**Assignment (Decision Trees)**

**Homework due on October 28, 2021**

**Total Points: 50 points**

**Group Work (3 people maximum)**

Names : Alejandra De Osma, Shannon Polk, Jack Gabriel

By writing my name, I acknowledge that I am aware and understand the Rollins Honor Code. Therefore, all

the answers of this homework are the product of my/our own intellectual effort.

| | Age | Job | House | Credit | Loan_Approved |
|---|---|---|---|---|---|
| 1 | Young | FALSE | No | Good | No |

| | Age | Job | House | Credit | Loan_Approved |
|---|---|---|---|---|---|
| 1 | Young | FALSE | No | Fair | No |
| 2 | Young | FALSE | No | Good | No |
| 3 | Young | TRUE | No | Good | Yes |
| 4 | Young | TRUE | Yes | Fair | Yes |
| 5 | Young | FALSE | No | Fair | No |
| 6 | Middle | FALSE | No | Fair | No |
| 7 | Middle | FALSE | No | Good | No |
| 8 | Middle | TRUE | Yes | Good | Yes |
| 9 | Middle | FALSE | Yes | Excellent | Yes |
| 10 | Middle | FALSE | Yes | Excellent | Yes |
| 11 | Old | FALSE | Yes | Excellent | Yes |
| 12 | Old | FALSE | Yes | Good | Yes |
| 13 | Old | TRUE | No | Good | Yes |
| 14 | Old | TRUE | No | Excellent | Yes |
| 15 | Old | FALSE | No | Fair | No |

| House | Loan_Approved |
|---|---|
| Yes | Yes |
| Yes | Yes |
| Yes | Yes |
| Yes | Yes |
| Yes | Yes |
| Yes | Yes |
| No | No |
| No | No |
| No | No |
| No | No |
| No | No |
| No | No |
| No | Yes |
| No | Yes |
| No | Yes |

| Frequency | House | Misses | Total Misses |
|---|---|---|---|
| 6 | Yes —> | 0/6 | |
| 9 | No —> | 3/9 | |
| 15 | | | 3/15 |

House = Yes is a pure node —> 0/6

| Job | Loan_Approved |
|---|---|
| TRUE | Yes |
| TRUE | Yes |
| TRUE | Yes |
| TRUE | Yes |
| TRUE | Yes |
| FALSE | No |
| FALSE | No |
| FALSE | No |
| FALSE | No |
| FALSE | No |
| FALSE | No |
| FALSE | Yes |
| FALSE | Yes |
| FALSE | Yes |
| FALSE | Yes |

| Frequency | Job | Misses | Total Misses |
|---|---|---|---|
| 5 | TRUE —> | 0/5 | |
| 10 | FALSE —> | 4/10 | |
| 15 | | | 4/15 |

