CSC 458 Data Mining and Predictive Analytics I, Fall 2019

Dr. Dale E. Parson, Assignment 2, Using Weka rules and trees to correlate several meteorological and temporal attributes with Hawk Mountain raptor migration counts for fall 2017 & fall 2018. Due by 11:59 PM on Saturday October 19 via make turnitin.

Perform the following steps to set up for this semester's projects and to get my handout. Start out in your login directory on csit (a.k.a. acad).

```
cd  $HOME
mkdir  DataMine  # This should already be there from assignment 1.
cp  ~parson/DataMine/csc458fall2019assn2.problem.zip DataMine/csc458fall2019assn2.problem.zip
cd   ./DataMine
unzip  csc458fall2019assn2.problem.zip
cd  ./csc458fall2019assn2
```

EDIT THE SUPPLIED FILE README.txt when the following questions starting at Q1 below. Keep with the supplied format, and do not turn in a Word or PDF or other file format. I will deduct 20% for other file formats, because with this many varying assignments being turned in, I need a way to grade these in reasonable time, which for me is a batch edit run on the **vim** editor.

Running Weka

If you do your work on campus PCs, make sure to save your work under your networked U:\ so you do not lose work. Campus PCs discard any file changes when you log out. You must still get your files to me via acad. You can also download Weka to your own machine and work there. Campus PC users can just run S:\ComputerScience\WEKA\WekaWith4GBcampus, which contains this batch command:

```
java -Xmx4096M -jar "S:\ComputerScience\WEKA\weka.jar"
```

Starting Weka from the command line allows you to increase its memory allotment, to 4GB in this case. Here is my command-line command on my Mac:

```
java -server -Xmx4000M -jar /Applications/weka-3-6-9/weka.jar
```

If you are using a computer with relatively modest amount of memory, the default memory size for Weka should be sufficient for assignment 2. This dataset is not as big as last year's.

PART I – Preparing the data. 5% for Q1 + 10% for the correct saved ARFF file.

1. Open ARFF file **JoinedHawkMtn20172018.arff** in Weka and observe that the attribute names and types in your dataset match those below; bring the Edit Preprocessor Window up, full screen, and scroll around inspecting for missing values that are grayed out in this Editor.

Date	date of Hawk Mountain observation
Start	starting hour of Hawk Mountain observation
End	ending hour of Hawk Mountain observation
Duration	duration of Hawk Mountain observation in minutes
Observer	duration primary observer's of Hawk Mountain observation in minutes
BV	Black Vulture count for that hour

TV	Turkey Vulture count for that hour
UV	Unidentified Vulture count for that hour
MK	Mississippi Kite count for that hour
OS	Osprey count for that hour
BE	Bald Eagle count for that hour
NH	Northern Harrier count for that hour
SS	Sharp-shinned Hawk count for that hour
CH	Cooper's Hawk count for that hour
NG	Northern Goshawk count for that hour
UA	Unidentified Accipiter count for that hour
RS	Red-shouldered Hawk count for that hour
BW	Broad-winged Hawk count for that hour
SW	Swainson's Hawk count for that hour
RT	Red-tailed Hawk count for that hour
RL	Rough-legged Hawk count for that hour
UB	Unidentified Buteo count for that hour
GE	Golden Eagle count for that hour
UE	Unidentified Eagle count for that hour
AK	American Kestrel count for that hour
ML	Merlin count for that hour
PG	Peregrine Falcon count for that hour
UF	Unidentified Falcon count for that hour
UR	Unidentified Raptor count for that hour
OR	Other raptor count for that hour
TOTAL	Total raptor count for that hour
WindSpd	North lookout wind speed as a nominal value, via portable anemometer
WindDir	North lookout wind direction
Temp	North lookout Celsius temperature
CloudCover	North lookout cloud cover, units of measure unknown
Visibility	North lookout visibility, units of measure unknown
FlightDIR	Raptor nominal flight direction (SE, etc.)
FlightHT	Raptor flight height as a nominal value
SkyCode	
Counter	Primary human counter
Observer1	Additional human observer
Observer2	Additional human observer
Observer3	Additional human observer
Observer4	Additional human observer
HawkYear	2017 or 2018 for this dataset
hawkStart	Combined Date and Start from above. This gives a complete, unique time stamp.
hawkEnd	Combined Date and End from above. This gives a complete, unique time stamp.
msnyHstart	Minutes since previous New Year (Jan. 1, 00:00) for hawkStart.
msmnHstart	Minutes since observation day’s previous midnight (00:00) for hawkStart.

