + TP))
[1] 0.04710471
> (Accuracy = 1 - ErrorRate)
[1] 0.9528953
> (Sensitivity = TP / (TP + FN))
[1] 0.747412
> (Specifity = TN / (TN + FP))
[1] 0.9877193
```