Job = True is a pure node —> 0/5

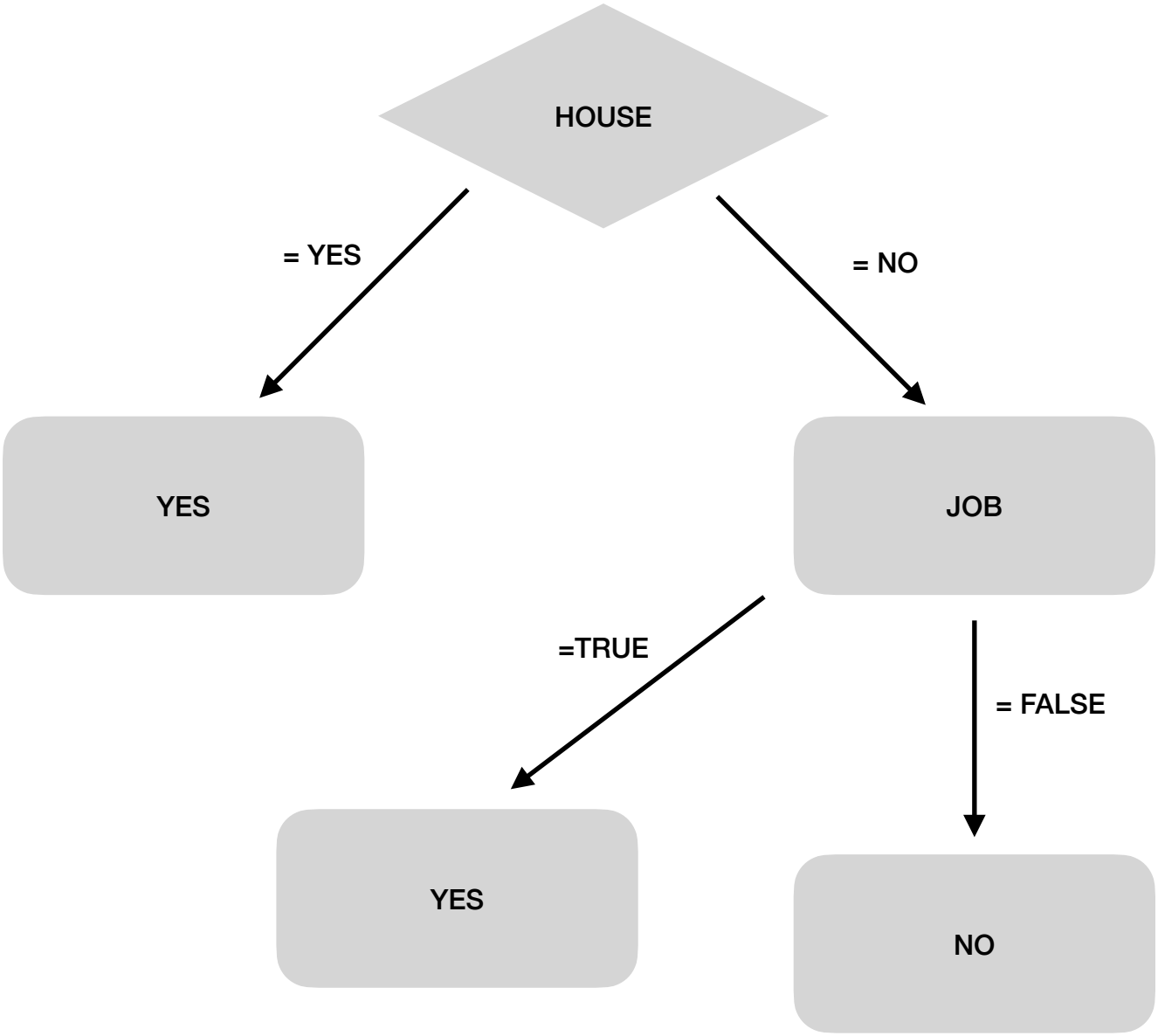
| Credit | Loan_Approved |
|---|---|
| Fair | No |
| Fair | No |
| Fair | No |
| Fair | No |
| Fair | Yes |
| Good | No |
| Good | No |
| Good | Yes |
| Good | Yes |
| Good | Yes |
| Good | Yes |
| Excellent | Yes |
| Excellent | Yes |
| Excellent | Yes |
| Excellent | Yes |

| Frequency | Job | Misses | Total Misses |
|---|---|---|---|
| 5 | Fair—> | 1/5 | |
| 5 | Good —> | 2/6 | |
| 5 | Excellent —> | 0/4 | |
| 15 | | | 3/15 |

