Name: _____ **25 points possible**

CS 5402 – Intro to Data Mining
Fall 2020
HW #3

Submit as a <u>single</u> pdf file via Canvas by 11:59 p.m. on Sep. 30, 2020

1. Consider the following dataset:

married	sportPref	income	pet	drinkPref	musicPref
no	football	low	dog	coke	rock
yes	football	low	dog	pepsi	classical
no	football	low	dog	coke	rock
yes	baseball	middle	cat	tea	country
yes	hockey	middle	cat	tea	country
no	baseball	high	snake	pepsi	jazz
no	football	middle	dog	coke	classical

a. Compute the **coverage** of each item set listed below. **(1 pt.)**

<u>Item Set</u>	**<u>Coverage</u>**
married = no, ***pet*** = dog	___
sportPref = football, ***musicPref*** = classical, ***drinkPref*** = tea	___

b. Write down <u>every</u> **association rule** that could be <u>generated</u> from the item set listed below, regardless of whether or not there are actually any instances of that rule in our given dataset. **(1.5 pts.)**

married = no, ***pet*** = dog

c. Compute the **accuracy** of each rule listed below. Express accuracy as a **<u>fraction</u>** (e.g., 2/3, 2/2, etc.), **<u>NOT</u>** as a decimal number (e.g., 0.67, 1.0, etc.). **(1.5 pts.)**

<u>Rule</u>	**<u>Accuracy</u>**
If *pet* = dog then *income* = middle	_____
If *married* = no and *sportPref* = *football* then *pet* = dog and *musicPref* = rock	_____
If _ then *drinkPref* = coke and *married* = yes	_____

Name: _____ **25 points possible**

d. In each rule listed below, specify whether any condition(s) in the antecedent (i.e., the "if' part) can be **dropped** without losing accuracy. **(1 pt.)**

If *married* = no and *sportPref* = football then *pet* = dog and *drinkPref* = coke

If *married* = yes and *pet* = cat then *musicPref* = country

2. Posted on Canvas along with this assignment is the source code for an implementation of the **Prism** algorithm (**prism.py and Arff2Skl.py**). It doesn't work; try it on the file contact-lenses.arff (which is posted on Canvas in Files->Files for Weka Examples) and you'll see that it returns no rules. Debug the code and fix it. Explain (in detail) what you had to do to fix the code **AND** show the results of running it on contact-lenses.arff. You will not get credit for showing results of running it on contact-lenses.arff unless you show how to correctly fix the program! **(7 pts.)**

Name: _____ **25 points possible**

3. Consider the dataset shown below where the decision attribute is ***d***. If attribute weights $\mathbf{w_a}$, $\mathbf{w_b}$, and $\mathbf{w_c}$ are all initialized to 2, **Θ** is 2, and **α** is 2, what will the **attribute weights** (i.e., $\mathbf{w_a}$, $\mathbf{w_b}$, and $\mathbf{w_c}$) be after **one** iteration of the **Winnow** algorithm? **YOU MUST SHOW YOUR WORK** in computing these values; otherwise, you will receive **NO CREDIT! (1.5 pts.)**

	a	b	c	d
x1	1	0	1	0
x2	0	1	0	1
x3	1	1	0	1
x4	1	0	1	0

Final values: $\mathbf{w_a}$ **= ___** $\mathbf{w_b}$ **= ___** $\mathbf{w_c}$ **= ___**

Name: _____ **25 points possible**

4. Consider the dataset given below where the decision attribute is the one labeled ***decision***. Build a **kd-tree** where **k = 3**. **No partial credit will be given unless you SHOW YOUR WORK! (6.5 pts.)**

   When computing medians, if you have a real number, **round** .1 to .4 **<u>down</u>** to the next integer, and **round** .5 to .9 **<u>up</u>** to the next integer (e.g., round 2.5 to 3, round 2.3 to 2, etc.).

x	y	z	decision
10	27	9	0
20	25	8	0
30	26	7	1
40	4	0	0
50	3	4	1
60	1	16	1
70	2	12	0

Name: _____ **25 points possible**

5. Consider the dataset given below where the decision attribute is the one labeled ***class***. Show how **k-means clustering** using **k = 3** would cluster the instances on attributes ***a*** and ***b*** assuming that the initial cluster centers you start with are **(2, 4)**, **(5, 6)**, and **(8, 1)**. **SHOW ALL OF YOUR WORK!**

   Use **Manhattan distance** for your calculations. When computing centers, if you have a real number, **round** .1 to .4 **down** to the next integer, and **round** .5 to .9 **up** to the next integer (e.g., round 2.5 to 3, round 2.3 to 2, etc.).

   Do **NOT** draw a graph showing the final clusters; simply specify what the clusters will be in terms of **what each cluster's center is and what instances from the dataset will be in each cluster**. **(5 pts.)**

a	b	c	class
2	4	11	true
5	6	5	false
8	1	7	false
7	3	4	true
4	10	8	true
3	0	3	true
9	8	1	false