Assessed Practicals 2

Praveen Gopal Reddy

21/07/2021

Table of Contents

Aim:

To Investigate the output variable called Hattack, and gives a 1/0 answer to the question 'did the person had a heart attack?'

```
# After importing csv file of heart_attack.csv, the dataframe we are using is
:
summary(heart_attack)
```

```
##     Hattack            age              cp            trtbps
##  Min.   :0.0000   Min.   :29.00   Min.   :0.000   Min.   : 94.0
##  1st Qu.:0.0000   1st Qu.:47.50   1st Qu.:0.000   1st Qu.:120.0
##  Median :1.0000   Median :55.00   Median :1.000   Median :130.0
##  Mean   :0.5446   Mean   :54.37   Mean   :0.967   Mean   :131.6
##  3rd Qu.:1.0000   3rd Qu.:61.00   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :1.0000   Max.   :77.00   Max.   :3.000   Max.   :200.0
##       chol          thalachh        oldpeak
##  Min.   :126.0   Min.   : 71.0   Min.   :0.00
##  1st Qu.:211.0   1st Qu.:133.5   1st Qu.:0.00
##  Median :240.0   Median :153.0   Median :0.80
##  Mean   :246.3   Mean   :149.6   Mean   :1.04
```

```
##  3rd Qu.:274.5   3rd Qu.:166.0   3rd Qu.:1.60
##  Max.   :564.0   Max.   :202.0   Max.   :6.20
```

Tasks

1.(a)

Fit a logistic regression model for the outputs using all of the available inputs. Explain your model and report your results. Identify the direction and magnitude of each effect from the fitted coefficients. Comment on your findings.

```
#Build logistic regression model on all data inputs
model_all <- glm(Hattack ~ age+cp+trtbps+chol+thalachh+oldpeak, data =
heart_attack,family=binomial(link = "logit"))
summary(model_all)
```

```
##
## Call:
## glm(formula = Hattack ~ age + cp + trtbps + chol + thalachh +
##     oldpeak, family = binomial(link = "logit"), data = heart_attack)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2816  -0.6995   0.3484   0.7282   2.3245
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.499685   1.938749  -0.258  0.79661
## age         -0.010199   0.018666  -0.546  0.58479
## cp           0.923408   0.156065   5.917 3.28e-09 ***
## trtbps      -0.016751   0.009316  -1.798  0.07217 .
## chol        -0.001630   0.002962  -0.550  0.58215
## thalachh     0.025266   0.007964   3.172  0.00151 **
## oldpeak     -0.823561   0.164197  -5.016 5.28e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 281.17  on 296  degrees of freedom
## AIC: 295.17
##
## Number of Fisher Scoring iterations: 5
```

- In above logistic model we have used binomial method because our dependent variable contains binary data i.e,0 or 1. Based on above summary we can see some coefficients are less significant and some are more significant.

- We see oldpeak and cp(chest pain) are highly significant with p-value closer to zero, next significant variable is thalachh(heart rate). The rest of inputs are not significant since p-value is greater than 0.05.
- There are total 4 negative estimates and two positive estimates. The negative value of oldpeak tells us that all other variables being equal ,the probability of getting heart attack is less likely to happen.
- The positive value of significant variable cp tell us tha all other variables being equal ,the probability of getting heart attack has more likely to occur
- The magnitude of of cp is 45 times greater than thalachh because if we look at coefficients of thalachh is 0.02*45=0.9 which is approximately equal to the coefficients of cp .The magnitude of age and trtbps is almost same so they have Strong association each other.

1.(b)