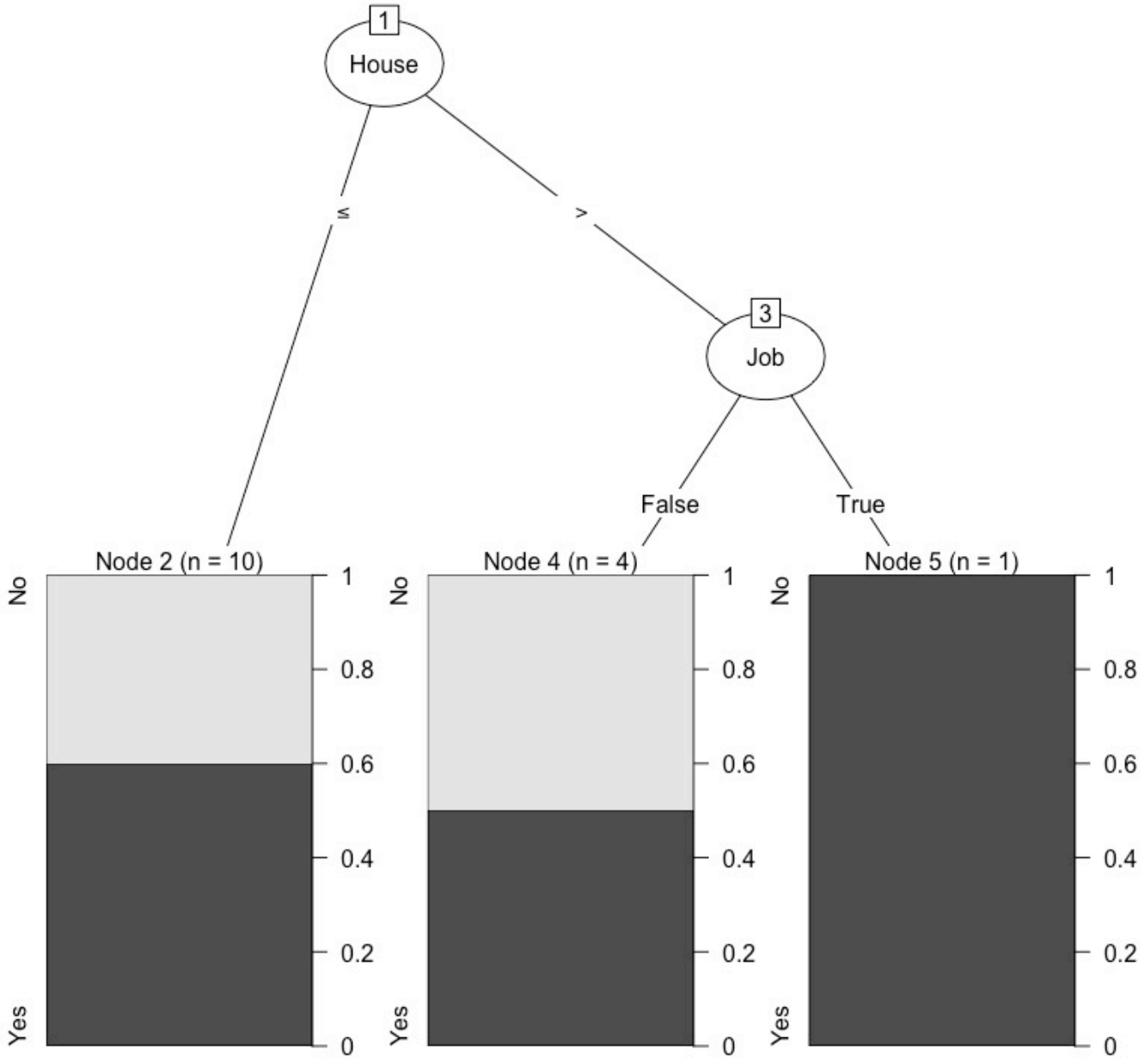
**Job = True is a pure node —> 0/4**

| | Outcomes | |
|---|---|---|
| **House** | Yes —> | Pure Node |
| | No —> | Job |
| | | |
| **Job** | TRUE —> | Pure Node |
| | FALSE —> | Loan Rejected |

1) Build the decision tree by hand following the method indicated in the lecture or in the class



2) Build the decision tree in R using following the procedure indicated in the lecture. Use R package C50.



