Homework 3

Matthew Nickols

2024-03-17

Problem 1

(a)

```
#Smoothing list of ages by bin mean with a bin depth of three
import numpy as np
#Copy over list
age = [13, 15, 16, 16, 19, 20, 20,
21, 22, 22, 25, 25, 25, 25, 30, 33,
33, 35, 35, 35, 35, 36, 40, 45, 46,
52, 70]

#Create list for bins
bins = []
#Loop through age counting by 3
for i in range(0, len(age), 3):
  #Take 3 ages and calculate mean
  mean = sum(age[i: i+2]) / 3
  #round mean to 2 decimal places to make it neater
  mean = round(mean, 2)
  #Create bin
  bin = [mean, mean, mean]
  #Place bin in list of bins
  bins.append(bin)

#Output bins
for bin in bins:
  print(bin)
```

```
## [9.33, 9.33, 9.33]
## [11.67, 11.67, 11.67]
## [13.67, 13.67, 13.67]
## [15.67, 15.67, 15.67]
## [16.67, 16.67, 16.67]
## [22.0, 22.0, 22.0]
## [23.33, 23.33, 23.33]
## [25.33, 25.33, 25.33]
## [32.67, 32.67, 32.67]
```

(b)

```
#Calculating IQR with Q1 and Q3 to find outliers
Q1 = np.percentile(age, 25)
Q3 = np.percentile(age, 75)
IQR = Q3 - Q1
#Calculate upper and lower limits using IQR
lowlim = Q1 - 1.5 * IQR
upperlim = Q3 + 1.5 * IQR

#Output
print('Q1 equals ', Q1, '\nQ3 equals ', Q3, '\nIQR equals ', IQR,
      '\nBounds for outliers are: (', lowlim, ' ', upperlim, ')')
```

```
## Q1 equals  20.5
## Q3 equals  35.0
## IQR equals  14.5
## Bounds for outliers are: ( -1.25   56.75 )
```

```
#Iterate through age checking each value against outlier limits
for i in age:
  if((i > upperlim) or (i < lowlim)):
    print(i, ' is an outlier in age data')
```

```
## 70  is an outlier in age data
```

(c)

```
#Use min-max normalization to transform 35 onto the range [0.0, 1.0]
#Min-max normalization
small = min(age)
big = max(age)
normalized = (35 - small) * (1 - 0) / (big - small + (0))


print('35 transformed onto the range [0.0, 1.0] using min-max normalization equals: ', normalized)
```

```
## 35 transformed onto the range [0.0, 1.0] using min-max normalization equals:  0.38596491228070173
```

(d)

```
#Use z-score normalization to transform 35 for age
avg = sum(age) / len(age)
stdev = np.std(age)

normalized = (35 - avg) / stdev

print('35 transformed using z-score normalization equals: ', normalized)
```

```
## 35 transformed using z-score normalization equals:  0.3966110348537352
```

(e)

```
#Use normalization by decimal scaling to transform the value 35 for age
normalized = 35 / 10**2
print('35 transformed using decimal scaling equals: ', normalized)
```

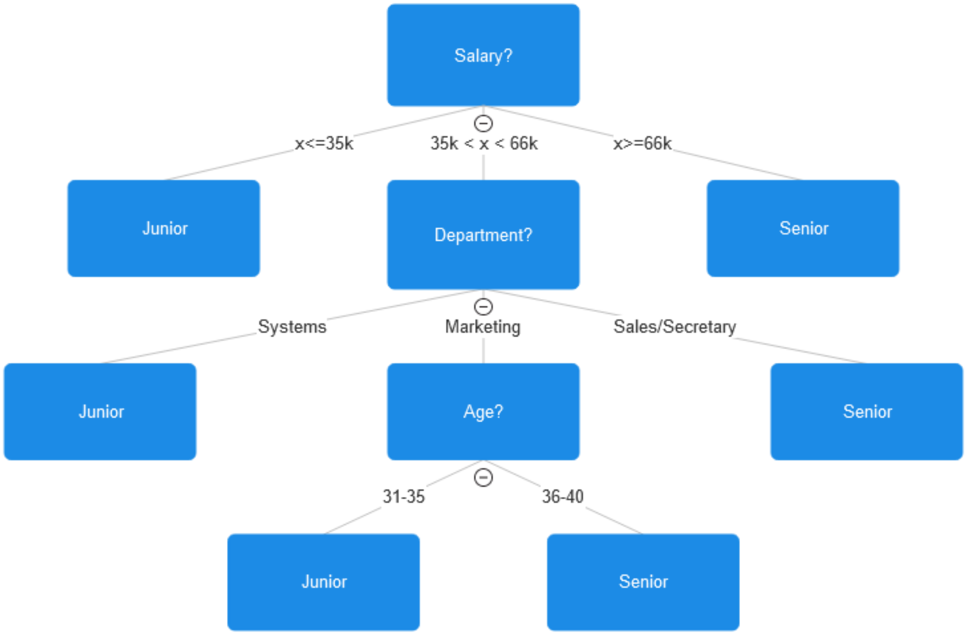
```
## 35 transformed using decimal scaling equals:  0.35
```