msnyHend	Minutes since previous New Year (Jan. 1, 00:00) for hawkEnd.
msmnHend	Minutes since observation day’s previous midnight (00:00) for hawkEnd.
datetime	Complete date and time for nearest previous weather data from weather station.[1]
station	Weather station ID.
student	Student weather data collector for assignment 1.
tempc	Weather station temperature in Celsius.
dewpc	Weather station dew point in Celsius.
humidity	Weather station % humidity as a fraction of 1.0.
winddir	Weather station wind direction as nominal.
windspd	Weather station wind speed in MPH.
windgust	Weather station wind gust speed in MPH.
barompres	Weather station barometric pressure in inches.
preciprt	Weather station precipitation in inches.
precipaccum	Weather station accumulated precipitation in inches.
msnyWeath	Minutes since previous New Year (Jan. 1, 00:00) for weather datetime.
msmnWeath	Minutes since observation day’s previous midnight (00:00) for weather datetime.
SunDate	Date of sunrise/sunset measurement, same as hawkStart.
Sunrise	Time of sunrise measurement, same as hawkStart.
Sunset	Time of sunset measurement, same as hawkStart.
SunMinutes	Duration of sunrise-to-sunset in minutes, same as hawkStart.
SunYear	Year of sunrise/sunset observation.
sunSunrise	Complete date + time timestamp for SunDate + SunRise.
sunSunset	Complete date + time timestamp for SunDate + SunSet.
msnySunrise	Minutes since previous New Year (Jan. 1, 00:00) for sunSunrise.
msmnSunrise	Minutes since observation day’s previous midnight (00:00) for sunSunrise.
msnySunset	Minutes since previous New Year (Jan. 1, 00:00) for sunSunset.
msmnSunset	Minutes since observation day’s previous midnight (00:00) for sunSunset.
tempPrev24	Weather station temperature in Celsius taken ~ 24 hours before this record.[2]
humidPrev24	Weather station % humidity taken ~ 24 hours before this record.
baromPrev24	Weather station barometric taken ~ 24 hours before this record.

On 9/18/2019 Dr. Laurie Goodrich wrote: "The minutes for hour are minutes covered for hour and observer min are number of observers working times minutes. I would just use minutes and not worry about the observer number as the effect is not additive exactly."

For now we will just use the start time in HawkStart (below) and assume 60 minutes Duration per row of data. Rarely does the Duration value exceed 60, and that is on low-count days. From Weka the mean for Duration is 54.7.

2. Run Weka’s **unsupervised -> attribute -> RemoveUseless** filter.

[1] All timestamps in this dataset are Eastern Standard Time. I have corrected the wunderground EDT times.

[2] Within 15 minutes of previous 24-hour time, usually much closer.