Present the value of each coefficient estimate with a 95% confidence interval. Which inputs would you say have strong effects? Order the inputs in terms of decreasing effect. Comment on your findings and justify your reasoning.

```
#Build confidence interval for coefficients estimates.
#age 95% confidence interval
t_critical=qnorm(0.975)
coeff_estimates=summary(model_all)$coefficients
estimate_age=coeff_estimates[2,1]
sterr_age=coeff_estimates[2,2]
interval_min_age=estimate_age-t_critical*sterr_age
interval_max_age=estimate_age+t_critical*sterr_age
print(paste(c("estimate_age:",estimate_age),collapse=""))

## [1] "estimate_age:-0.010198937863734"

print(paste(c("min_age:",interval_min_age),collapse=""))

## [1] "min_age:-0.0467833521856259"

print(paste(c("max_age:",interval_max_age),collapse=""))

## [1] "max_age:0.0263854764581579"

#cp 95% confidence interval
estimate_cp=coeff_estimates[3,1]
sterr_cp=coeff_estimates[3,2]
interval_min_cp=estimate_cp-t_critical*sterr_cp
interval_max_cp=estimate_cp+t_critical*sterr_cp
print(paste(c("estimate_cp:",estimate_cp),collapse=""))

## [1] "estimate_cp:0.923408214525919"
```

```r
print(paste(c("min_cp:",interval_min_cp),collapse=""))
```

```
## [1] "min_cp:0.617526798374202"
```

```r
print(paste(c("max_cp:",interval_max_cp),collapse=""))
```

```
## [1] "max_cp:1.22928963067764"
```

```r
#trtbps(resting blood pressure) 95% confidence interval
estimate_trtbps=coeff_estimates[4,1]
sterr_trtbps=coeff_estimates[4,2]
interval_min_trtbps=estimate_trtbps-t_critical*sterr_trtbps
interval_max_trtbps=estimate_trtbps+t_critical*sterr_trtbps
print(paste(c("estimate_trtbps:",estimate_trtbps),collapse=""))
```

```
## [1] "estimate_trtbps:-0.016750962906939"
```

```r
print(paste(c("min_trtbps:",interval_min_trtbps),collapse=""))
```

```
## [1] "min_trtbps:-0.0350104589009701"
```

```r
print(paste(c("max_trtbps:",interval_max_trtbps),collapse=""))
```

```
## [1] "max_trtbps:0.00150853308709208"
```

```r
#chol(Cholestoral) 95% confidence interval thalachh
estimate_chol=coeff_estimates[5,1]
sterr_chol=coeff_estimates[5,2]
interval_min_chol=estimate_chol-t_critical*sterr_chol
interval_max_chol=estimate_chol+t_critical*sterr_chol
print(paste(c("estimate_chol:",estimate_chol),collapse=""))
```

```
## [1] "estimate_chol:-0.00163012058232201"
```

```r
print(paste(c("min_chol:",interval_min_chol),collapse=""))
```

```
## [1] "min_chol:-0.00743648945696864"
```

```r
print(paste(c("max_chol:",interval_max_chol),collapse=""))
```

```
## [1] "max_chol:0.00417624829232463"
```

```r
#thalachh(maximum hear rate) 95% confidence interval
estimate_thalachh=coeff_estimates[6,1]
sterr_thalachh=coeff_estimates[6,2]
interval_min_thalachh=estimate_thalachh-t_critical*sterr_thalachh
interval_max_thalachh=estimate_thalachh+t_critical*sterr_thalachh
print(paste(c("estimate_thalachh:",estimate_thalachh),collapse=""))
```

```
## [1] "estimate_thalachh:0.025265887682973"
```

```r
print(paste(c("min_thalachh:",interval_min_thalachh),collapse=""))
```

```
## [1] "min_thalachh:0.00965645311517867"
```

```
print(paste(c("max_thalachh:",interval_max_thalachh),collapse=""))
```

```
## [1] "max_thalachh:0.0408753222507673"
```

```
#oldpeak(previois peak) 95% confidence interval
estimate_oldpeak=coeff_estimates[7,1]
sterr_oldpeak=coeff_estimates[7,2]
interval_min_oldpeak=estimate_oldpeak-t_critical*sterr_oldpeak
interval_max_oldpeak=estimate_oldpeak+t_critical*sterr_oldpeak
print(paste(c("estimate_oldpeak:",estimate_oldpeak),collapse=""))
```

```
## [1] "estimate_oldpeak:-0.823561355641486"
```

```
print(paste(c("min_oldpeak:",interval_min_oldpeak),collapse=""))
```

```
## [1] "min_oldpeak:-1.14538110039984"
```

```
print(paste(c("max_oldpeak:",interval_max_oldpeak),collapse=""))
```

```
## [1] "max_oldpeak:-0.501741610883129"
```

As you see all estimates are within the confidence interval.

Lets find out strong inputs based on coefficients.

```
#first find max value of each input variable from data set.
max_value=apply(heart_attack[2:7],2,max)

#coefficients from  age to oldpeak
coeff=model_all$coefficients[2:7]
#mutliply max value with coeffiecients of estimates.
extreme_effects=sort(coeff*max_value,decreasing = TRUE)
extreme_effects
```

```
##   thalachh         cp        age       chol     trtbps    oldpeak
##  5.1037093  2.7702246 -0.7853182 -0.9193880 -3.3501926 -5.1060804
```

- The maximum value of oldpeak input is 6.2 from the data set.so the maximum possible effect for oldpeak is 6.2*(-0.82356)=-5.1060804, and thus the most extreme possible effect for oldpeak is greater than the effect for any of the other variables in terms of highest negative impact.
- Similarly thalachh has strong effect on dependent variable , if we do same calculation like above we get 202* 0.025266=5.103732. so thalachh has strong positive impact but since this variable is less significant than CP. CP will have higher effect than thalachh.
- So our final inputs in terms of decreasing effect based on its extreme possible effect and p-value (significant level) is:
  - oldpeak
  - cp

- thalachh
- trtbps
- age
- chol

However we need to check how these factors importance change based on AIC in next question.

