Exercise 6

This patients' (heart attack diagnosis) dataset (Patient_data.xlsx) was retrieved from the Internet. But I forgot about its source. We should give the provider/publisher credits for their efforts (if you find its source, please let me know).

The file contains 7998 records. The following screenshot shows you what the dataset actually contains.

	A	B	C	D	E	F	G	H	I
1	age	gender	diabetes	smoker	active	obesity	heartattac	bp	cholesteral
2	54	Female	No	No	Yes	No	No	Hypertension	Normal
3	64	Female	No	No	No	No	Yes	Normal	Normal
4	63	Female	No	No	No	No	Yes	Normal	Highl
5	67	Male	No	Yes	No	No	No	Hypotension	Highl
6	76	Male	No	No	No	No	No	Hypotension	Normal
7	69	Male	No	No	No	No	No	Normal	Normal
8	67	Male	Yes	Yes	Yes	No	Yes	Hypertension	Normal
9	74	Male	No	No	No	Yes	Yes	Normal	Normal
10	69	Male	No	Yes	Yes	No	No	Normal	Highl
11	54	Female	No	Yes	No	No	Yes	Normal	Highl
12	57	Male	No	No	Yes	No	No	Normal	Normal
13	49	Female	No	No	Yes	No	No	Normal	Normal
14	66	Female	No	Yes	No	Yes	Yes	Normal	Normal
15	51	Female	No	No	No	Yes	No	Hypertension	Highl
16	63	Male	No	Yes	Yes	No	Yes	Normal	Highl
17	71	Female	No	Yes	Yes	No	Yes	Normal	Normal
18	70	Female	No	No	Yes	No	No	Normal	Normal
19	76	Male	No	No	Yes	No	No	Normal	Highl
20	50	Male	No	Yes	No	No	Yes	Normal	Normal

1. (50 points) Explore the data set, then use C5.0 to model this classification problem (no partition at this step).

```
# Installing and loading all the libraries
#install.packages("rattle")
#install.packages("polycor")
library(polycor)
library(readxl)
library("readxl")
library('C50')
library(rpart)
library(caret)
library(rattle)
library(psych)#categorical correlation
```

```
df <- read_excel("R://downloads//Patient_Data.xlsx")
#now we inspect data to see each variable
str(df)
#since variable type is char, we change it to factor for all the applicable vairables
df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)],as.factor)
#lets check if they are converted to factors
str(df)
```

From the initial inspection we can see that all the variables except age was given the type as chr, hence I converted all of them to factors for visualization and analysis purposes.

```
> library(psych)#categorical correlation
> df <- read_excel("R://downloads//Patient_Data.xlsx")
> #now we inspect data to see each variable
> str(df)
tibble [7,998 x 9] (S3: tbl_df/tbl/data.frame)
 $ age          : num [1:7998] 54 64 63 67 76 69 67 74 69 54 ...
 $ gender       : chr [1:7998] "Female" "Female" "Female" "Male" ...
 $ diabetes     : chr [1:7998] "No" "No" "No" "No" ...
 $ smoker       : chr [1:7998] "No" "No" "No" "Yes" ...
 $ active       : chr [1:7998] "Yes" "No" "No" "No" ...
 $ obesity      : chr [1:7998] "No" "No" "No" "No" ...
 $ heartattack_s: chr [1:7998] "No" "Yes" "Yes" "No" ...
 $ bp           : chr [1:7998] "Hypertension" "Normal" "Normal" "Hypotension" ...
 $ cholesteral  : chr [1:7998] "Normal" "Normal" "Highl" "Highl" ...
> #since variable type is char, we change it to factor for all the applicable vairables
> df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)],as.factor)
> #lets check if they are converted to factors
> str(df)
tibble [7,998 x 9] (S3: tbl_df/tbl/data.frame)
 $ age          : num [1:7998] 54 64 63 67 76 69 67 74 69 54 ...
 $ gender       : Factor w/ 2 levels "Female","Male": 1 1 1 2 2 2 2 2 2 1 ...
 $ diabetes     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 1 ...
 $ smoker       : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 1 2 2 ...
 $ active       : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 2 1 2 1 ...
 $ obesity      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
 $ heartattack_s: Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 2 2 1 2 ...
 $ bp           : Factor w/ 3 levels "Hypertension",..: 1 3 3 2 2 3 1 3 3 3 ...
 $ cholesteral  : Factor w/ 2 levels "Highl","Normal": 2 2 1 1 2 2 2 2 1 1 ...
> |
```

```
#Lets summarize our data
summary(df)
```

From the summary we can say that the dataset contains total 9 variables with only one numeric variable, 7 binary categories and 1 category variable with 3 categories.
From summary we can also observe that the gender almost evenly split same even distribution applies for attribute active as well. Diabetes has most in the no category. Smoker most have answered no. Obesity has the majority categorized as a 'no'. Bp variable has normal, but hypertension has the second highest, followed by hypotension. The cholesterol variable is close to even too.

```
> summary(df)
      age            gender      diabetes     smoker       active       obesity      heartattack_s             bp          cholesteral
 Min.   :45.00   Female:3959   No :7456    No :6346    No :3858    No :6230    No :4491      Hypertension:2011   Highl :3064
 1st Qu.:55.00   Male  :4039   Yes: 542    Yes:1652    Yes:4140    Yes:1768    Yes:3507      Hypotension : 985   Normal:4934
 Median :61.00                                                                               Normal      :5002
 Mean   :61.85
 3rd Qu.:68.00
 Max.   :89.00
> |
```

Now we check the correlation between the binary categorical variables with respect to our dependent variable heartattach_s.
The correlation between binary category variables is calculated using tetrachoric correlation. This test is only performed between variables that have just two potential values.

A tetrachoric correlation can have a value ranging from -1 to 1, where:

- A high negative correlation between the two variables is indicated by a value of -1.
- There is no association between the two variables if the value is 0.
- A significant positive correlation between the two variables is indicated by a value of 1.

```
# to check correlation between each binary categorical data
cc=table(df$diabetes,df$heartattack_s)
tetrachoric(cc)
cc=table(df$gender,df$heartattack_s)
tetrachoric(cc)
cc=table(df$smoker,df$heartattack_s)
tetrachoric(cc)
cc=table(df$active,df$heartattack_s)
tetrachoric(cc)
cc=table(df$obesity,df$heartattack_s)
tetrachoric(cc)
cc=table(df$cholesteral,df$heartattack_s)
tetrachoric(cc)
cc=table(df$bp,df$heartattack_s)
tetrachoric(cc)
```

```
library(psych)
cc=table(df$diabetes,df$heartattack_s)
tetrachoric(cc)

cc=table(df$gender,df$heartattack_s)
tetrachoric(cc)

cc=table(df$smoker,df$heartattack_s)
tetrachoric(cc)

cc=table(df$active,df$heartattack_s)
tetrachoric(cc)

cc=table(df$obesity,df$heartattack_s)
tetrachoric(cc)

cc=table(df$cholesteral,df$heartattack_s)
tetrachoric(cc)
```

```
> cc=table(df$diabetes,df$heartattack_s)
> cc

        No  Yes
  No  4394 3062
  Yes   97  445
> tetrachoric(cc)
Call: tetrachoric(x = cc)
tetrachoric correlation
[1] 0.5
```

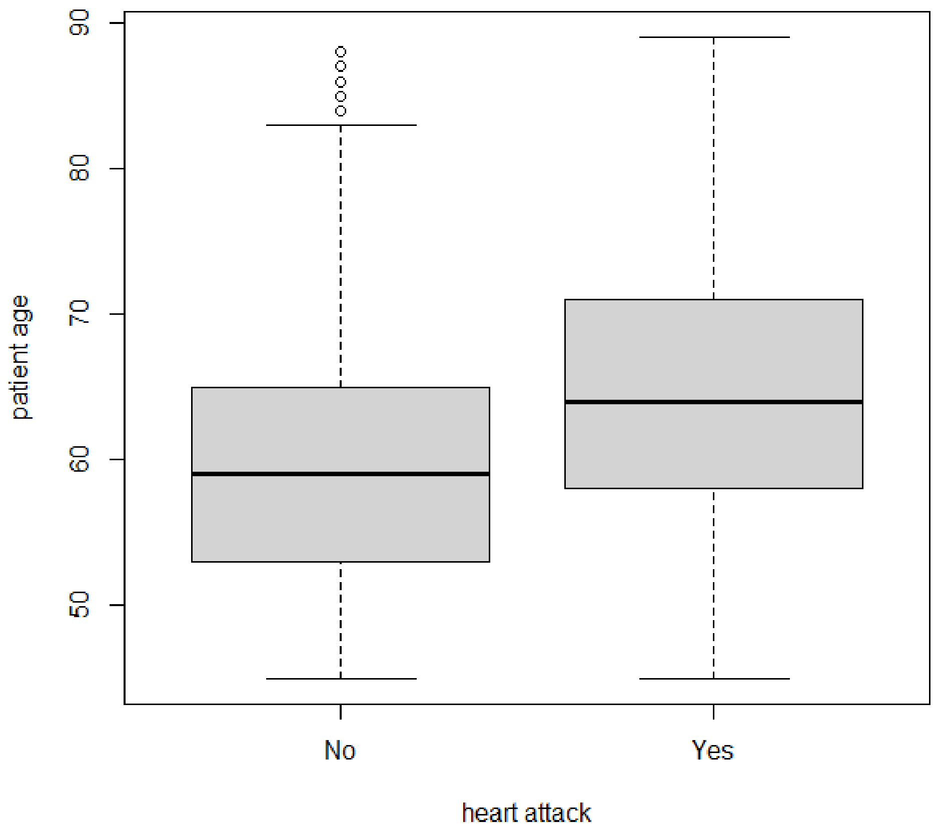
```
>
> cc=table(df$gender,df$heartattack_s)
> tetrachoric(cc)
Call: tetrachoric(x = cc)
tetrachoric correlation
[1] -0.018
```

```
>
> cc=table(df$smoker,df$heartattack_s)
> tetrachoric(cc)
Call: tetrachoric(x = cc)
tetrachoric correlation
[1] 0.39

 with tau of
  No   No
0.82 0.15
>
> cc=table(df$active,df$heartattack_s)
> tetrachoric(cc)
Call: tetrachoric(x = cc)
tetrachoric correlation
[1] -0.34

 with tau of
    No     No
-0.044  0.155
>
> cc=table(df$obesity,df$heartattack_s)
> tetrachoric(cc)
Call: tetrachoric(x = cc)
tetrachoric correlation
[1] 0.34
```