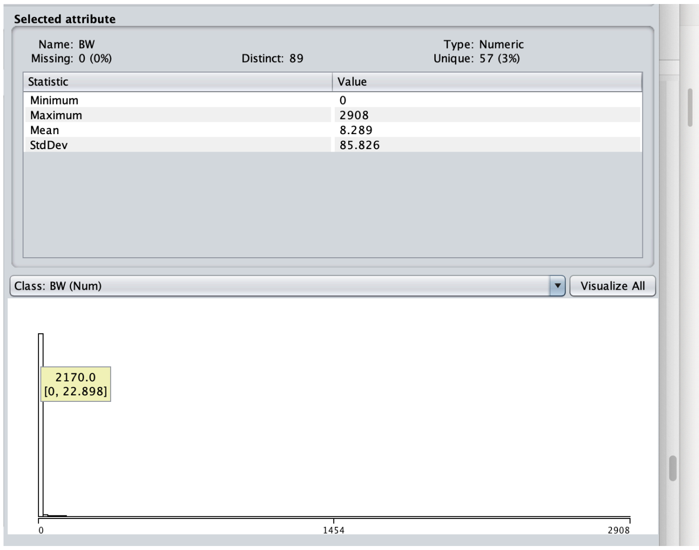
Q1 in README.txt: Which attributes did RemoveUseless remove, and why? Be specific for each. Read the pop-up RemoveUseless documentation in Weka, and execute Undo in the Weka preprocessor if you need to inspect the pre-RemoveUseless attribute values. Make sure to re-run RemoveUseless if you execute Undo. Click **More** in all of the Filters and Classifiers you use to browse documentation.

3. Run the Weka **Unsupervised -> Attribute -> StringToNominal** preprocessor filter on all attributes. This conversion will affect only string-valued attributes, converting them to sets of nominal (symbolic) values, which Weka can use in pattern matching. Make sure that no string-valued attributes remain.
4. **Delete all attributes associated with people's names or people's IDs**. This is a subset of the string-to-nominal attributes in the previous step. We will not correlate humans with counts in this assignment.
5. Run the Weka **Unsupervised -> Attribute -> RemoveType** preprocessor filter on all **date**-type attributes. This conversion will delete only date-valued attributes. We do not need them because the msny* and msmn* attributes are integers and easier to interpret that the **date** timestamps within Weka. Make sure that no date attributes remain, and that no other attributes are lost.
6. Delete the **Duration** and **Observer** (minutes) attributes, since we are assuming 60 minutes of observations for the vast majority of rows of data. See Dr. Laurie Goodrich's statement of 9/18/2019 above.
7. Delete ALL raptor counts, **BV** through **TOTAL**, except **BW** (KEEP Broad-winged Hawk count for each row in the data). We will analyze several aspects of the **BW** counts for 2017 and 2018.
8. Use the preprocessor's **AddExpression** to add a new attribute named **baromdelta** that is the record's value of **barompress** MINUS its **baromPrev24** value. Derived attribute **baromdelta** gives the rise or fall in barometric pressure for the last 24 hours. AddExpression addresses attribute 1 as **a1**, attribute 2 as **a2**, etc. You need to use the attribute indices for **barompress** and **baromPrev24** in your expression. Use the AddExpression **More** help faculty as needed.[3] Make sure there is no trailing space in the name **baromdelta**.
9. Similarly, use preprocessor's **AddExpression** to add a new attribute named **tempdelta** that is the record's value of **tempc** MINUS its **tempPrev24** value from the weather station. This is the rise or fall in temperature from 24 hours previous.
10. Run the Weka **Unsupervised -> Attribute -> Reorder** filter to **make BW the last attribute** in the display without changing the order of any other attributes. BW is numeric at this point. The rule- and tree-based classifiers use the last position as the default position of the *class* (a.k.a. *target attribute*).
11. **Save this ARFF dataset as FilteredCSC458assn2.arff**, and make sure to copy this file into your handout directory alongside your README.txt file before running **make turnitin** by the due date. **There should be 36 attributes with BW as the last in FilteredCSC458assn2.arff**, with all of the preceding ones relating to weather, other physical conditions, or times since New Year's and midnight. This file is the modified dataset for your analysis. You can always re-load this file after taking a break from the remaining steps. If you do not have 36 attributes, go back and check your steps. Having different attributes will change your results in the subsequent steps, and could result in point deductions.

PART II – Analyzing the data. 5% for each of Q1 through Q18.