1.(c)

Using aic, perform model selection to determine which factors are useful to predict the result of the heart attack. Use a 'greedy' input selection procedure, as follows: At each stage evaluate the quality of fit using aic and stop if this gets worse. Report your results and comment on your findings. Are your findings consistent with the Task 1.(b)?

(i)

select the best model with 1 input;

```
#lets build models for one input
one_input_model1=model_all <- glm(formula=Hattack ~ age,family=binomial,data
= heart_attack)
one_input_model1$aic  #AIC is 405.86

## [1] 405.8614

one_input_model2=one_input_model_all <- glm(formula=Hattack ~
cp,family=binomial,data = heart_attack)
one_input_model2$aic  #AIC is 360.060

## [1] 360.0609

one_input_model3=one_input_model_all <- glm(formula=Hattack ~
trtbps,family=binomial,data = heart_attack)
one_input_model3$aic  #AIC is 415.2248

## [1] 415.2248

one_input_model4=one_input_model_all <- glm(formula=Hattack ~
chol,family=binomial,data = heart_attack)
one_input_model4$aic  #AIC is 419.4295

## [1] 419.4295

one_input_model5=one_input_model_all <- glm(formula=Hattack ~
thalachh,family=binomial,data = heart_attack)
one_input_model5$aic  #AIC is 363.2569

## [1] 363.2569
```

```
one_input_model6=one_input_model_all <- glm(formula=Hattack ~
oldpeak,family=binomial,data = heart_attack)
one_input_model6$aic  #AIC is 358.9977
```

```
## [1] 358.9977
```

- so our best model for one input is one_input_model6 i.e, oldpeak with AIC=358.99

(ii)

fixing that input, select the best two-input model (i.e. try all the other 5 inputs with the one you selected first).

```
#models for two input
two_input_model1=model_all <- glm(formula=Hattack ~
oldpeak+age,family=binomial,data = heart_attack)
two_input_model1$aic
```

```
## [1] 354.5698
```

```
two_input_model2=model_all <- glm(formula=Hattack ~
oldpeak+cp,family=binomial,data = heart_attack)
two_input_model2$aic
```

```
## [1] 306.529
```

```
two_input_model3=model_all <- glm(formula=Hattack ~
oldpeak+chol,family=binomial,data = heart_attack)
two_input_model3$aic
```

```
## [1] 359.7496
```

```
two_input_model4=model_all <- glm(formula=Hattack ~
oldpeak+trtbps,family=binomial,data = heart_attack)
two_input_model4$aic
```

```
## [1] 359.1039
```

```
two_input_model5=model_all <- glm(formula=Hattack ~
oldpeak+thalachh,family=binomial,data = heart_attack)
two_input_model5$aic
```

```
## [1] 331.8846
```

- Our best model is two_input_model2 with oldpeak+cp where AIC is smaller than others AIC=306.529

(iii)

select the best three-input model containing the first two inputs you chose, etc.

```
#models for three input
three_input_model1=model_all <- glm(formula=Hattack ~
```

```
oldpeak+cp+age,family=binomial,data = heart_attack)
three_input_model1$aic

## [1] 302.1412

three_input_model2=model_all <- glm(formula=Hattack ~
oldpeak+cp+chol,family=binomial,data = heart_attack)
three_input_model2$aic

## [1] 308.047

three_input_model3=model_all <- glm(formula=Hattack ~
oldpeak+cp+trtbps,family=binomial,data = heart_attack)
three_input_model3$aic

## [1] 304.2503

three_input_model4=model_all <- glm(formula=Hattack ~
oldpeak+cp+thalachh,family=binomial,data = heart_attack)
three_input_model4$aic

## [1] 294.3654
```

- Our best model for three input is three_input_model4 with oldpeak+cp+thalachh where AIC is smaller than others which is AIC=294.3654

```
#four input model
four_input_model1=model_all <- glm(formula=Hattack ~
oldpeak+cp+thalachh+age,family=binomial,data = heart_attack)
four_input_model1$aic

## [1] 294.8701

four_input_model2=model_all <- glm(formula=Hattack ~
oldpeak+cp+thalachh+chol,family=binomial,data = heart_attack)
four_input_model2$aic

## [1] 295.523

four_input_model3=model_all <- glm(formula=Hattack ~
oldpeak+cp+thalachh+trtbps,family=binomial,data = heart_attack)
four_input_model3$aic

## [1] 291.9414
```

- Our best model for four input is four_input_model3 with oldpeak+cp+thalachh+trtbps where AIC is smaller than others which is AIC=291.9414

```
#five input model
five_input_model1=model_all <- glm(formula=Hattack ~
oldpeak+cp+thalachh+trtbps+age,family=binomial,data = heart_attack)
five_input_model1$aic
```

```
## [1] 293.4722

five_input_model2=model_all <- glm(formula=Hattack ~
oldpeak+cp+thalachh+trtbps+chol,family=binomial,data = heart_attack)
five_input_model2$aic

## [1] 293.4702
```

- Our best model for five input is five_input_model2 with oldpeak+cp+thalachh+trtbps+chol where AIC is smaller than others which is AIC=293.4702

As input variables are increased AIC is getting smaller. From results of above models we can say that it is consistent with 1(b) results with little variation in order of importance for factors like cp and thalachh.