Problem 2

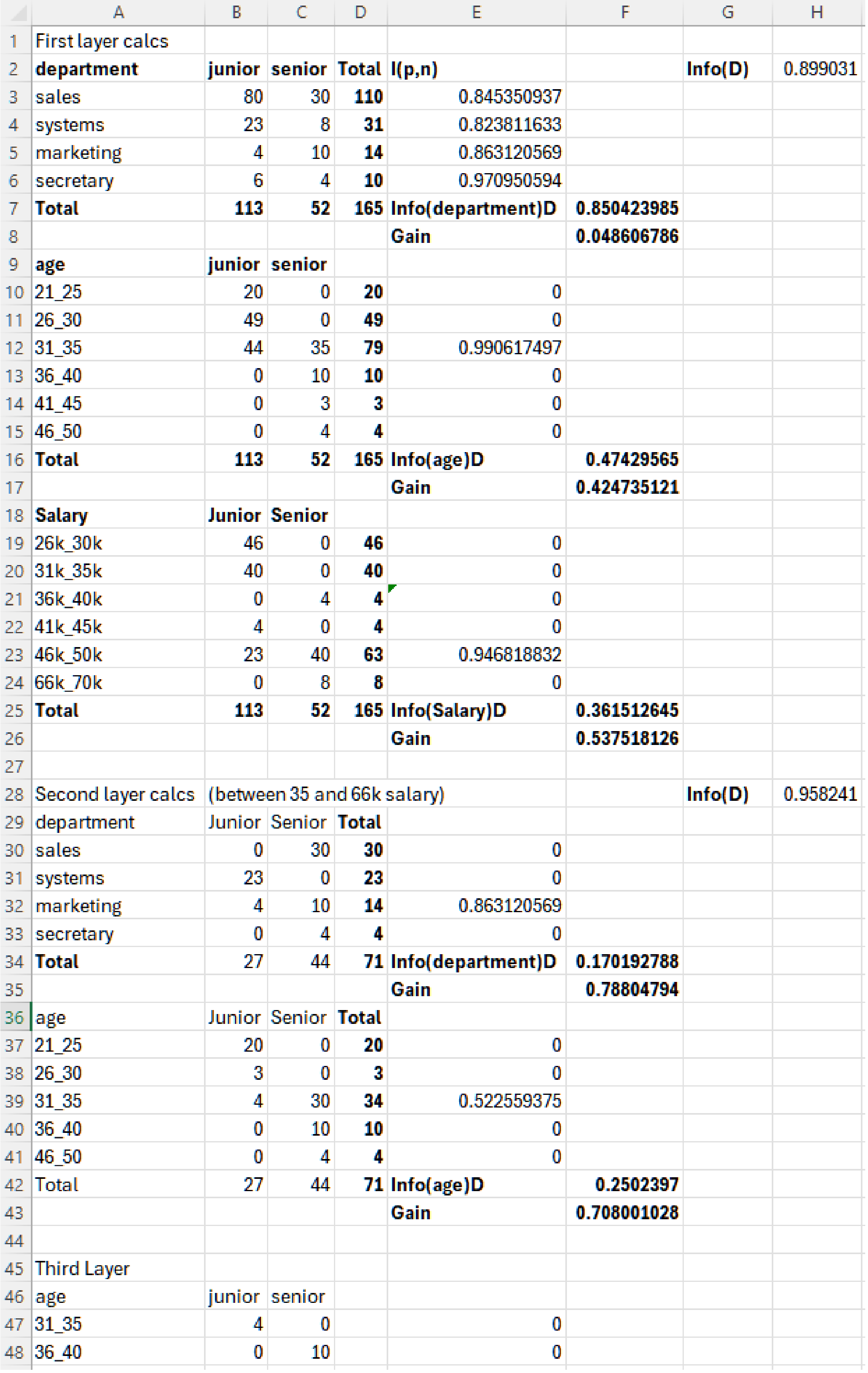
```
##Write function to normalize data to new min and max
##Define normalize function
def normalize (list, new_min, new_max):

  #printing old data that was given
  print('old data: ')
  for i in list:
    print(i)

  #Creating empty list for normalized values
  normalized_data = []

  #Defining old min and max values
  old_min = min(list)
  old_max = max(list)

  #iterating through the list and normalizing each value
  for i in list:
    normal = (i-old_min) * (new_max - new_min) / (old_max - old_min +
              new_min)

    #Rounding to 2 decimal places for neatness
    normal = round(normal, 2)

    #appending normalized value to list or normalized data
    normalized_data.append(normal)

  #printing new normalized data
  print('Normalized data: ')
  for i in normalized_data:
    print(i)

#Calling the normalize function on the age list from problem 1.
#Using 0 and 1 as example range, could be changed to any 2 values
normalize(age, 0, 1)
```

```
## old data:
## 13
## 15
## 16
## 16
## 19
## 20
## 20
## 21
## 22
## 22
## 25
## 25
## 25
## 25
## 30
## 33
## 33
## 35
## 35
## 35
## 35
## 36
## 40
## 45
## 46
## 52
## 70
## Normalized data:
## 0.0
## 0.04
## 0.05
## 0.05
## 0.11
## 0.12
## 0.12
## 0.14
## 0.16
## 0.16
## 0.21
## 0.21
## 0.21
## 0.21
## 0.3
## 0.35
## 0.35
## 0.39
## 0.39
## 0.39
## 0.39
## 0.4
## 0.47
## 0.56
## 0.58
## 0.68
## 1.0
```