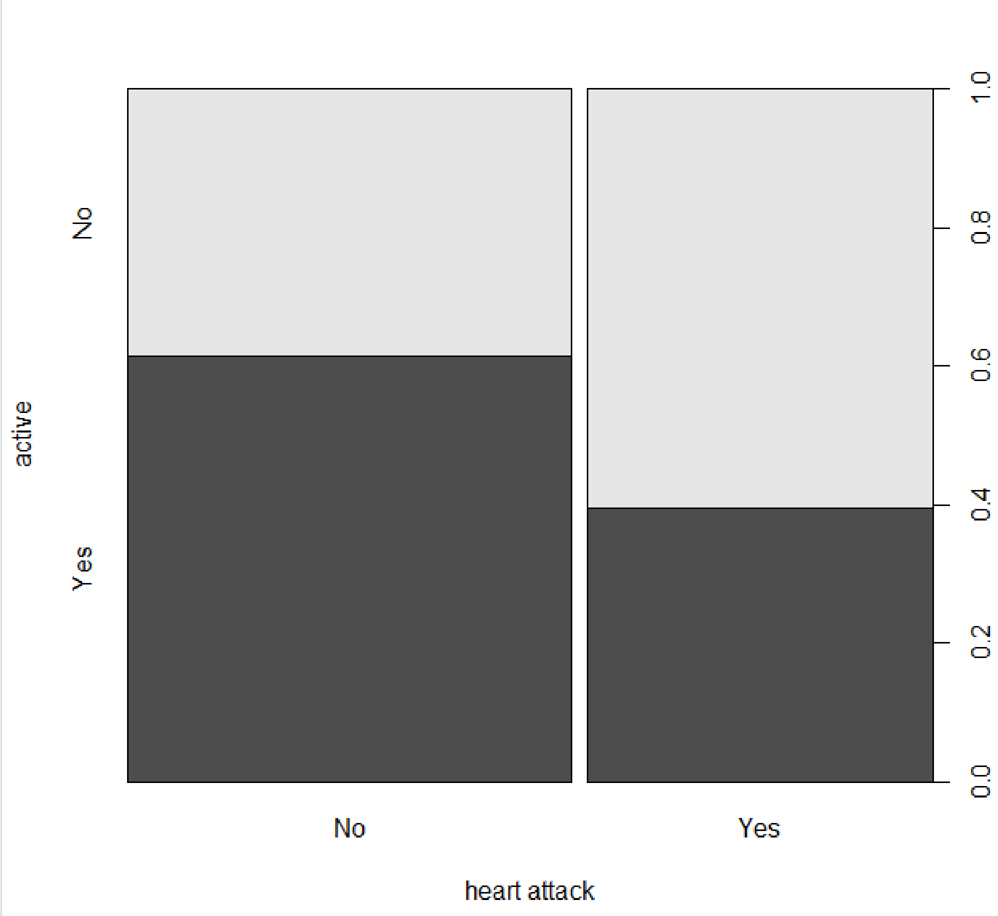
```
> cc=table(df$cholesteral,df$heartattack_s)
> tetrachoric(cc)
Call: tetrachoric(x = cc)
tetrachoric correlation
[1] -0.25

 with tau of
High1    No
-0.30  0.15
> |
```

From the above correlations we can see that some categorical variables have medium negative correlation with respect to dependent variable and some have medium positive correlation, with diabetes having highest positive correlation of 0.5 , active having lowest negative correlation of -0.34 and gender having almost no correlation (close to 0).

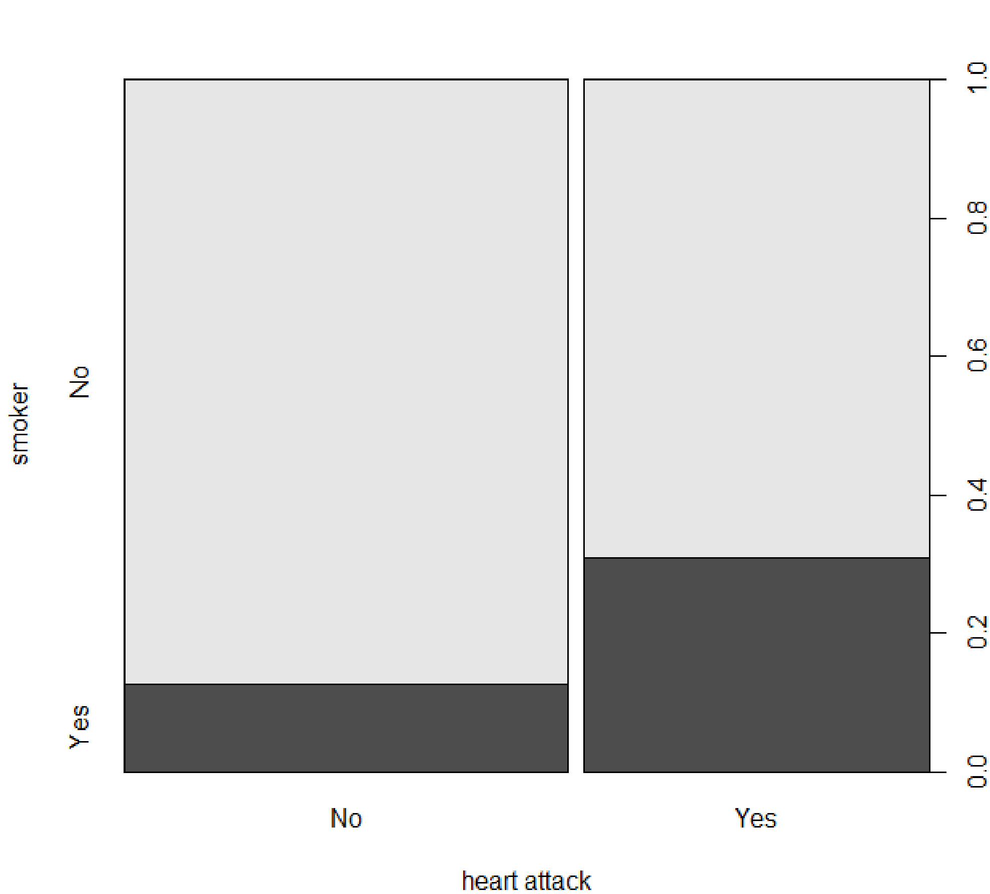
```
#EDA

plot(df$heartattack_s,df$age, xlab="heart attack", ylab="patient age")
plot(df$heartattack_s,df$active, xlab="heart attack", ylab="active")
plot(df$heartattack_s,df$smoker, xlab="heart attack", ylab="smoker")
plot(df$heartattack_s,df$gender, xlab="heart attack", ylab="gender")

#outliers in age variable
```