2

Use the rpart package to create a decision tree classification model. Explain and visualise your model and interpret the fitted model.

```
#install.packages('rpart')
library(rpart)
#install.packages("rpart.plot")
library(rpart.plot)

#Building Decision tree model
dectree <- rpart(Hattack~., data = heart_attack, method = 'class',
control=rpart.control(cp = 0.01),parms = list(split = "gini"))

#summary of decision tree
summary(dectree)

## Call:
## rpart(formula = Hattack ~ ., data = heart_attack, method = "class",
##     parms = list(split = "gini"), control = rpart.control(cp = 0.01))
##   n= 303
##
##           CP nsplit rel error    xerror       xstd
## 1 0.47101449      0 1.0000000 1.0000000 0.06281757
## 2 0.02173913      1 0.5289855 0.5289855 0.05394174
## 3 0.01449275      5 0.4420290 0.5869565 0.05582370
## 4 0.01000000     11 0.3478261 0.6159420 0.05666792
##
## Variable importance
##       cp  oldpeak thalachh      age     chol   trtbps
##       37       21       18       12        8        4
##
## Node number 1: 303 observations,    complexity param=0.4710145
##   predicted class=1  expected loss=0.4554455  P(node) =1
##     class counts:   138   165
```

```
##    probabilities: 0.455 0.545
##   left son=2 (143 obs) right son=3 (160 obs)
##   Primary splits:
##       cp       < 0.5   to the left,  improve=40.019760, (0 missing)
##       thalachh < 147.5 to the left,  improve=26.321280, (0 missing)
##       oldpeak  < 1.7   to the right, improve=24.729700, (0 missing)
##       age      < 54.5  to the right, improve=12.330570, (0 missing)
##       chol     < 245.5 to the right, improve= 3.634721, (0 missing)
##   Surrogate splits:
##       thalachh < 148.5 to the left,  agree=0.693, adj=0.350, (0 split)
##       oldpeak  < 0.85  to the right, agree=0.640, adj=0.238, (0 split)
##       age      < 54.5  to the right, agree=0.584, adj=0.119, (0 split)
##       chol     < 246.5 to the right, agree=0.568, adj=0.084, (0 split)
##       trtbps   < 117.5 to the left,  agree=0.545, adj=0.035, (0 split)
##
## Node number 2: 143 observations,    complexity param=0.02173913
##   predicted class=0  expected loss=0.2727273  P(node) =0.4719472
##     class counts:   104    39
##    probabilities: 0.727 0.273
##   left son=4 (90 obs) right son=5 (53 obs)
##   Primary splits:
##       oldpeak  < 0.7   to the right, improve=11.001070, (0 missing)
##       thalachh < 177.5 to the left,  improve= 6.147643, (0 missing)
##       trtbps   < 143   to the right, improve= 2.450775, (0 missing)
##       age      < 53.5  to the right, improve= 1.972605, (0 missing)
##       chol     < 272.5 to the right, improve= 1.641414, (0 missing)
##   Surrogate splits:
##       thalachh < 146.5 to the left,  agree=0.755, adj=0.340, (0 split)
##       age      < 49.5  to the right, agree=0.657, adj=0.075, (0 split)
##       trtbps   < 113   to the right, agree=0.650, adj=0.057, (0 split)
##
## Node number 3: 160 observations,    complexity param=0.01449275
##   predicted class=1  expected loss=0.2125  P(node) =0.5280528
##     class counts:    34   126
##    probabilities: 0.213 0.788
##   left son=6 (62 obs) right son=7 (98 obs)
##   Primary splits:
##       age      < 56.5  to the right, improve=6.171461, (0 missing)
##       oldpeak  < 1.95  to the right, improve=6.050000, (0 missing)
##       thalachh < 150.5 to the left,  improve=3.868207, (0 missing)
##       chol     < 223.5 to the right, improve=2.430000, (0 missing)
##       trtbps   < 151   to the right, improve=1.875495, (0 missing)
##   Surrogate splits:
##       trtbps   < 139   to the right, agree=0.738, adj=0.323, (0 split)
##       thalachh < 151.5 to the left,  agree=0.713, adj=0.258, (0 split)
##       oldpeak  < 1.45  to the right, agree=0.662, adj=0.129, (0 split)
##       chol     < 268.5 to the right, agree=0.650, adj=0.097, (0 split)
##       cp       < 2.5   to the right, agree=0.644, adj=0.081, (0 split)
##
## Node number 4: 90 observations
```