3) Write a function in R that Calculate the Information Gain of the Parent Table with respect to its children. Calculate the IG of the grandchildren or great grandchildren. Explain if the result obtained by the IG of the Children of the root agree or disagree with the results obtained in part 1 of this homework. You may have to create several data frame or use the dplyr functions to extract the data from the original table that will be a data frame that you can create by any

First writing a function to calculate the entropy:

```
## This function calculates the entropy of a parent of child
## Parameters: Number of misses and number of Successes

entropy <-function(m,s){

  ## Testing value to avoid division by 0
  if(m <= 0 || s <= 0 ){

    return(as.numeric(0))

  }else{

  ## Calculates total
  total = m+s

  ##Probability of both misses and successes
  p = m/total
  q = s/total

  ## Calculates entropy:
  e <- (p*log2(p) + q*log2(q))*(-1)

  ## Returns calculated entropy
  return(e)

  }

}
```

Then writing a function to compute the information gain, which uses the entropy function above.

```
info_gain <- function(df1,parent_1,child_1){

 dataf<- data.frame(select(df1,child_1,parent_1))
 print(dataf)
 cat("\n\n")

 #Parent entropy
 pt <- table(select(dataf,parent_1))
 pt <- as.data.frame(pt)
 ptotal <- sum(pt$Freq)

 x<- pt$Freq[1]
 y<- pt$Freq[2]

 parent_1_entropy = entropy(pt$Freq[1],pt$Freq[2])

 cat("Parent Entropy:",parent_1_entropy,"\n\n")

 table_1 <- table(dataf)

 #Check length of table
 if (length(table_1)>4){

  #Child1:
  c1= table_1[1,1]
  c1_2= table_1[1,2]
  pc1 <- (c1+c1_2)/ptotal
  ec1 <- entropy(c1,c1_2)


  #Child2:
  c2= table_1[2,1]
  c2_2= table_1[2,2]
  pc2 <- (c2+c2_2)/ptotal
  ec2 <- entropy(c2,c2_2)

  #Child3:
  c3= table_1[3,1]
  c3_2= table_1[3,2]
  pc3 <- (c3+c3_2)/ptotal
  ec3 <- entropy(c3,c3_2)


  cat("entropy 1 :",ec1,"\n")
  cat("entropy 2 :",ec2,"\n")
  cat("entropy 3 :",ec3,"\n\n")


  ig = parent_1_entropy - ((pc1*ec1)+(pc2*ec2)+(pc3*ec3))

  cat("Information Gain:",ig,"\n")

 }else{

  #Child1:
  c1= table_1[1,1]
  c1_2= table_1[1,2]
  pc1 <- (c1+c1_2)/ptotal
  ec1 <- entropy(c1,c1_2)


  #Child2:
  c2= table_1[2,1]
  c2_2= table_1[2,2]
  pc2 <- (c2+c2_2)/ptotal
  ec2 <- entropy(c2,c2_2)

  cat("\n\n")
  print(table_1)
  cat("\n\n")
  cat("entropy 1 :",ec1,"\n")
  cat("entropy 2 :",ec2,"\n")


  ig = parent_1_entropy - ((pc1*ec1)+(pc2*ec2))

  cat("Information Gain:",ig,"\n")

 }
}
```

Test Output:

```
> ## Reading the given data with the read.table() function
> df<- read.table("Desktop/tree_data2",header = TRUE)
>
> ## getting colnames
> colnames(df)
[1] "Age"           "Job"           "House"         "Credit"        "Loan_Approved"
>
> ## Calling function info_gain with Loan_Approved as the parent and Credit as the Child
> info_gain(df,"Loan_Approved","Credit")
      Credit Loan_Approved
1       Fair            No
2       Good            No
3       Good           Yes
4       Fair           Yes
5       Fair            No
6       Fair            No
7       Good            No
8       Good           Yes
9  Excellent           Yes
10 Excellent           Yes
11 Excellent           Yes
12      Good           Yes
13      Good           Yes
14 Excellent           Yes
15      Fair            No


Parent Entropy: 0.9709506

entropy 1 : 0
entropy 2 : 0.7219281
entropy 3 : 0.9182958

Information Gain: 0.3629896
>
```