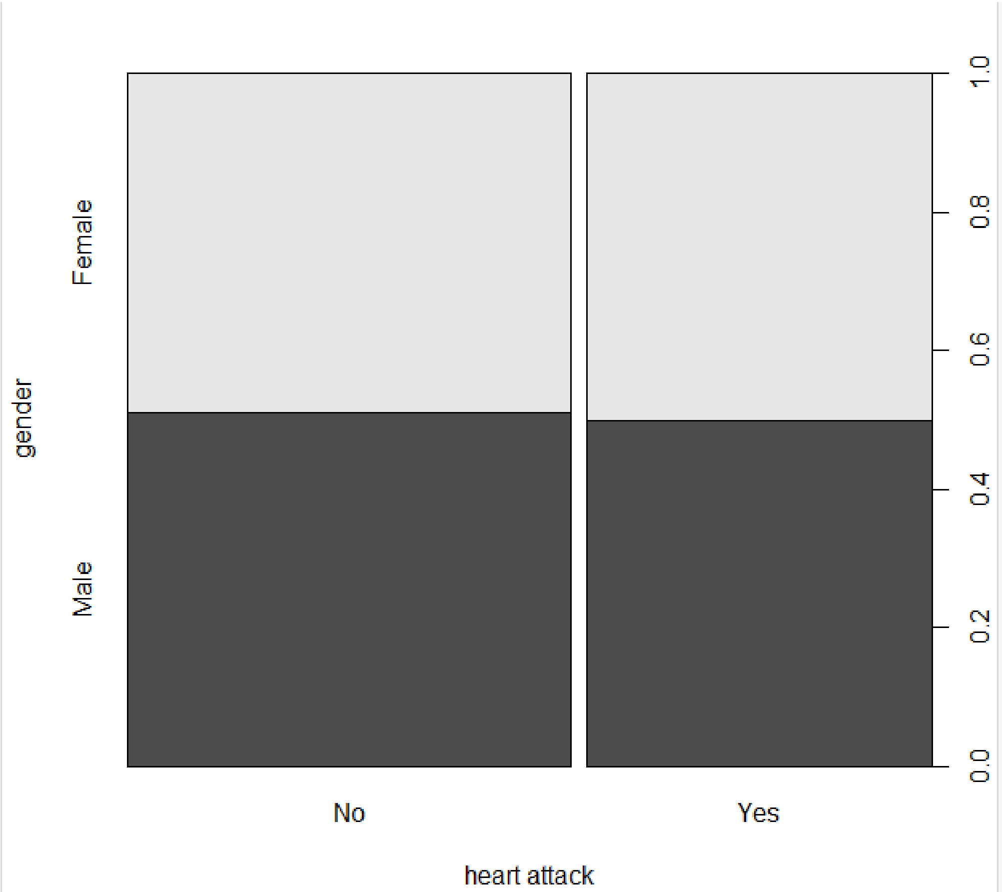
First I have plotted heart attacks wrt to age. For patients with no heart attack, the median age is 58 or 59, whereas for patients who got heart attack the median age is higher, around 62. There also appears to be a few outliers in age with no heart attack. From this we can see that as the age goes up, so does the probability of having a heart attack, hence there is small positive correlation between the two.



As seen in correlation active having lowest negative correlation, we can confirm that with this graph as more the patient is active the lesser the chances of them having a heart attack.

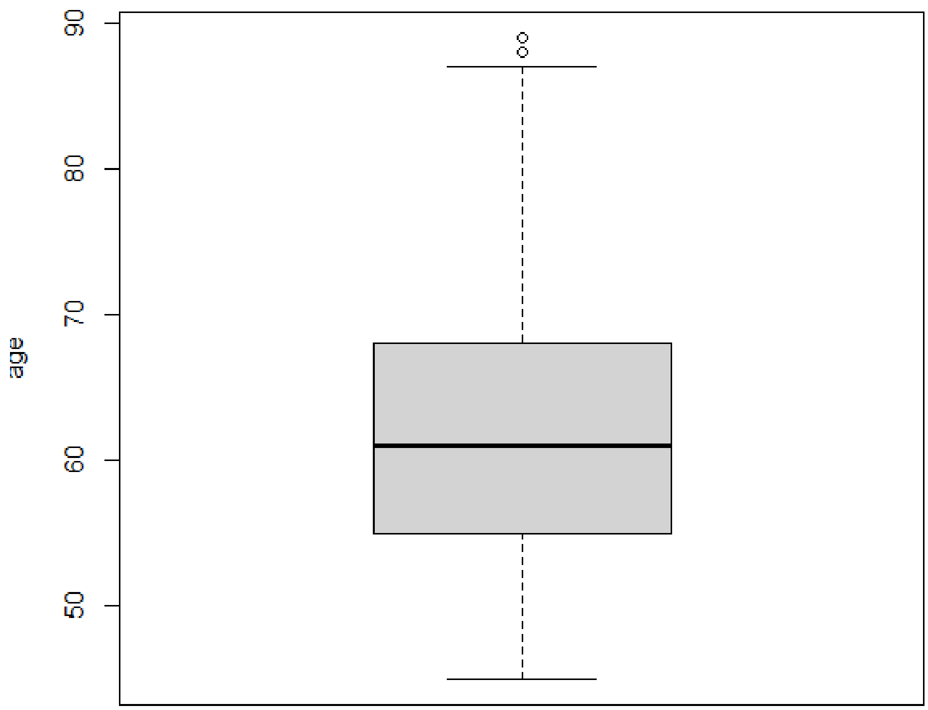


Here the plot proves our previous point that there is slight positive correlation between the the smoker and heart attack as the chances of having a heart attack is higher for patients that smoke.



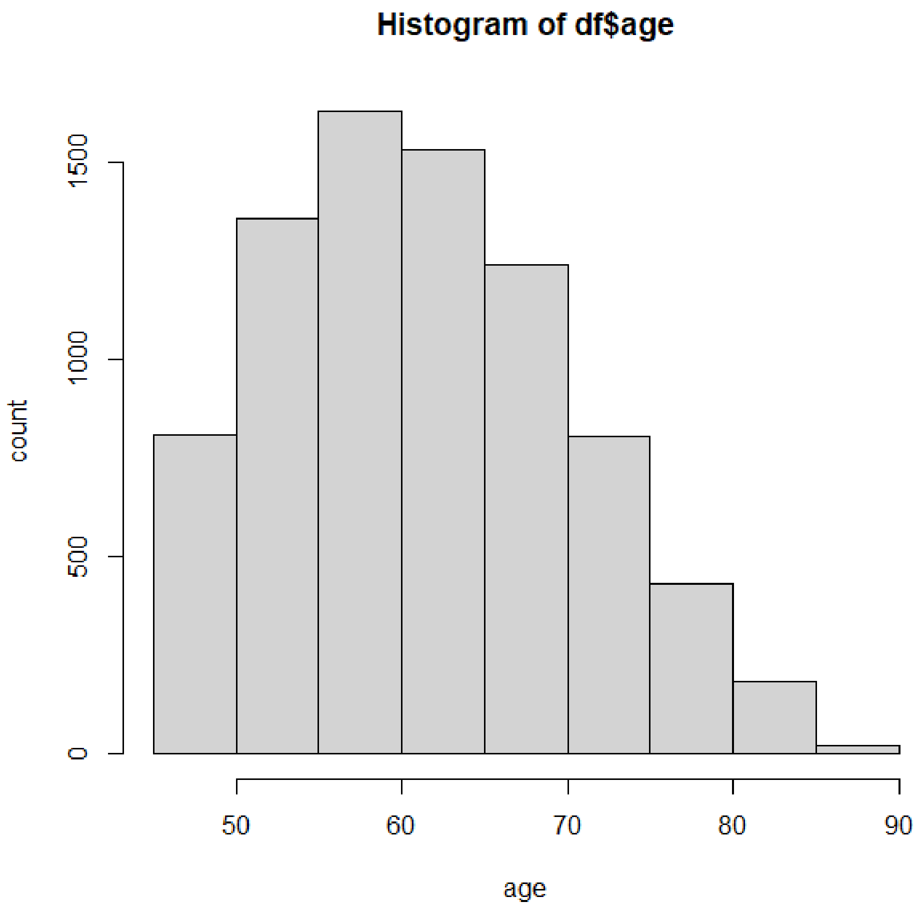
As shown in correlation plot, the gender hardly has any correlation with respect to heart attack. This graph further proves our point.

```
#outliers in age variable
boxplot(df$age)
```