[3] According to one source, "If barometric pressure rises or falls more than 0.18 in-Hg in less than three hours, barometric pressure is said to be changing rapidly. A change of 0.003 to 0.04 in-Hg in less than three hours indicates a slow change in barometric pressure. A change of less than 0.003 in-Hg in less than three hours is considered to be holding steady." https://sciencing.com/high-low-reading-barometric-pressure-5814364.html

Initially we have a distribution histogram for numeric BW that looks like this. There are many low-value sightings for a given hour, a few larger-valued, and one very-large-valued sightings for BW within an hour.



Selected attribute

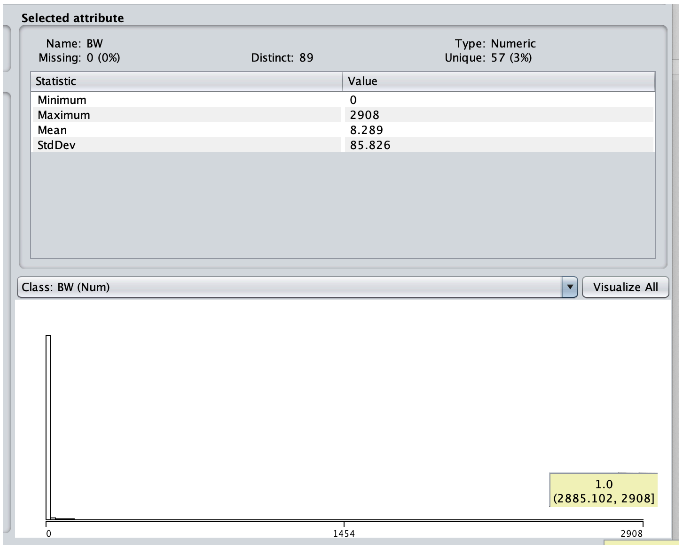
Name: BW
Missing: 0 (0%)
Distinct: 89
Type: Numeric
Unique: 57 (3%)

Statistic	Value
Minimum	0
Maximum	2908
Mean	8.289
StdDev	85.826

Class: BW (Num)

Figure 1

Notice that the leftmost, high-occurrence bar represents 0 through 22 BW sightings in those hours. The next screenshot shows the rightmost end of this histogram with the highest count for exact 1 hour within the dataset, with the count of 2908.



Selected attribute

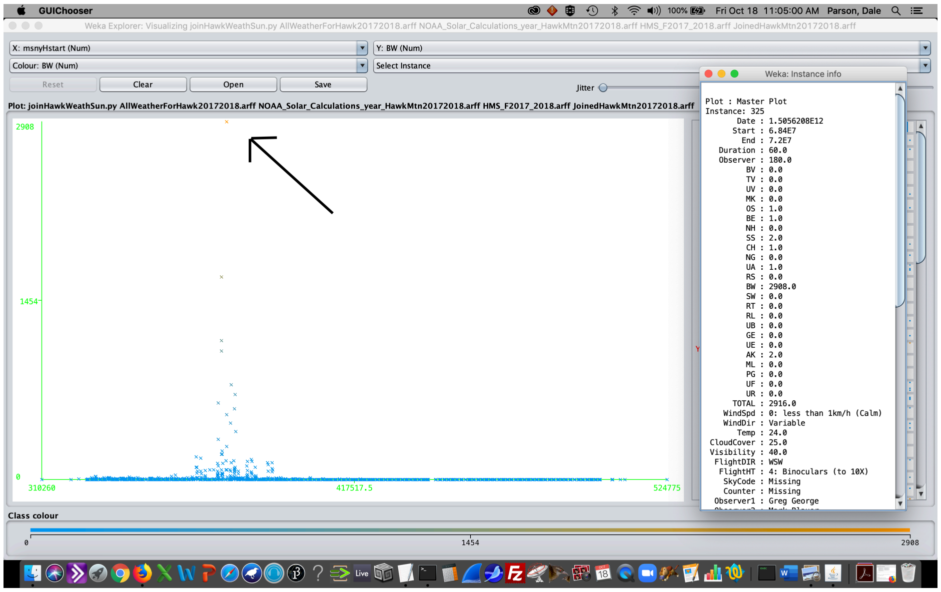
Name: BW
Missing: 0 (0%)
Distinct: 89
Type: Numeric
Unique: 57 (3%)