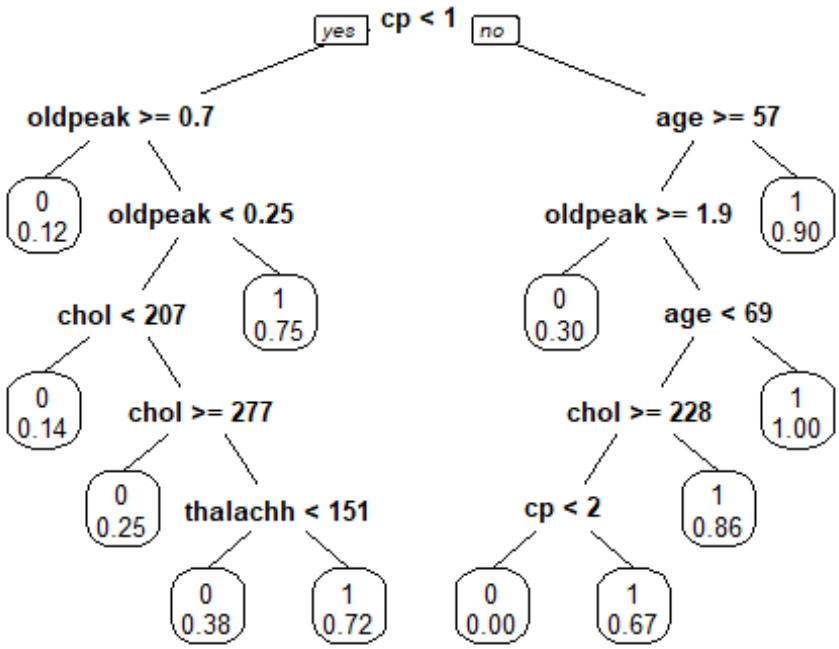
```
##   predicted class=0  expected loss=0.1222222  P(node) =0.2970297
##     class counts:    79    11
##    probabilities: 0.878 0.122
##
## Node number 5: 53 observations,    complexity param=0.02173913
##   predicted class=1  expected loss=0.4716981  P(node) =0.1749175
##     class counts:    25    28
##    probabilities: 0.472 0.528
##   left son=10 (41 obs) right son=11 (12 obs)
##   Primary splits:
##       oldpeak  < 0.25  to the left,  improve=1.5248500, (0 missing)
##       chol     < 272.5 to the right, improve=1.1793220, (0 missing)
##       trtbps   < 141   to the right, improve=0.9492558, (0 missing)
##       age      < 44.5  to the left,  improve=0.8241852, (0 missing)
##       thalachh < 174   to the left,  improve=0.5579515, (0 missing)
##   Surrogate splits:
##       age    < 64.5  to the left,  agree=0.792, adj=0.083, (0 split)
##       trtbps < 107   to the right, agree=0.792, adj=0.083, (0 split)
##
## Node number 6: 62 observations,    complexity param=0.01449275
##   predicted class=1  expected loss=0.3870968  P(node) =0.2046205
##     class counts:    24    38
##    probabilities: 0.387 0.613
##   left son=12 (10 obs) right son=13 (52 obs)
##   Primary splits:
##       oldpeak < 1.9   to the right, improve=2.3347390, (0 missing)
##       age     < 59.5  to the left,  improve=1.7102640, (0 missing)
##       trtbps  < 139   to the left,  improve=1.3017080, (0 missing)
##       chol    < 211.5 to the right, improve=1.1270200, (0 missing)
##       cp      < 1.5   to the left,  improve=0.4610215, (0 missing)
##
## Node number 7: 98 observations
##   predicted class=1  expected loss=0.1020408  P(node) =0.3234323
##     class counts:    10    88
##    probabilities: 0.102 0.898
##
## Node number 10: 41 observations,    complexity param=0.02173913
##   predicted class=0  expected loss=0.4634146  P(node) =0.1353135
##     class counts:    22    19
##    probabilities: 0.537 0.463
##   left son=20 (7 obs) right son=21 (34 obs)
##   Primary splits:
##       chol     < 206.5 to the left,  improve=1.734782000, (0 missing)
##       age      < 57.5  to the right, improve=1.342625000, (0 missing)
##       thalachh < 174   to the left,  improve=1.062513000, (0 missing)
##       trtbps   < 127   to the left,  improve=0.639047700, (0 missing)
##       oldpeak  < 0.05  to the right, improve=0.008299458, (0 missing)
##
## Node number 11: 12 observations
##   predicted class=1  expected loss=0.25  P(node) =0.03960396
```

```
##     class counts:     3     9
##    probabilities: 0.250 0.750
##
## Node number 12: 10 observations
##   predicted class=0  expected loss=0.3  P(node) =0.0330033
##     class counts:     7     3
##    probabilities: 0.700 0.300
##
## Node number 13: 52 observations,    complexity param=0.01449275
##   predicted class=1  expected loss=0.3269231  P(node) =0.1716172
##     class counts:    17    35
##    probabilities: 0.327 0.673
##   left son=26 (45 obs) right son=27 (7 obs)
##   Primary splits:
##       age     < 68.5  to the left,  improve=1.7290600, (0 missing)
##       chol    < 228   to the right, improve=1.5801110, (0 missing)
##       cp      < 1.5   to the left,  improve=1.1477730, (0 missing)
##       trtbps  < 122   to the right, improve=0.8012821, (0 missing)
##       oldpeak < 1.15  to the right, improve=0.6282051, (0 missing)
##
## Node number 20: 7 observations
##   predicted class=0  expected loss=0.1428571  P(node) =0.02310231
##     class counts:     6     1
##    probabilities: 0.857 0.143
##
## Node number 21: 34 observations,    complexity param=0.02173913
##   predicted class=1  expected loss=0.4705882  P(node) =0.1122112
##     class counts:    16    18
##    probabilities: 0.471 0.529
##   left son=42 (8 obs) right son=43 (26 obs)
##   Primary splits:
##       chol     < 276.5 to the right, improve=1.6334840, (0 missing)
##       thalachh < 151   to the left,  