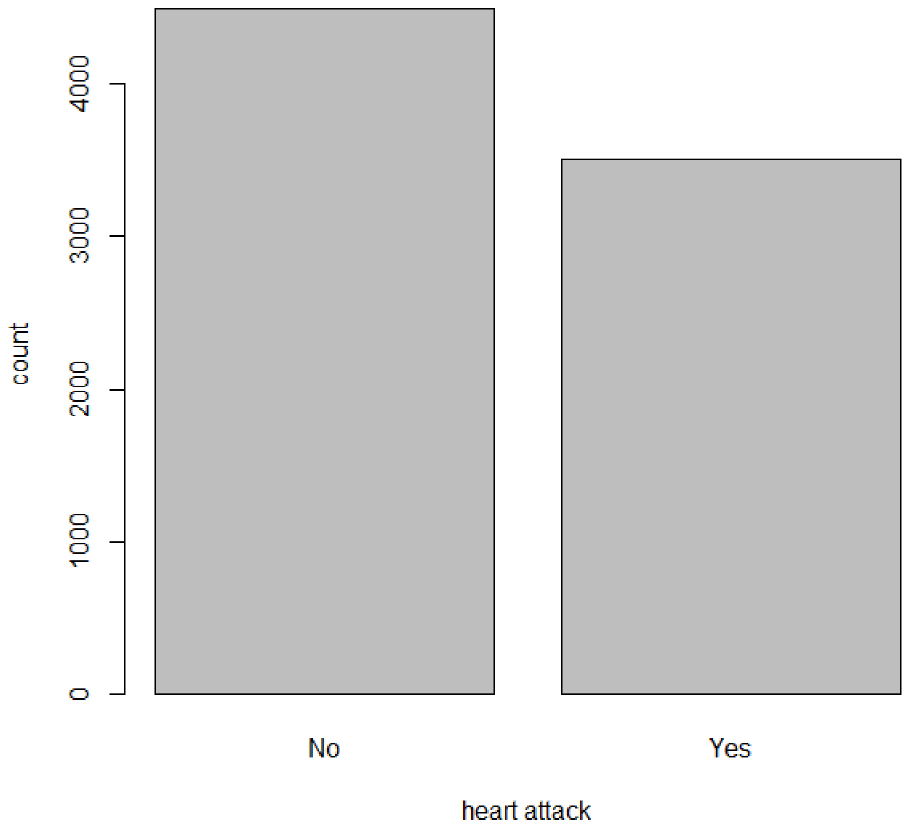
As we had seen some outliers in the age variable, I plotted this boxplot to see there are few outliers (patients with age near to 90), we won't be pruning them as they might hold some valuable information and are not that large in number. Since we are using tree-based methods, they are robust to outliers.

```
#skewness of age variable
hist(df$age,ylab="count",xlab="age")
```



The age variable is slightly right skewed here. Since we are making decision trees, I would not like to normalize and change the age values as the raw values would be better for interpretation.

```
#checking if the data is balanced
barplot(table(df$heartattack_s))
```



From this graph we can observe that the dependent variable (heart attack) is slightly imbalanced here as the number of heart attacks are quite low as compared to patients with no heart attacks. We will be trying to train a model with balanced heart attack attribute as well (please refer the last model in assignment)

```
#to check if there are any duplicate values
#duplicated(df)
sum(duplicated(df))
nrow(df)
df2 =unique(df)
nrow(df2)
```

```
> #to check if there are any duplicate values
> #duplicated(df)
> sum(duplicated(df))
[1] 4492
> nrow(df)
[1] 7998
> df2 =unique(df)
> nrow(df2)
[1] 3506
```

From this we can observe that there are a lot of duplicate values (4492 rows). Once we prune them we will be only left with 3,506 rows from the original data.

```
#check for missing values
sum(is.na(df))
```

```
> #check for missing values
> sum(is.na(df))
[1] 0
```

As seen in the summary there are no missing values in the dataset.

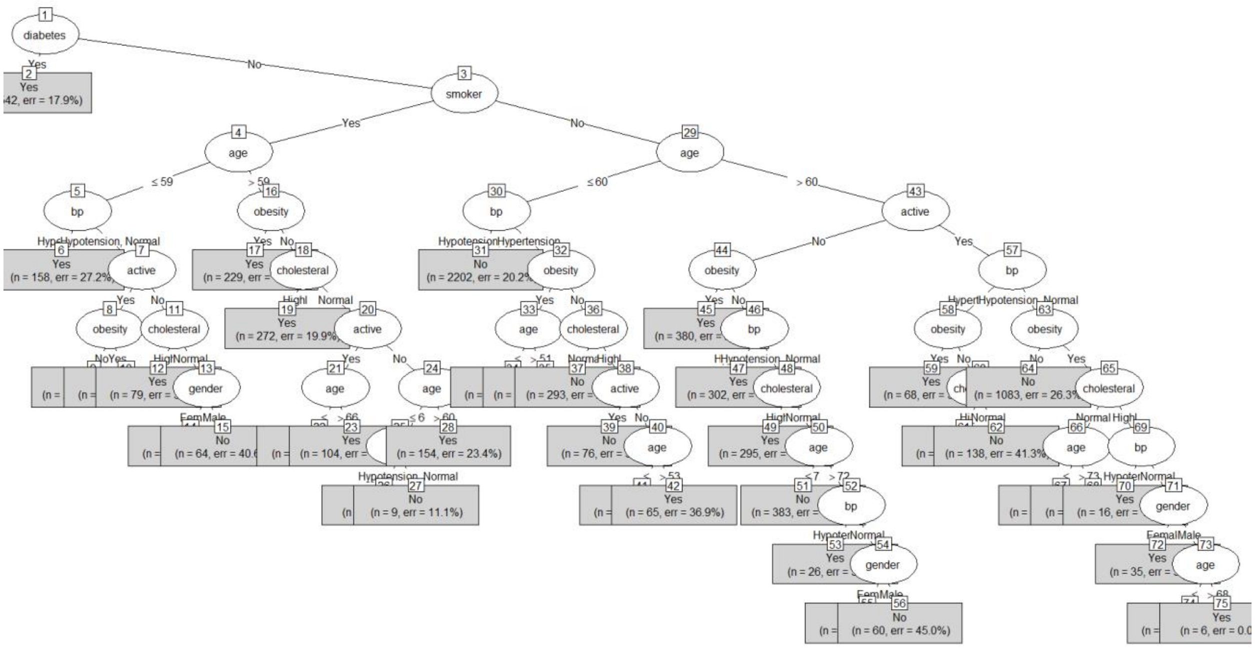
```
#building the c5.0 model
c50tree1 <- C5.0(df[,-7], as.factor(df$heartattack_s))
#summary
summary(c50tree1)
```

```
> summary(c50tree1)

Call:
C5.0.default(x = df[, -7], y = as.factor(df$heartattack_s))


C5.0 [Release 2.07 GPL Edition]         Sat Apr 09 17:42:20 2022
-------------------------------

Class specified by attribute `outcome'

Read 7998 cases (9 attributes) from undefined.data

Decision tree:

diabetes = Yes: Yes (542/97)
diabetes = No:
:...smoker = Yes:
    :...age <= 59:
    :   :...bp = Hypertension: Yes (158/43)
    :   :   bp in {Hypotension,Normal}:
    :   :   :...active = Yes:
    :   :       :...obesity = No: No (262/75)
    :   :       :   obesity = Yes: Yes (42/17)
    :   :       active = No:
    :   :       :...cholesteral = Highl: Yes (79/27)
    :   :           cholesteral = Normal:
    :   :           :...gender = Female: Yes (60/26)
    :   :               gender = Male: No (64/26)
    :   age > 59:
    :   :...obesity = Yes: Yes (229/26)
    :       obesity = No:
    :       :...cholesteral = Highl: Yes (272/54)
    :           cholesteral = Normal:
    :           :...active = Yes:
    :               :...age <= 66: No (90/34)
    :               :   age > 66: Yes (104/39)
    :               active = No:
    :               :...age > 60: Yes (154/36)
    :                   age <= 60:
    :                   :...bp = Hypertension: Yes (3)
    :                       bp in {Hypotension,Normal}: No (9/1)
    smoker = No:
    :...age <= 60:
        :...bp in {Hypotension,Normal}: No (2202/445)
        :   bp = Hypertension:
        :   :...obesity = Yes:
        :       :...age <= 51: No (45/16)
        :       :   age > 51: Yes (121/44)
        :       obesity = No:
        :       :...cholesteral = Normal: No (293/64)
        :           cholesteral = Highl:
        :           :...active = Yes: No (76/29)
        :               active = No:
        :               :...age <= 53: No (51/19)
        :                   age > 53: Yes (65/24)
        age > 60:
        :...active = No:
```

```
age > 60:
:...active = No:
    :...obesity = Yes: Yes (380/93)
    :   obesity = No:
    :   :...bp = Hypertension: Yes (302/100)
    :       bp in {Hypotension,Normal}:
    :       :...cholesteral = High1: Yes (295/124)
    :           cholesteral = Normal:
    :           :...age <= 72: No (383/136)
    :               age > 72:
    :               :...bp = Hypotension: Yes (26/8)
    :                   bp = Normal:
    :                   :...gender = Female: Yes (62/28)
    :                       gender = Male: No (60/27)
    active = Yes:
    :...bp = Hypertension:
        :...obesity = Yes: Yes (68/21)
        :   obesity = No:
        :   :...cholesteral = High1: Yes (78/32)
        :       cholesteral = Normal: No (138/57)
        bp in {Hypotension,Normal}:
        :...obesity = No: No (1083/285)
            obesity = Yes:
            :...cholesteral = Normal:
                :...age <= 73: No (104/31)
                :   age > 73: Yes (26/10)
                cholesteral = High1:
                :...bp = Hypotension: Yes (16/2)
                    bp = Normal:
                    :...gender = Female: Yes (35/12)
                        gender = Male:
                        :...age <= 68: No (15/2)
                            age > 68: Yes (6)


Evaluation on training data (7998 cases):

	    Decision Tree   
	  ----------------  
	  Size      Errors  

	    38 2110(26.4%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	  3628   863    (a): class No
	  1247  2260    (b): class Yes


	Attribute usage:

	100.00%	diabetes
	 93.22%	age
	 93.22%	smoker
	 77.86%	bp
	 61.18%	obesity
	 51.71%	active
	 32.06%	cholesteral
	  3.78%	gender
```

```
#Now let us compute the accuracy of our model
predictions <- predict(c50tree1, df)
mean(predictions==df$heartattack_s)
```

```
> plot(c50tree1,type="simple",cex=.7)
> #Now let us compute the accuracy of our model
> predictions <- predict(c50tree1, df)
> mean(predictions==df$heartattack_s)
[1] 0.736184
> 
```



First, we use summary to see the results in texts then plot it. It is easier to understand from the text summary than the graph, as the graph is convoluted and complex (might be overfitting as well).

