Algorithm

Anandu R

5/31/2020

Importing necessary packages/Libraries

```
invisible(library(dplyr))
invisible(library(lubridate))
invisible(library(caTools))
invisible(library(data.table))
invisible(library(rpart))
invisible(library(rpart.plot))
invisible(library(C50))
```

Generating the dataset

```
set.seed(1)
speed = round(rnorm(1000,50,15),2)
dist_prev = abs(round(rnorm(1000,2,1),2))
dist_next = abs(round(rnorm(1000,2,1),2))
crowd_curr = rpois(1000,25)
crowd_next = rpois(1000,25)
booked = rpois(1000,40)
schd_time = sample(seq(strptime('01/01/2018',format = "%d/%m/%Y"),
                       strptime('01/01/2019',format = "%d/%m/%Y"),
                       by="hour"), 1000, replace = T)
arr_time = schd_time+(rnorm(1000,300,350)*-1)
on_time = ifelse(difftime(arr_time,schd_time)<=0,1,0)
data = data.frame(crowd_curr,crowd_next,booked,
                  dist_prev, dist_next,speed,
                  schd_time,arr_time,on_time)
head(select(data,crowd_curr,crowd_next,booked,on_time))
```

```
##   crowd_curr crowd_next booked on_time
## 1         28         27     39       1
## 2         26         24     38       1
## 3         31         21     28       1
## 4         20         28     41       1
## 5         27         21     36       1
## 6         23         21     43       1
```

Generating an algorithm to label the datasets

Each record is considered as a bus and the label is the indication given to the bus driver whether to maintain speed, decrease speed, or to increase represented by 0,1,2 respectively

```
indicate = ifelse((
    (crowd_curr<28)&
    (crowd_next>28)&
    (booked>30)&
    (on_time==0)),2,ifelse((
      (crowd_curr>28)&
      (crowd_next<28)&
      (booked<30)&
      (on_time==1)),1,0))
data$indicate = indicate
head(select(data,crowd_curr,booked,on_time,indicate))
```

```
##   crowd_curr booked on_time indicate
## 1         28     39       1        0
## 2         26     38       1        0
## 3         31     28       1        1
## 4         20     41       1        0
## 5         27     36       1        0
## 6         23     43       1        0
```

The table below indicates the indications that each of the bus instances receive

```
table(data$indicate)
```

```
##
##   0   1   2
## 948   5  47
```

Modelling a decision tree algorithm to make future scheduling

Splitting the data into train and test

```
set.seed(1)
split = sample.split(data$indicate, SplitRatio = 0.75)
train = data[split,]
test = data[!split,]
```

Creating a penalty matrix to avoid miscalculation

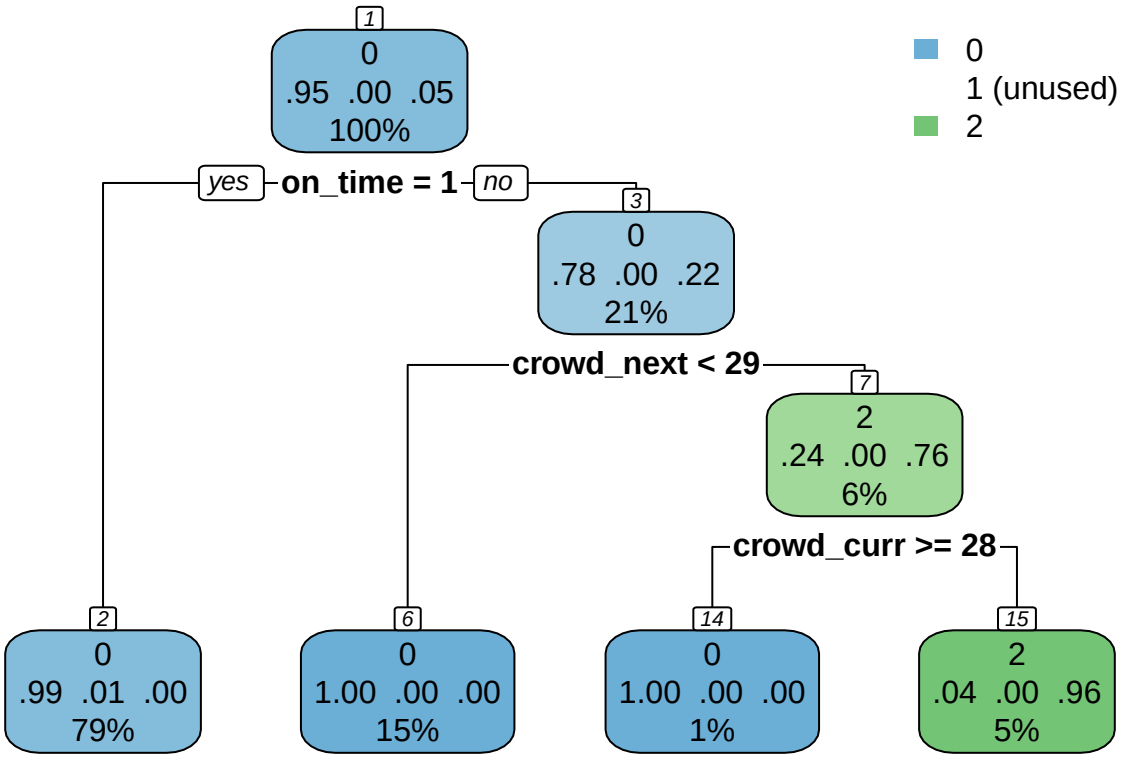
```
penalty.matrix <- matrix(c(1,1,0,10,0,10,0,0,0), byrow=TRUE, nrow=3)
```