improve=0.9411765, (0 missing)
##       age      < 57.5  to the right, improve=0.8937456, (0 missing)
##       trtbps   < 135   to the left,  improve=0.6987522, (0 missing)
##       oldpeak  < 0.05  to the right, improve=0.4988688, (0 missing)
##
## Node number 26: 45 observations,    complexity param=0.01449275
##   predicted class=1  expected loss=0.3777778  P(node) =0.1485149
##     class counts:    17    28
##    probabilities: 0.378 0.622
##   left son=52 (31 obs) right son=53 (14 obs)
##   Primary splits:
##       chol     < 228   to the right, improve=2.2431130, (0 missing)
##       cp       < 1.5   to the left,  improve=1.9469990, (0 missing)
##       thalachh < 132   to the left,  improve=1.8773600, (0 missing)
##       trtbps   < 151   to the right, improve=1.2698410, (0 missing)
##       age      < 59.5  to the left,  improve=0.8962963, (0 missing)
##   Surrogate splits:
##       thalachh < 172.5 to the left,  agree=0.711, adj=0.071, (0 split)
```

```
##
## Node number 27: 7 observations
##   predicted class=1  expected loss=0  P(node) =0.02310231
##     class counts:     0     7
##    probabilities: 0.000 1.000
##
## Node number 42: 8 observations
##   predicted class=0  expected loss=0.25  P(node) =0.02640264
##     class counts:     6     2
##    probabilities: 0.750 0.250
##
## Node number 43: 26 observations,    complexity param=0.01449275
##   predicted class=1  expected loss=0.3846154  P(node) =0.08580858
##     class counts:    10    16
##    probabilities: 0.385 0.615
##   left son=86 (8 obs) right son=87 (18 obs)
##   Primary splits:
##       thalachh < 151   to the left,  improve=1.3354700, (0 missing)
##       oldpeak  < 0.05  to the right, improve=1.3354700, (0 missing)
##       age      < 51.5  to the right, improve=0.8076923, (0 missing)
##       chol     < 228   to the right, improve=0.4188034, (0 missing)
##       trtbps   < 135   to the left,  improve=0.2326923, (0 missing)
##   Surrogate splits:
##       oldpeak < 0.05  to the right, agree=0.769, adj=0.250, (0 split)
##       age     < 63    to the right, agree=0.731, adj=0.125, (0 split)
##       trtbps  < 141   to the right, agree=0.731, adj=0.125, (0 split)
##
## Node number 52: 31 observations,    complexity param=0.01449275
##   predicted class=1  expected loss=0.483871  P(node) =0.1023102
##     class counts:    15    16
##    probabilities: 0.484 0.516
##   left son=104 (7 obs) right son=105 (24 obs)
##   Primary splits:
##       cp       < 1.5   to the left,  improve=4.817204, (0 missing)
##       age      < 59.5  to the left,  improve=2.020235, (0 missing)
##       oldpeak  < 1.1   to the right, improve=1.527349, (0 missing)
##       thalachh < 143.5 to the left,  improve=1.527349, (0 missing)
##       chol     < 311   to the left,  improve=1.179523, (0 missing)
##   Surrogate splits:
##       age      < 57.5  to the left,  agree=0.839, adj=0.286, (0 split)
##       thalachh < 122.5 to the left,  agree=0.806, adj=0.143, (0 split)
##
## Node number 53: 14 observations
##   predicted class=1  expected loss=0.1428571  P(node) =0.04620462
##     class counts:     2    12
##    probabilities: 0.143 0.857
##
## Node number 86: 8 observations
##   predicted class=0  expected loss=0.375  P(node) =0.02640264
##     class counts:     5     3
```

```
##    probabilities: 0.625 0.375
##
## Node number 87: 18 observations
##   predicted class=1  expected loss=0.2777778  P(node) =0.05940594
##     class counts:     5    13
##    probabilities: 0.278 0.722
##
## Node number 104: 7 observations
##   predicted class=0  expected loss=0  P(node) =0.02310231
##     class counts:     7     0
##    probabilities: 1.000 0.000
##
## Node number 105: 24 observations
##   predicted class=1  expected loss=0.3333333  P(node) =0.07920792
##     class counts:     8    16
##    probabilities: 0.333 0.667
```

```
#visualization of decision tree
prp(dectree, extra=6, xpd=TRUE, cex=0.8)
```