As we can observe diabetes is the first segregation attribute in the tree. This result is quite obvious as in our EDA we got to know diabetes had the highest correlation. It shows if someone has diabetes, they will be categorized as having a heart attack. Whereas if they don't have diabetes, they continue getting segregated down the tree. After Diabetes we can see that age and smoker were the next two top variable used for segregation, again this was quite evident from EDA. The error rate for this model is 26.4%. From the confusion matrix we can see that the false negative classification are 1247 which isn’t a good number in heart attack diagnosis. Lastly, we can also see the attribute usage, as discussed diabetes, age, smoker is on top of the list. We can see that the accuracy of our model is 73.6% for whole data.

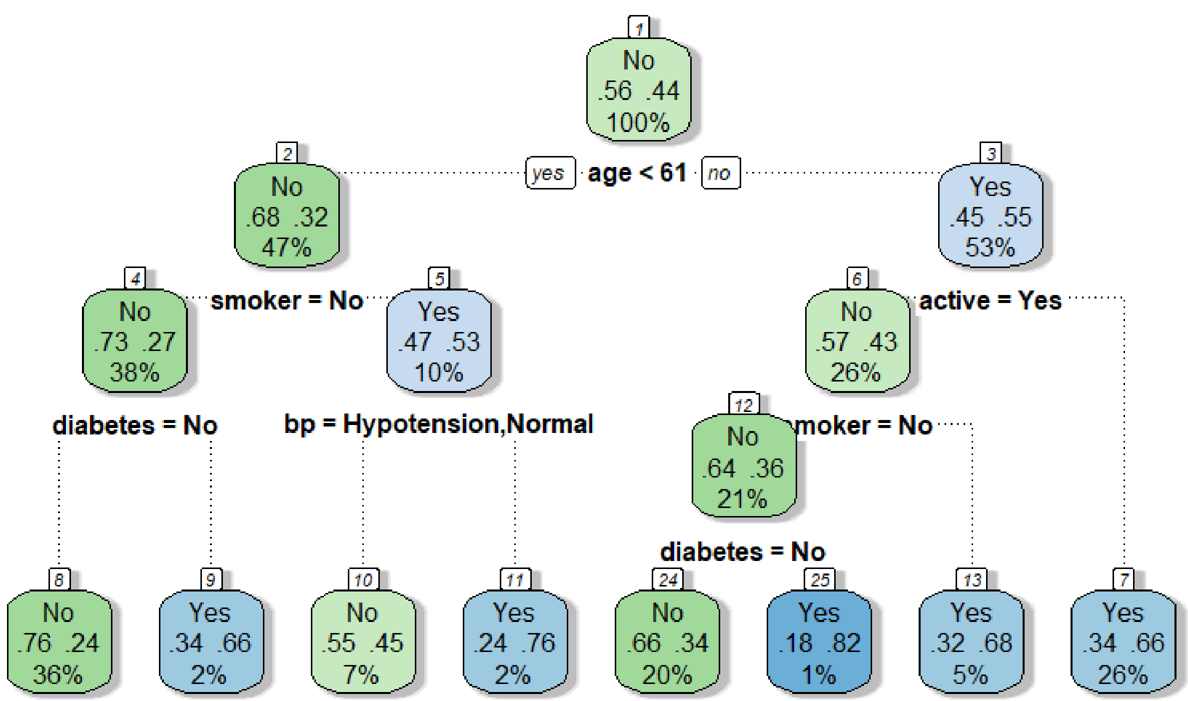
2. (30 points) Comparing with a CART model, is there any difference?

```
#building the cart model
cart <- rpart(heartattack_s~., data = df, method = "class")
summary(cart)
#plotting
fancyRpartPlot(cart, cex = 1.0, caption = "Heart Attack")
```

```
Call:
rpart(formula = heartattack_s ~ ., data = df, method = "class")
  n= 7998

          CP nsplit rel error    xerror       xstd
1 0.11149130      0 1.0000000 1.0000000 0.01265357
2 0.08525806      1 0.8885087 0.8930710 0.01244717
3 0.04334189      2 0.8032506 0.8140861 0.01221757
4 0.01824922      3 0.7599088 0.7610493 0.01202460
5 0.01454234      4 0.7416595 0.7433704 0.01195305
6 0.01000000      7 0.6980325 0.7051611 0.01178560

Variable importance
     age   smoker   active diabetes       bp  obesity 
      33       24       18       16        6        2 

Node number 1: 7998 observations,    complexity param=0.1114913
  predicted class=No   expected loss=0.4384846  P(node) =1
    class counts:  4491  3507
   probabilities: 0.562 0.438 
  left son=2 (3793 obs) right son=3 (4205 obs)
  Primary splits:
      age      < 60.5 to the left,  improve=206.8729, (0 missing)
      smoker   splits as  LR,       improve=197.3318, (0 missing)
      active   splits as  RL,       improve=188.0524, (0 missing)
      bp       splits as  RLL,      improve=180.1255, (0 missing)
      diabetes splits as  LR,       improve=170.1680, (0 missing)

Node number 2: 3793 observations,    complexity param=0.01454234
  predicted class=No   expected loss=0.3187451  P(node) =0.4742436
    class counts:  2584  1209
   probabilities: 0.681 0.319 
  left son=4 (3022 obs) right son=5 (771 obs)
  Primary splits:
      smoker   splits as  LR,  improve=83.60781, (0 missing)
      bp       splits as  RLL, improve=71.29711, (0 missing)
      diabetes splits as  LR,  improve=71.17877, (0 missing)
      active   splits as  RL,  improve=59.03512, (0 missing)
      obesity  splits as  LR,  improve=51.13655, (0 missing)

Node number 3: 4205 observations,    complexity param=0.08525806
  predicted class=Yes  expected loss=0.4535077  P(node) =0.5257564
    class counts:  1907  2298
   probabilities: 0.454 0.546 
  left son=6 (2089 obs) right son=7 (2116 obs)
  Primary splits:
      active   splits as  RL,  improve=115.71960, (0 missing)
      smoker   splits as  LR,  improve=110.93340, (0 missing)
      bp       splits as  RLL, improve=104.50390, (0 missing)
      obesity  splits as  LR,  improve= 99.09606, (0 missing)
      diabetes splits as  LR,  improve= 85.30530, (0 missing)
  Surrogate splits:
      bp       splits as  RLL,      agree=0.555, adj=0.105, (0 split)
      obesity  splits as  LR,       agree=0.549, adj=0.092, (0 split)
      age      < 65.5 to the left,  agree=0.521, adj=0.036, (0 split)
      diabetes splits as  LR,       agree=0.512, adj=0.019, (0 split)
      gender   splits as  RL,       agree=0.509, adj=0.011, (0 split)
```

```
Node number 4: 3022 observations,    complexity param=0.01454234
  predicted class=No   expected loss=0.2657181  P(node) =0.3778445
    class counts:  2219   803
   probabilities: 0.734 0.266
  left son=8 (2853 obs) right son=9 (169 obs)
  Primary splits:
      diabetes    splits as  LR,  improve=56.42853, (0 missing)
      bp          splits as  RLL, improve=46.25869, (0 missing)
      active      splits as  RL,  improve=40.62085, (0 missing)
      cholesteral splits as  RL,  improve=31.72896, (0 missing)
      obesity     splits as  LR,  improve=31.02963, (0 missing)

Node number 5: 771 observations,    complexity param=0.01454234
  predicted class=Yes  expected loss=0.4734112  P(node) =0.0963991
    class counts:   365   406
   probabilities: 0.473 0.527
  left son=10 (579 obs) right son=11 (192 obs)
  Primary splits:
      bp          splits as  RLL, improve=26.72597, (0 missing)
      active      splits as  RL,  improve=22.07201, (0 missing)
      obesity     splits as  LR,  improve=18.57453, (0 missing)
      cholesteral splits as  RL,  improve=14.88015, (0 missing)
      diabetes    splits as  LR,  improve=12.89033, (0 missing)

Node number 6: 2089 observations,    complexity param=0.04334189
  predicted class=No   expected loss=0.4284347  P(node) =0.2611903
    class counts:  1194   895
   probabilities: 0.572 0.428
  left son=12 (1669 obs) right son=13 (420 obs)
  Primary splits:
      smoker      splits as  LR,  improve=67.04172, (0 missing)
      diabetes    splits as  LR,  improve=45.94836, (0 missing)
      bp          splits as  RLL, improve=41.62284, (0 missing)
      obesity     splits as  LR,  improve=35.40730, (0 missing)
      cholesteral splits as  RL,  improve=33.77798, (0 missing)

Node number 7: 2116 observations
  predicted class=Yes  expected loss=0.3369565  P(node) =0.2645661
    class counts:   713  1403
   probabilities: 0.337 0.663

Node number 8: 2853 observations
  predicted class=No   expected loss=0.2422012  P(node) =0.3567142
    class counts:  2162   691
   probabilities: 0.758 0.242

Node number 9: 169 observations
  predicted class=Yes  expected loss=0.3372781  P(node) =0.02113028
    class counts:    57   112
   probabilities: 0.337 0.663

Node number 10: 579 observations
  predicted class=No   expected loss=0.4507772  P(node) =0.0723931
    class counts:   318   261
   probabilities: 0.549 0.451

Node number 11: 192 observations
  predicted class=Yes  expected loss=0.2447917  P(node) =0.024006
    class counts:    47   145
```

```
Node number 11: 192 observations
  predicted class=Yes  expected loss=0.2447917  P(node) =0.024006
    class counts:    47   145
   probabilities: 0.245 0.755

Node number 12: 1669 observations,    complexity param=0.01824922
  predicted class=No   expected loss=0.3648892  P(node) =0.2086772
    class counts:  1060   609
   probabilities: 0.635 0.365
  left son=24 (1569 obs) right son=25 (100 obs)
  Primary splits:
      diabetes    splits as  LR,       improve=44.06540, (0 missing)
      bp          splits as  RLL,      improve=38.34859, (0 missing)
      obesity     splits as  LR,       improve=26.38097, (0 missing)
      cholesteral splits as  RL,       improve=22.65515, (0 missing)
      age         < 71.5 to the left,  improve=16.19905, (0 missing)

Node number 13: 420 observations
  predicted class=Yes  expected loss=0.3190476  P(node) =0.05251313
    class counts:   134   286
   probabilities: 0.319 0.681

Node number 24: 1569 observations
  predicted class=No   expected loss=0.3358827  P(node) =0.196174
    class counts:  1042   527
   probabilities: 0.664 0.336

Node number 25: 100 observations
  predicted class=Yes  expected loss=0.18  P(node) =0.01250313
    class counts:    18    82
   probabilities: 0.180 0.820
```