Information gain summary:

| | Information Gain |
|---|---|
| **IG(Parent, House)** | 0.4199731 |
| **IG(Parent, Job)** | 0.3236502 |
| **IG(Parent, Credit)** | 0.3629896 |
| **IG(Parent, Age)** | 0.0830075 |

House has the highest value, therefore becomes the root since it provides the maximum Information. Job is then considered as a child node.  Age provides the least information therefore it is not considered when building a tree. After completing calculations, my initial predictions coincide with the results gathered by the functions.

CODE IS ATTACHED BELOW.

```
## Project 1
## Names: Alejandra De Osma, Shannon Polk, Jack Gabriel


info_gain <- function(df1,parent_1,child_1){

 dataf<- data.frame(select(df1,child_1,parent_1))
 print(dataf)
 cat("\n\n")

 #Parent entropy
 pt <- table(select(dataf,parent_1))
 pt <- as.data.frame(pt)
 ptotal <- sum(pt$Freq)

 x<- pt$Freq[1]
 y<- pt$Freq[2]

 parent_1_entropy = entropy(pt$Freq[1],pt$Freq[2])

 cat("Parent Entropy:",parent_1_entropy,"\n\n")

 table_1 <- table(dataf)

 #Check length of table
 if (length(table_1)>4){

  #Child1:
  c1= table_1[1,1]
  c1_2= table_1[1,2]
  pc1 <- (c1+c1_2)/ptotal
  ec1 <- entropy(c1,c1_2)


  #Child2:
  c2= table_1[2,1]
  c2_2= table_1[2,2]
  pc2 <- (c2+c2_2)/ptotal
  ec2 <- entropy(c2,c2_2)

  #Child3:
  c3= table_1[3,1]
  c3_2= table_1[3,2]
  pc3 <- (c3+c3_2)/ptotal
  ec3 <- entropy(c3,c3_2)


  cat("entropy 1 :",ec1,"\n")
  cat("entropy 2 :",ec2,"\n")
```

```
  cat("entropy 3 :",ec3,"\n\n")


  ig = parent_1_entropy - ((pc1*ec1)+(pc2*ec2)+(pc3*ec3))

  cat("Information Gain:",ig,"\n")


 }else{

   #Child1:
   c1= table_1[1,1]
   c1_2= table_1[1,2]
   pc1 <- (c1+c1_2)/ptotal
   ec1 <- entropy(c1,c1_2)


   #Child2:
   c2= table_1[2,1]
   c2_2= table_1[2,2]
   pc2 <- (c2+c2_2)/ptotal
   ec2 <- entropy(c2,c2_2)

   cat("\n\n")
   print(table_1)
   cat("\n\n")
   cat("entropy 1 :",ec1,"\n")
   cat("entropy 2 :",ec2,"\n")


   ig = parent_1_entropy - ((pc1*ec1)+(pc2*ec2))

   cat("Information Gain:",ig,"\n")

 }
}




## This function calculates the entropy of a parent of child
## Parameters: Number of misses and number of Successes

entropy <-function(m,s){

  ## Testing value to avoid division by 0
  if(m <= 0 || s <= 0 ){

    return(as.numeric(0))
```

```
  }else{

  ## Calculates total
  total = m+s

  ##Probability of both misses and successes
  p = m/total
  q = s/total

  ## Calculates entropy:
  e <- (p*log2(p) + q*log2(q))*(-1)

  ## Returns calculated entropy
  return(e)

  }

}
```