Problem 3



#
#Decision Tree calculations were made in Excel Notebook. Excel file is InformationGainCalc.xlsx



	A	B	C	D	E	F	G	H
1	First layer calcs							
2	**department**	**junior**	**senior**	**Total**	**I(p,n)**		**Info(D)**	0.899031
3	sales	80	30	**110**	0.845350937			
4	systems	23	8	**31**	0.823811633			
5	marketing	4	10	**14**	0.863120569			
6	secretary	6	4	**10**	0.970950594			
7	**Total**	**113**	**52**	**165**	**Info(department)D**	**0.850423985**		
8					**Gain**	**0.048606786**		
9	**age**	**junior**	**senior**					
10	21_25	20	0	**20**	0			
11	26_30	49	0	**49**	0			
12	31_35	44	35	**79**	0.990617497			
13	36_40	0	10	**10**	0			
14	41_45	0	3	**3**	0			
15	46_50	0	4	**4**	0			
16	**Total**	**113**	**52**	**165**	**Info(age)D**	**0.47429565**		
17					**Gain**	**0.424735121**		
18	**Salary**	**Junior**	**Senior**					
19	26k_30k	46	0	**46**	0			
20	31k_35k	40	0	**40**	0			
21	36k_40k	0	4	**4**	0			
22	41k_45k	4	0	**4**	0			
23	46k_50k	23	40	**63**	0.946818832			
24	66k_70k	0	8	**8**	0			
25	**Total**	**113**	**52**	**165**	**Info(Salary)D**	**0.361512645**		
26					**Gain**	**0.537518126**		
27								
28	Second layer calcs	(between 35 and 66k salary)					**Info(D)**	0.958241
29	department	Junior	Senior	**Total**				
30	sales	0	30	**30**	0			
31	systems	23	0	**23**	0			
32	marketing	4	10	**14**	0.863120569			
33	secretary	0	4	**4**	0			
34	**Total**	27	44	**71**	**Info(department)D**	**0.170192788**		
35					**Gain**	**0.78804794**		
36	age	Junior	Senior	**Total**				
37	21_25	20	0	**20**	0			
38	26_30	3	0	**3**	0			
39	31_35	4	30	**34**	0.522559375			
40	36_40	0	10	**10**	0			
41	46_50	0	4	**4**	0			
42	Total	27	44	**71**	**Info(age)D**	**0.2502397**		
43					**Gain**	**0.708001028**		
44								
45	Third Layer							
46	age	junior	senior					
47	31_35	4	0		0			
48	36_40	0	10		0			

#

Problem 4

Generate If-Then rules for decision tree

Rules:

R1: IF salary $<$ 35k THEN status $=$ Junior

R2: IF salary $>$ 66k THEN status $=$ Senior

R3: IF 35k $<$ salary $<$ 66k AND Department $=$ Systems THEN status $=$ Junior

R4: IF 35k $<$ salary $<$ 66k AND Department $=$ Sales THEN status $=$ Senior

R5: IF 35k $<$ salary $<$ 66k AND Department $=$ Secretary THEN status $=$ Senior

R6: IF 35k $<$ salary $<$ 66k AND Department $=$ Marketing AND age $=$ 31 - 35 THEN status $=$ Junior

R7: IF 35k $<$ salary $<$ 66k AND Department $=$ Marketing AND age $=$ 36 - 40 THEN status $=$ Senior