Heart Attack

```
Predictcart = predict(cart, data = df, type = "class")
table(df$heartattack_s, Predictcart)
mean(Predictcart==df$heartattack_s)
```

```
> table(df$heartattack_s, Predictcart)
     Predictcart
        No  Yes
  No  3522  969
  Yes 1479 2028
> mean(Predictcart==df$heartattack_s)
[1] 0.6939235
> 
```

From the cart models summery we can see that the level of importance for each attribute has changed drastically. According to this cart model it shows the different nodes, their complexity parameters and variable breakdown in each node. Here we can observe the best complexity parameter is 0.01 and there are four different splits. The main difference being the change in the variable importance of the model, as age is most important variable in splitting, followed by smoker active and diabetes.

Here the output of the cart model is much less complex and easy to interpret than the c.50 model. It has much lesser number of node and branches, and also does not require pruning due to good complexity parameter value. Whereas c.50 model might be overfitting due to high number of branches and nodes and hence might require pruning to be done. Another difference as discussed was the change in priority or importance of attributes for splitting the nodes in both the models. Some variables in c.50 model do not play any significant role in cart model (gender, cholesterol). Another obvious difference being the accuracy of CART model (69.3%) is lower than that of c5.0 with worse false negative predictions.

3. (20 points) Using partition ratio 80:20 to rerun you C5.0 model. Do you have the similar accuracy for the test data? If not, how will you improve your model?

```
#Now building c5.0 model with 80:20 partitioning
#train test split with seed
set.seed(100)
df_split <- createDataPartition(df$heartattack_s, p = 0.80, list =FALSE)
df_train <- df[df_split,]
df_train_labels <- df$heartattack_s[df_split]
df_test <- df[-df_split,]
```

```
#build model
c50tree3 <- C5.0(df_train[,-7], as.factor(df_train$heartattack_s))
#summary
summary(c50tree3)
#plotting
plot(c50tree3, type="simple")
#predicting and checking the accuracy of our model
df_predictions <- predict(c50tree3, df_test[,-7])
```

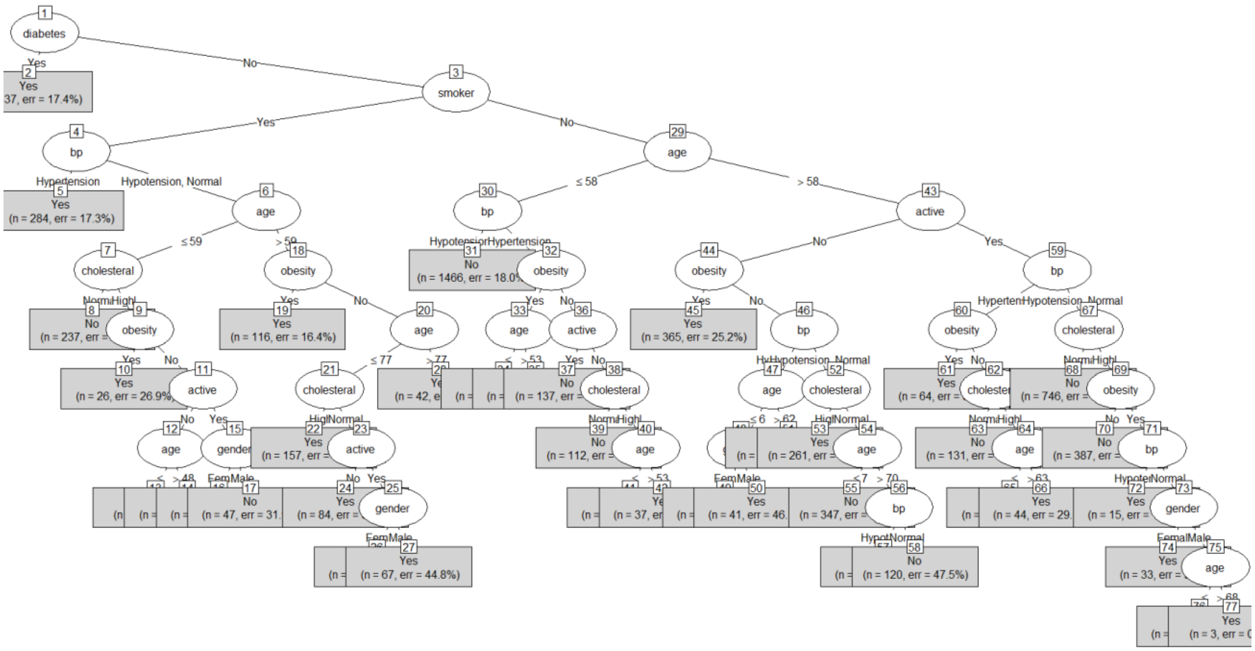
```
> summary(c50tree3)

Call:
C5.0.default(x = df_train[, -7], y = as.factor(df_train$heartattack_s))


C5.0 [Release 2.07 GPL Edition]          Sun Apr 10 10:45:35 2022
-------------------------------

Class specified by attribute `outcome'

Read 6399 cases (9 attributes) from undefined.data

Decision tree:

diabetes = Yes: Yes (437/76)
diabetes = No:
:...smoker = Yes:
    :...bp = Hypertension: Yes (284/49)
    :   bp in {Hypotension,Normal}:
    :   :...age <= 59:
    :       :...cholesteral = Normal: No (237/81)
    :       :   cholesteral = Highl:
    :       :   :...obesity = Yes: Yes (26/7)
    :       :       obesity = No:
    :       :       :...active = No:
    :       :           :...age <= 48: No (3)
    :       :           :   age > 48: Yes (49/14)
    :       :           active = Yes:
    :       :           :...gender = Female: Yes (34/15)
    :       :               gender = Male: No (47/15)
    :       age > 59:
    :       :...obesity = Yes: Yes (116/19)
    :           obesity = No:
    :           :...age > 77: Yes (42/4)
    :               age <= 77:
    :               :...cholesteral = Highl: Yes (157/43)
    :                   cholesteral = Normal:
    :                   :...active = No: Yes (84/31)
    :                       active = Yes:
    :                       :...gender = Female: No (59/23)
    :                           gender = Male: Yes (67/30)
    smoker = No:
    :...age <= 58:
        :...bp in {Hypotension,Normal}: No (1466/264)
        :   bp = Hypertension:
        :   :...obesity = Yes:
        :       :...age <= 53: No (44/17)
        :       :   age > 53: Yes (57/20)
        :       obesity = No:
        :       :...active = Yes: No (137/23)
        :           active = No:
        :           :...cholesteral = Normal: No (112/34)
        :               cholesteral = Highl:
        :               :...age <= 53: No (42/15)
        :                   age > 53: Yes (37/12)
        age > 58:
        :...active = No:
            :...obesity = Yes: Yes (365/92)
```

```
    :...age <= 58:
        :...bp in {Hypotension,Normal}: No (1466/264)
        :   bp = Hypertension:
        :   :...obesity = Yes:
        :       :...age <= 53: No (44/17)
        :       :   age > 53: Yes (57/20)
        :       obesity = No:
        :       :...active = Yes: No (137/23)
        :           active = No:
        :           :...cholesteral = Normal: No (112/34)
        :               cholesteral = Highl:
        :               :...age <= 53: No (42/15)
        :                   age > 53: Yes (37/12)
        age > 58:
        :...active = No:
            :...obesity = Yes: Yes (365/92)
            :   obesity = No:
            :   :...bp = Hypertension:
            :       :...age > 62: Yes (216/65)
            :       :   age <= 62:
            :       :   :...gender = Female: No (22/8)
            :       :       gender = Male: Yes (41/19)
            :       bp in {Hypotension,Normal}:
            :       :...cholesteral = Highl: Yes (261/115)
            :           cholesteral = Normal:
            :           :...age <= 70: No (347/116)
            :               age > 70:
            :               :...bp = Hypotension: Yes (26/9)
            :                   bp = Normal: No (120/57)
            active = Yes:
            :...bp = Hypertension:
                :...obesity = Yes: Yes (64/20)
                :   obesity = No:
                :   :...cholesteral = Normal: No (131/49)
                :       cholesteral = Highl:
                :       :...age <= 63: No (26/11)
                :           age > 63: Yes (44/13)
                bp in {Hypotension,Normal}:
                :...cholesteral = Normal: No (746/164)
                    cholesteral = Highl:
                    :...obesity = No: No (387/136)
                        obesity = Yes:
                        :...bp = Hypotension: Yes (15/2)
                            bp = Normal:
                            :...gender = Female: Yes (33/12)
                                gender = Male:
                                :...age <= 68: No (15/2)
                                    age > 68: Yes (3)


Evaluation on training data (6399 cases):

	    Decision Tree   
	  ----------------  
	  Size      Errors  

	    39 1682(26.3%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	  2926   667    (a): class No
	  1015  1791    (b): class Yes
```

```
Evaluation on training data (6399 cases):

            Decision Tree
          ----------------
          Size      Errors

            39 1682(26.3%)   <<


           (a)   (b)    <-classified as
          ----  ----
          2926   667    (a): class No
          1015  1791    (b): class Yes


        Attribute usage:

        100.00% diabetes
         93.17% smoker
         88.73% age
         87.47% bp
         55.21% active
         50.46% obesity
         48.57% cholesteral
          5.02% gender


Time: 0.0 secs
```

```
#plotting
plot(c50tree3, type="simple")
#Now let us compute the accuracy of our model
train_predictions <- predict(c50tree3, df_train)
mean(train_predictions==df_train$heartattack_s)
test_predictions <- predict(c50tree3, df_test)
mean(test_predictions==df_test$heartattack_s)
```

```
> train_predictions <- predict(c50tree3, df_train)
> mean(train_predictions==df_train$heartattack_s)
[1] 0.7371464
> test_predictions <- predict(c50tree3, df_test)
> mean(test_predictions==df_test$heartattack_s)
[1] 0.7148218
>
```