Statistic	Value
Minimum	0
Maximum	2908
Mean	8.289
StdDev	85.826

Class: BW (Num)

Figure 2

For an alternative perspective, the next screenshot from Weka's Visualize tab shows the BW count in the Y direction as a function of minutes-from-New-Year, i.e., the minute of the year, in the X. Note the pop-up for the 2908 observation at hour xxx. Some of the earlier days are covered up by the pop-up.



Weka Explorer: Visualizing joinHawkWeathSun.py AllWeatherForHawk20172018.arff NOAA_Solar_Calculations_year_HawkMtn20172018.arff HMS_F2017_2018.arff JoinedHawkMtn20172018.arff

X: msnyHstart (Num)

Y: BW (Num)

Colour: BW (Num)

Select Instance

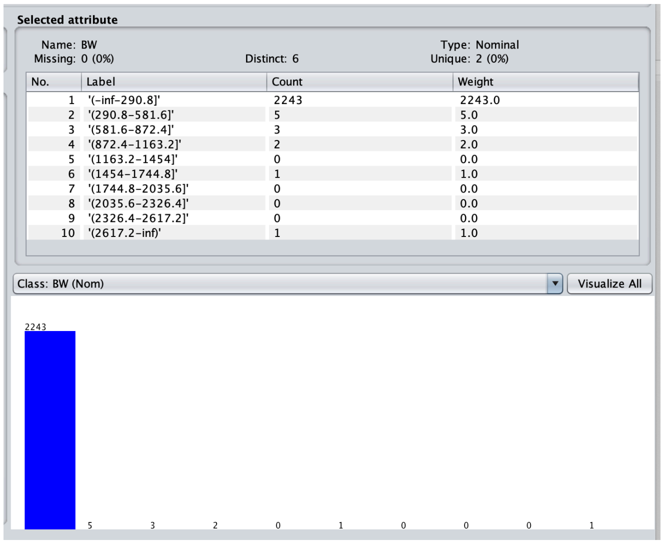
Plot: joinHawkWeathSun.py AllWeatherForHawk20172018.arff NOAA_Solar_Calculations_year_HawkMtn20172018.arff HMS_F2017_2018.arff JoinedHawkMtn20172018.arff

Weka: Instance info

```
Plot : Master Plot
Instance: 325
       Date : 1.5056208E12
      Start : 6.84E7
        End : 7.2E7
   Duration : 60.0
   Observer : 180.0
         BV : 0.0
         TV : 0.0
         UV : 0.0
         MK : 0.0
         OS : 1.0
         BE : 1.0
         NH : 0.0
         SS : 2.0
         CH : 1.0
         NG : 0.0
         UA : 1.0
         RS : 0.0
         BW : 2908.0
         SW : 0.0
         RT : 0.0
         RL : 0.0
         UB : 0.0
         GE : 0.0
         UE : 0.0
         AK : 2.0
         ML : 0.0
         PG : 0.0
         UF : 0.0
         UR : 0.0
      TOTAL : 2916.0
    WindSpd : 0: less than 1km/h (Calm)
    WindDir : Variable
       Temp : 24.0
 CloudCover : 25.0
 Visibility : 40.0
  FlightDIR : WSW
   FlightHT : 4: Binoculars (to 10X)
    SkyCode : Missing
    Counter : Missing
  Observer1 : Greg George
```

Figure 3 (corrected from msmnHstart to msnyHstart on X axis, 10/18/2019)

The classifiers of Assignment 2 require the *class* (*target attribute*) to be of nominal type, i.e., discrete values. Use Weka's **Unsupervised -> Attribute -> Discretize** filter to break **<u>ONLY</u>** the BW attribute into 10 bins according to the BW unit of measure, i.e., linear count subranges. Leave the **useEqualFrequency** configuration parameter of **Discretize** to **False**, and **ignoreClass** to **True**. The former partitions the bins by linear unit measures, and **ignoreClass=True** is necessary since we are discretizing, and therefore changing, the attribute that we are trying to predict. You should wind up with a nominal BW distribution like this. Make sure not to discretize any other attributes.