Building the decision tree model with rpart

```
dtree <- rpart(indicate~.,data=data,method = "class")
```

Visualizing the decision tree

```
rpart.plot(dtree, nn=TRUE)
```



Using C50 algorithm to model

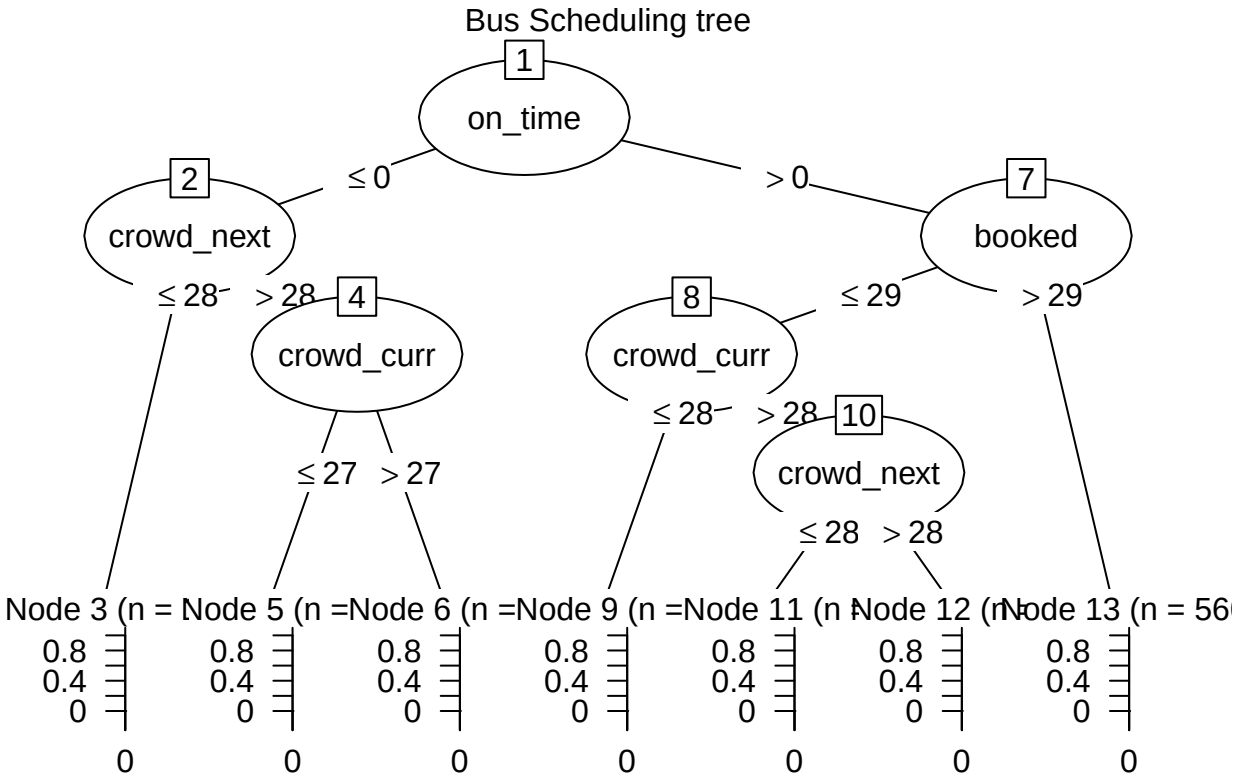
```
dtree = C5.0(train[,-c(4:8,10)],as.factor(train[,10]))
summary(dtree)
```

```
##
## Call:
## C5.0.default(x = train[, -c(4:8, 10)], y = as.factor(train[, 10]))
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun May 31 21:20:29 2020
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 750 cases (5 attributes) from undefined.data
##
## Decision tree:
##
## on_time <= 0:
## :...crowd_next <= 28: 0 (118)
## :   crowd_next > 28:
## :   :...crowd_curr <= 27: 2 (36/1)
```

```
## :       crowd_curr > 27: 0 (11)
## on_time > 0:
## :...booked > 29: 0 (560)
##     booked <= 29:
##     :...crowd_curr <= 28: 0 (18)
##         crowd_curr > 28:
##         :...crowd_next <= 28: 1 (4)
##             crowd_next > 28: 0 (3)
##
##
## Evaluation on training data (750 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       7    1( 0.1%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##     710           1    (a): class 0
##             4          (b): class 1
##                  35    (c): class 2
##
##
##  Attribute usage:
##
##  100.00% on_time
##   78.00% booked
##   22.93% crowd_next
##    9.60% crowd_curr
##
##
## Time: 0.0 secs
```

Visualize outcome

```
plot(dtree, main = 'Bus Scheduling tree')
```


Bus Scheduling tree
1
on_time
≤ 0
> 0
2
crowd_next
7
booked
≤ 28
> 28
4
crowd_curr
8
crowd_curr
≤ 29
> 29
≤ 28
> 28
10
crowd_next
≤ 27
> 27
≤ 28
> 28
Node 3 (n =
Node 5 (n =
Node 6 (n =
Node 9 (n =
Node 11 (n =
Node 12 (n =
Node 13 (n = 56
0.8
0.4
0
0