With the change in splitting the dataset into train and test we can observe that the train accuracy went slightly up from 73.6% to 73.7% and test accuracy was 71.4%. Though the sequence of importance of the attributes remained the same in the non partitioned and portioned models, the percentage of attribute usage changed significantly (the percentage of attribute usage for age was 93%, it changed to 88% and the percentage of bp was 77%, it changed to 87%).

Since this model is build to predict the diagnosis of heart attacks, the false negative cases are very important. As we would not want to give a prediction to a patient saying they do not have chances of heart attack when they actually do. From these portioned and non-partitioned model we can observe that the false negative in portioned model dropped significantly though the accuracy remained the same (from 1247 to 1015 in partitioned model). This is a huge plus as false negative should be as low as possible.

Now we will do some exploration to see if we can further improve this model:

```
## creating c50 with rules= True
c50tree3rules <- C5.0(heartattack_s ~ ., data=df_train, rules = TRUE)
summary(c50tree3rules)

#Now let us compute the accuracy of our model
train_predictions <- predict(c50tree3rules, df_train)
mean(train_predictions==df_train$heartattack_s)
test_predictions <- predict(c50tree3rules, df_test)
mean(test_predictions==df_test$heartattack_s)
```

```
> summary(c50tree3rules)

Call:
C5.0.formula(formula = heartattack_s ~ ., data = df_train, rules = TRUE)


C5.0 [Release 2.07 GPL Edition]         Sun Apr 10 10:56:35 2022
-------------------------------

Class specified by attribute `outcome'

Read 6399 cases (9 attributes) from undefined.data

Rules:

Rule 1: (636/117, lift 1.5)
        age <= 77
        gender = Female
        diabetes = No
        active = Yes
        obesity = No
        bp in {Hypotension, Normal}
        cholesteral = Normal
        ->  class No  [0.815]

Rule 2: (285/54, lift 1.4)
        age <= 48
        ->  class No  [0.808]

Rule 3: (868/171, lift 1.4)
        gender = Male
        diabetes = No
        smoker = No
        active = Yes
        bp = Normal
        ->  class No  [0.802]

Rule 4: (5962/2445, lift 1.1)
        diabetes = No
        ->  class No  [0.590]

Rule 5: (19, lift 2.2)
        age > 68
        gender = Male
        active = Yes
        obesity = Yes
        cholesteral = Highl
        ->  class Yes  [0.952]
```

```
Rule 6: (141/13, lift 2.1)
        smoker = Yes
        obesity = Yes
        cholesteral = Highl
        ->  class Yes  [0.902]

Rule 7: (265/40, lift 1.9)
        age > 48
        smoker = Yes
        active = No
        cholesteral = Highl
        ->  class Yes  [0.846]

Rule 8: (339/52, lift 1.9)
        smoker = Yes
        bp = Hypertension
        ->  class Yes  [0.845]

Rule 9: (255/41, lift 1.9)
        age > 63
        bp = Hypertension
        cholesteral = Highl
        ->  class Yes  [0.837]

Rule 10: (407/69, lift 1.9)
        age > 53
        obesity = Yes
        bp = Hypertension
        ->  class Yes  [0.829]

Rule 11: (437/76, lift 1.9)
        diabetes = Yes
        ->  class Yes  [0.825]

Rule 12: (299/52, lift 1.9)
        age > 53
        active = No
        bp = Hypertension
        cholesteral = Highl
        ->  class Yes  [0.824]

Rule 13: (467/86, lift 1.9)
        age > 62
        active = No
        bp = Hypertension
        ->  class Yes  [0.814]

Rule 14: (364/68, lift 1.9)
        age > 58
        obesity = Yes
        cholesteral = Highl
        ->  class Yes  [0.811]
```

```
Rule 15: (551/107, lift 1.8)
        age > 58
        active = No
        obesity = Yes
        ->  class Yes  [0.805]

Rule 16: (264/51, lift 1.8)
        gender = Female
        smoker = Yes
        cholesteral = High1
        ->  class Yes  [0.805]

Rule 17: (51/10, lift 1.8)
        age > 70
        active = No
        bp = Hypotension
        cholesteral = Normal
        ->  class Yes  [0.792]

Rule 18: (751/187, lift 1.7)
        age > 59
        smoker = Yes
        ->  class Yes  [0.750]

Rule 19: (723/189, lift 1.7)
        age > 58
        active = No
        cholesteral = High1
        ->  class Yes  [0.738]

Default class: No


Evaluation on training data (6399 cases):

                Rules
          ----------------
            No      Errors

            19 1683(26.3%)   <<


           (a)   (b)    <-classified as
          ----  ----
          2943   650    (a): class No
          1033  1773    (b): class Yes


        Attribute usage:

        100.00% diabetes
         46.69% age
         45.91% active
         39.90% bp
         29.43% cholesteral
         29.32% smoker
         27.88% gender
         23.43% obesity


Time: 0.0 secs
```

With the creation of rules (19 of them were created). We can see that the accuracy remained the same but the false negative cases slightly increased, hence this does not provide any improvement in our model.