Selected attribute

Name: BW
Missing: 0 (0%)
Distinct: 6
Type: Nominal
Unique: 2 (0%)

No.	Label	Count	Weight
1	'(-inf-290.8]'	2243	2243.0
2	'(290.8-581.6]'	5	5.0
3	'(581.6-872.4]'	3	3.0
4	'(872.4-1163.2]'	2	2.0
5	'(1163.2-1454]'	0	0.0
6	'(1454-1744.8]'	1	1.0
7	'(1744.8-2035.6]'	0	0.0
8	'(2035.6-2326.4]'	0	0.0
9	'(2326.4-2617.2]'	0	0.0
10	'(2617.2-inf)'	1	1.0

Class: BW (Nom)

Figure 4

Q2: In Weka's Classify tab, run the **rule -> ZeroR classifier**, and paste the following results into README.txt at Q2. We will discuss some of the error measures later. For now we focus on Kappa. Also, make sure you understand the confusion matrix, but **<u>don't</u>** copy it into README.txt.

```
ZeroR predicts class value: '(RANGE]'
Correctly Classified Instances       N           N.n %
Incorrectly Classified Instances       N           N.n %
Kappa statistic                    N
Mean absolute error                    N.n
Root mean squared error                  N.n
Relative absolute error                N     %
Root relative squared error            N     %
Total Number of Instances              2255
```

Q3: How did ZeroR predict the class?

Q4: For this dataset and class, what is the expected accuracy of ZeroR?

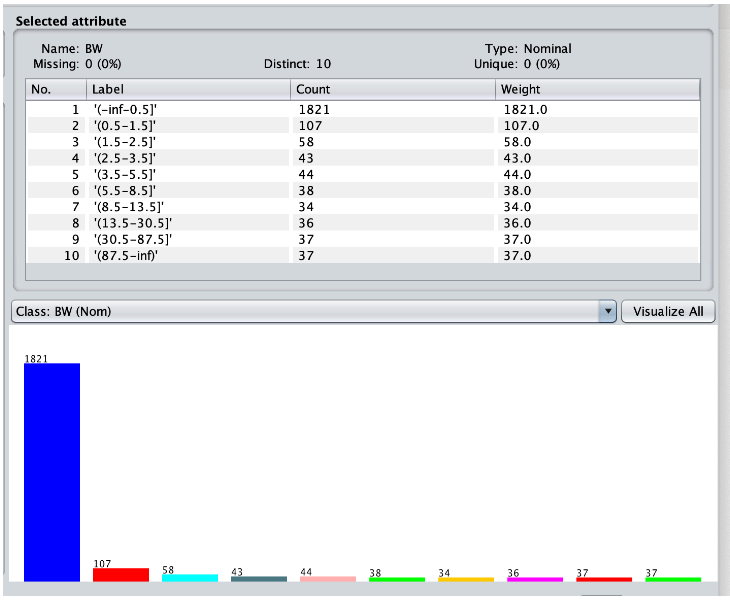
Q5: Run the **rule -> OneR classifier**, and paste the following results into README.txt at Q5. Do you see any improvement over ZeroR? Explain. What is the predictive non-target attribute in the rule?

```
Correctly Classified Instances       N           N.n %
Incorrectly Classified Instances       N           N.n %
Kappa statistic                    N
Mean absolute error                    N.n
Root mean squared error                  N.n
Relative absolute error                N     %
Root relative squared error            N     %
Total Number of Instances              2255
```