- There are total 12 terminal nodes and 93 internal nodes in dectree model
- The decision tree started from most important variable cp with conidtion(less than 1).
- If cp is less than 1 then we check oldpeak>=0.7. If it is true then it classifies into X0(means person has heart attack) and the probability of this class is 0.12.

- If it is less than 0.7 then we check second condition if oldpeak is less than 0.25. if this condition is false then it classifies into X1(means person had heart attack). the probability of this class is 0.75
- similarly when CP >1, we check age>=57, if this condition false then it classifies into X1 with probability of 0.90

3

Compare your decision tree model and your logistic regression model. Do they attribute high importance to the same factors? Interpret each model to explain the heart attack occurrence.

```
#Variable importance from decision tree model
imp_var <- as.data.frame(dectree$variable.importance)
imp_var
```

```
##          dectree$variable.importance
## cp                         45.334660
## oldpeak                    25.506029
## thalachh                   21.505633
## age                        15.158733
## chol                        9.566919
## trtbps                      4.306793
```

- The dataframe “imp_var” contains most significant features, the importance of variables in decreasing order are : cp,oldpeak,thalachh,age,chol and trtbps.
- lets now check with variable importance of logistic regression based on AIC that we have done in 1.c The best variable importance in decreasing order in one input model are: oldpeak,cp,thalachh,age,trtbps and chol
- As you see in decision tree, CP is valued has highest importance and strong effect on dependent variable where as in logistic regression CP is second importance and first is oldpeak. But if we compare combination of best two and three input i.e, cp+oldpeak or cp+oldpeak+thalachh both models agree on this.
- From logistic regression model, the age+chol+trtbps has least importance on heart attach target variable as compared to combination of cp+oldpeak+thalachh.This combination of importance also agrees with decision tree variable importance as you can see in last three rows in “imp_var” data frame
- The lowest important factor in regression model is chol(in terms of AIC) and in decision tree least important factor is trtbps which contradicts each other on level of importance order.

4

Which model would you use if you were explaining the heart attack data to an audience, and why?

```r
#lets build training and test set for logistic regression and find out
accuracy of test test.
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

#install.packages('e1071', dependencies=TRUE)
data<-heart_attack
set.seed(1237)
train <- sample(nrow(data), .8*nrow(data), replace = FALSE)
TrainSet <- data[train,]
ValidSet <- data[-train,]
dim(TrainSet)

## [1] 242   7

dim(ValidSet)

## [1] 61  7

#Tuning parameters
fitControl <- trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 10,
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)

TrainSet$Hattack<-make.names(TrainSet$Hattack)
set.seed(6000)
# Logistic Regression with the train function in caret package
gbm<- caret::train(Hattack ~ .,
                   data = TrainSet ,
                   method = "glm",
                   trControl = fitControl,
                   metric="ROC")

gbm

## Generalized Linear Model
##
## 242 samples
##   6 predictor
##   2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 218, 218, 218, 218, 218, 219, ...
## Resampling results:
##
```

```
##    ROC        Sens       Spec
##    0.8513252  0.7003636  0.8559341

#  we predict on the Test Set.
pred <- predict(gbm,ValidSet)
t<-table(pred, ValidSet$Hattack)
t.df<-as.data.frame(t)
#Plotting the Confusion Matrix for Logistic Regression
#install.packages("ggthemes")
library(ggthemes)
logisticplot =ggplot(data = t.df, aes(x = Var2, y = pred, label=Freq)) +
  geom_tile(aes(fill = Freq)) +
  scale_fill_gradient(low="green", high="blue") +
  theme_economist()+
  xlab("Actual Hattack") +
  ylab("Predicted Hattack") +
  geom_text(size=8) +
  ggtitle("Logistic Regression confusion matrix")
logisticplot + theme_bw()
```