Now lets try balancing the dataset with the help of up sampling as we had observed that the people with heart attacks were lesser.

```
### Dealing with the slight imbalance in the data
table(df_train$heartattack_s)
```

```
> table(df_train$heartattack_s)

  No  Yes
3593 2806
```

After upsampling:

```
set.seed(100)
trainup<-upSample(x=df_train[,-ncol(df_train)],y=df_train$heartattack_s)
table(trainup$heartattack_s)
```

```
> set.seed(100)
> trainup<-upSample(x=df_train[,-ncol(df_train)],y=df_train$heartattack_s)
> table(trainup$heartattack_s)

  No  Yes
3593 3593
```

```
# building model for the balanced model
str(trainup)
vars <- c("age", "gender","diabetes","smoker","active","obesity","bp")
c50tree4 <- C5.0(x = trainup[, vars], y = as.factor(trainup$heartattack_s))
#summary
summary(c50tree4)
#plotting
plot(c50tree4, type="simple")
#Now let us compute the accuracy of our model
train_predictions <- predict(c50tree4, trainup)
mean(train_predictions==trainup$heartattack_s)
test_predictions <- predict(c50tree4, df_test)
mean(test_predictions==df_test$heartattack_s)
```

```
> #summary
> summary(c50tree4)

Call:
C5.0.default(x = trainup[, vars], y = as.factor(trainup$heartattack_s))


C5.0 [Release 2.07 GPL Edition]         Sun Apr 10 11:03:32 2022
-------------------------------

Class specified by attribute `outcome'

Read 7186 cases (8 attributes) from undefined.data

Decision tree:

diabetes = Yes: Yes (528/76)
diabetes = No:
:...smoker = No:
    :...age <= 58:
    :   :...bp in {Hypotension,Normal}: No (1534/332)
    :   :   bp = Hypertension:
    :   :   :...obesity = No: No (348/117)
    :   :       obesity = Yes:
    :   :       :...age <= 52: No (43/19)
    :   :           age > 52: Yes (77/23)
    :   age > 58:
    :   :...active = No:
    :       :...obesity = Yes: Yes (449/92)
    :       :   obesity = No:
    :       :   :...bp = Hypertension: Yes (319/98)
    :       :       bp in {Hypotension,Normal}:
    :       :       :...age <= 67: No (484/217)
    :       :           age > 67: Yes (372/151)
    :       active = Yes:
    :       :...bp = Normal: No (1034/323)
    :           bp in {Hypertension,Hypotension}:
    :           :...obesity = Yes: Yes (124/35)
    :               obesity = No:
    :               :...age <= 71: No (307/118)
    :                   age > 71: Yes (132/55)
    smoker = Yes:
    :...bp = Hypertension: Yes (357/49)
```

```
    :                   age > 71: Yes (132/55)
    smoker = Yes:
    :...bp = Hypertension: Yes (357/49)
        bp in {Hypotension,Normal}:
        :...age > 66: Yes (354/69)
            age <= 66:
            :...obesity = Yes: Yes (138/39)
                obesity = No:
                :...age <= 52:
                    :...bp = Hypotension: Yes (12/4)
                    :   bp = Normal: No (119/38)
                    age > 52:
                    :...gender = Female: Yes (220/80)
                        gender = Male:
                        :...active = Yes: No (140/67)
                            active = No:
                            :...age <= 54: Yes (25/6)
                                age > 54:
                                :...bp = Hypotension: Yes (10/3)
                                    bp = Normal: No (60/25)


Evaluation on training data (7186 cases):

            Decision Tree
          ----------------
          Size      Errors

            23 2036(28.3%)   <<


           (a)   (b)    <-classified as
          ----  ----
          2813   780    (a): class No
          1256  2337    (b): class Yes


        Attribute usage:

        100.00% diabetes
         92.65% smoker
         87.68% age
         86.40% bp
         48.09% active
         47.02% obesity
          6.33% gender


Time: 0.0 secs
> #plotting
> plot(c50tree4, type="simple")
> #Now let us compute the accuracy of our model
> train_predictions <- predict(c50tree4, trainup)
> mean(train_predictions==trainup$heartattack_s)
[1] 0.7166713
> test_predictions <- predict(c50tree4, df_test)
> mean(test_predictions==df_test$heartattack_s)
[1] 0.6985616
> 
```

Again we can see that the accuracy decreased than previous observations, we cannot compare the false negative cases directly here as the number of patients with heart attacks were up sampled in this model.

I believe having more data would be better as after removing duplicates we are left only with 3506 rows.

Appendix (The complete version of your solution scripts in R)

```
# Installing and loading all the libraries
#install.packages("rattle")
#install.packages("polycor")
#install.packages("dplyr")


library(dplyr)
library(polycor)
library(readxl)
library("readxl")
library('C50')
library(rpart)
library(caret)
library(rattle)
library(psych)#categorical correlation


df <- read_excel("R://downloads//Patient_Data.xlsx")
#now we inspect data to see each variable
str(df)
#since variable type is char, we change it to factor for all the applicable vairables
df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)],as.factor)
#lets check if they are converted to factors
str(df)
#Lets summarize our data
```

```
summary(df)

# to check correlation between each binary categorical data
cc=table(df$diabetes,df$heartattack_s)
tetrachoric(cc)
cc=table(df$gender,df$heartattack_s)
tetrachoric(cc)
cc=table(df$smoker,df$heartattack_s)
tetrachoric(cc)
cc=table(df$active,df$heartattack_s)
tetrachoric(cc)
cc=table(df$obesity,df$heartattack_s)
tetrachoric(cc)
cc=table(df$cholesteral,df$heartattack_s)
tetrachoric(cc)

#EDA
plot(df$heartattack_s,df$age, xlab="heart attack", ylab="patient age")
plot(df$heartattack_s,df$active, xlab="heart attack", ylab="active")
plot(df$heartattack_s,df$smoker, xlab="heart attack", ylab="smoker")
plot(df$heartattack_s,df$gender, xlab="heart attack", ylab="gender")

#outliers in age variable
boxplot(df$age, ylab="age")
```

```
#skewness of age variable
hist(df$age,ylab="count",xlab="age")

#checking if the data is balanced
barplot(table(df$heartattack_s),ylab="count",xlab="heart attack")

#to check if there are any duplicate values
#duplicated(df)
sum(duplicated(df))
nrow(df)
df2 =unique(df)
nrow(df2)

#check for missing values
sum(is.na(df))

#building the c5.0 model
c50tree1 <- C5.0(df[,-7], as.factor(df$heartattack_s))
#summary
summary(c50tree1)
#plotting
```

```
plot(c50tree1,type= "simple", cex=.7)

#Now let us compute the accuracy of our model

predictions <- predict(c50tree1, df)

mean(predictions==df$heartattack_s)



#building the cart model

cart <- rpart(heartattack_s~., data = df, method = "class")

summary(cart)

#plotting

fancyRpartPlot(cart, cex = 1.0, caption = "Heart Attack")

Predictcart = predict(cart, data = df, type = "class")

table(df$heartattack_s, Predictcart)

mean(Predictcart==df$heartattack_s)


#Now building c5.0 model with 80:20 partitioning

#train test split with seed

set.seed(100)

df_split <- createDataPartition(df$heartattack_s, p = 0.80, list =FALSE)

df_train <- df[df_split,]

df_train_labels <- df$heartattack_s[df_split]

df_test <- df[-df_split,]

#build model

c50tree3 <- C5.0(df_train[,-7], as.factor(df_train$heartattack_s))

#summary

summary(c50tree3)

#plotting
```

```
plot(c50tree3, type="simple")

#Now let us compute the accuracy of our model

train_predictions <- predict(c50tree3, df_train)

mean(train_predictions==df_train$heartattack_s)

test_predictions <- predict(c50tree3, df_test)

mean(test_predictions==df_test$heartattack_s)



## creating c50 with rules= True

c50tree3rules <- C5.0(heartattack_s ~ ., data=df_train, rules = TRUE)

summary(c50tree3rules)

#Now let us compute the accuracy of our model

train_predictions <- predict(c50tree3rules, df_train)

mean(train_predictions==df_train$heartattack_s)

test_predictions <- predict(c50tree3rules, df_test)

mean(test_predictions==df_test$heartattack_s)



### Dealing with the slight imbalance in the data

table(df_train$heartattack_s)

set.seed(100)

trainup<-upSample(x=df_train[,-ncol(df_train)],y=df_train$heartattack_s)

table(trainup$heartattack_s)

# building model for the balanced model

str(trainup)

vars <- c("age", "gender","diabetes","smoker","active","obesity","bp")
```

```
c50tree4 <- C5.0(x = trainup[, vars], y = as.factor(trainup$heartattack_s))

#summary

summary(c50tree4)

#plotting

plot(c50tree4, type="simple")

#Now let us compute the accuracy of our model

train_predictions <- predict(c50tree4, trainup)

mean(train_predictions==trainup$heartattack_s)

test_predictions <- predict(c50tree4, df_test)

mean(test_predictions==df_test$heartattack_s)
```