From confusion matrix the accuracy is: accuracy = TP+TN/TP+TN+FP+FN = 20+21/61=0.67

```
# Decision tree train set
gbmGrid <-  expand.grid(cp=c(0.01))
TrainSet$Hattack<-make.names(TrainSet$Hattack)
system.time(decstree <- caret::train(Hattack ~ .,
```

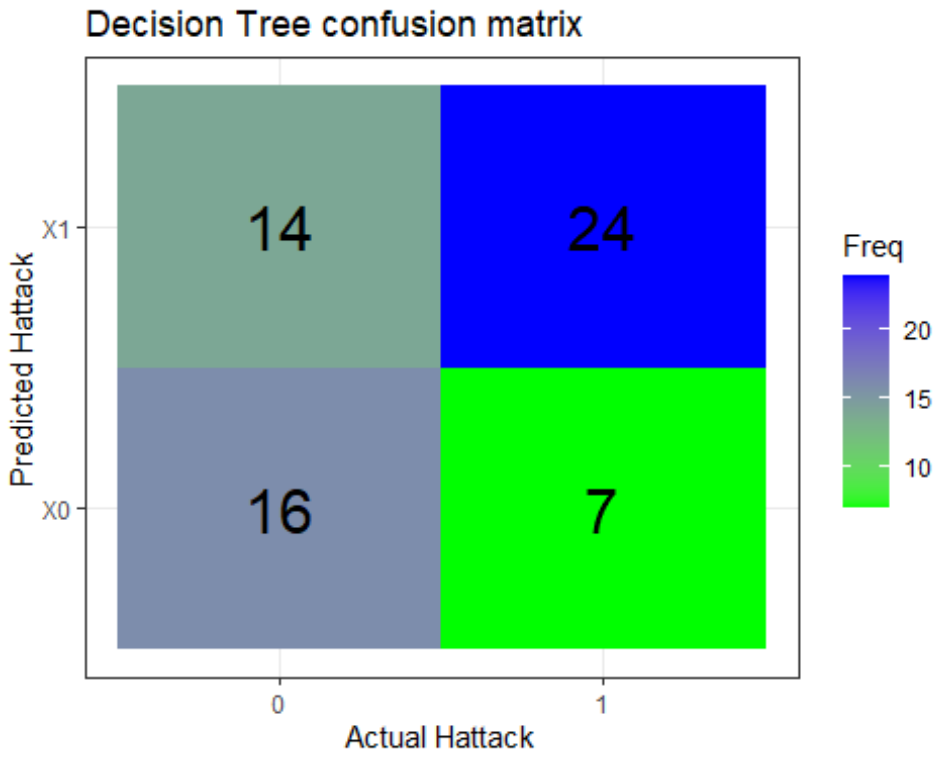
```
                                    data = TrainSet,
                                    method = "rpart",
                                    trControl = fitControl,
                                    metric="ROC",
                                    tuneGrid=gbmGrid))
```

```
##    user  system elapsed
##    1.50    0.02    1.53
```

```
decstree
```

```
## CART
##
## 242 samples
##   6 predictor
##   2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 219, 218, 217, 218, 217, 218, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.8060832  0.6656364  0.801044
##
## Tuning parameter 'cp' was held constant at a value of 0.01
```

```
#  we predict on the Test Set.
pred <- predict(decstree,ValidSet)
t<-table(pred, ValidSet$Hattack)
t.df<-as.data.frame(t)
#plotting confusion matrix for decision tree
regressionplot =ggplot(data = t.df, aes(x = Var2, y = pred, label=Freq)) +
  geom_tile(aes(fill = Freq)) +
  scale_fill_gradient(low="green", high="blue") +
  theme_economist()+
  xlab("Actual Hattack") +
  ylab("Predicted Hattack") +
  geom_text(size=8) +
  ggtitle("Decision Tree confusion matrix")
regressionplot + theme_bw()
```



From confusion matrix the accuracy is: accuracy = TP+TN/TP+TN+FP+FN = 16+24/61=0.65

Based on accuracy results, the logistic regression has little upper hand than decision tree test results.

we can also observe that the ROC(receiver operating characteristic curve) for regression model is more than decision tree model

- ROC for logistic regression = 0.8513252
- ROC for decision tree = 0.8060832

ROC is a probability curve which is capable of distinguishing between classes. so higher the ROC better the model. Based on above results and interpretation we must choose logistic regression model over decision tree.