Q6: Run the **tree -> J48 classifier**, and paste the following results into README.txt at Q6. Do you see any improvement over OneR? Is the "J48 pruned tree" reported closer in structure to ZeroR or OneR?

```
Correctly Classified Instances       N           N.n %
Incorrectly Classified Instances       N           N.n %
Kappa statistic                    N
Mean absolute error                    N.n
Root mean squared error                  N.n
Relative absolute error                N     %
Root relative squared error            N     %
Total Number of Instances              2255
```

SETUP: From Weka's Preprocess tab, execute Undo once to get BW back to numeric type, then re-run Discretize as before, but with **useEqualFrequency** set to True. You should get a BW distribution like this:



Selected attribute

Name: BW
Missing: 0 (0%)
Distinct: 10
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-0.5]'	1821	1821.0
2	'(0.5-1.5]'	107	107.0
3	'(1.5-2.5]'	58	58.0
4	'(2.5-3.5]'	43	43.0
5	'(3.5-5.5]'	44	44.0
6	'(5.5-8.5]'	38	38.0
7	'(8.5-13.5]'	34	34.0
8	'(13.5-30.5]'	36	36.0
9	'(30.5-87.5]'	37	37.0
10	'(87.5-inf)'	37	37.0

Class: BW (Nom)

Figure 5

Note the varying ranges of values from bin to bin. The value distributions are too widely scattered to make the bins of equal height, but they are more evenly distributed than before. Next re-run previous steps on this BW class as follows:

Q7: In Weka's Classify tab, run the **rule -> ZeroR classifier**, and paste the following results into README.txt at Q7. Pay attention to Kappa values as measures of improvement.
ZeroR predicts class value: '(RANGE]'
Correctly Classified Instances N N.n %
Incorrectly Classified Instances N N.n %
Kappa statistic N
Mean absolute error N.n
Root mean squared error N.n
Relative absolute error N %
Root relative squared error N %
Total Number of Instances 2255

Q8: For this dataset and class, what is the expected accuracy of ZeroR? Why did it change in this direction compared to the **useEqualFrequency**=False dataset?

Q9: Run the **rule -> OneR classifier**, and paste the following results into README.txt at Q9. Is there any improvement over ZeroR for this dataset?

(AAAA/2255 instances correct)
Correctly Classified Instances BBBB N.n %
Incorrectly Classified Instances N N.n %
Kappa statistic N
Mean absolute error N.n

```
Root mean squared error            N.n
Relative absolute error          N     %
Root relative squared error        N     %
Total Number of Instances          2255
```

Q10: Copy and paste the entire OneR rule into Q10 in README.txt. Can you observe a pattern in the relationship between the non-target attribute that OneR selects and the target attribute? Explain it in terms of one of the illustrations above, **giving the Figure number in your answer**.

```
=== Classifier model (full training set) ===

Predictive-target-attribute-name (paste the actual attribute name):
        THESE ARE THE RULE LINES YOU MUST COPY & PASTE.
(M/N instances correct)
```

Q11: Run the **tree -> J48 classifier**, and paste the following results into README.txt at Q11. Do you see any improvement over OneR for this dataset?[4]

```
Correctly Classified Instances       N              N.n %
Incorrectly Classified Instances       N              N.n %
Kappa statistic                    N
Mean absolute error                   N.n
Root mean squared error                N.n
Relative absolute error            N     %
Root relative squared error          N     %
Total Number of Instances            2255
```

Parson’s Observation: The BW histograms so far are severely skewed towards 0 BW counted in an hour. There are several reasons for these zeroes:

1. The BWs are not here yet. Their start of migration is likely triggered by remote conditions.
2. The local conditions may not be conducing to flight over North Lookout. We can look for that.
3. The BWs have passed PA and are gone to the south. This happens later in the season.

Also, I believe for this assignment we should re-count BWs in the following categories (classes):

A. We did not count any BWs this hour.
B. We counted a smallish number of BWs this hour.
C. We counted a midrange number of BWs this hour.
D. We counted a large number of BWs this hour.

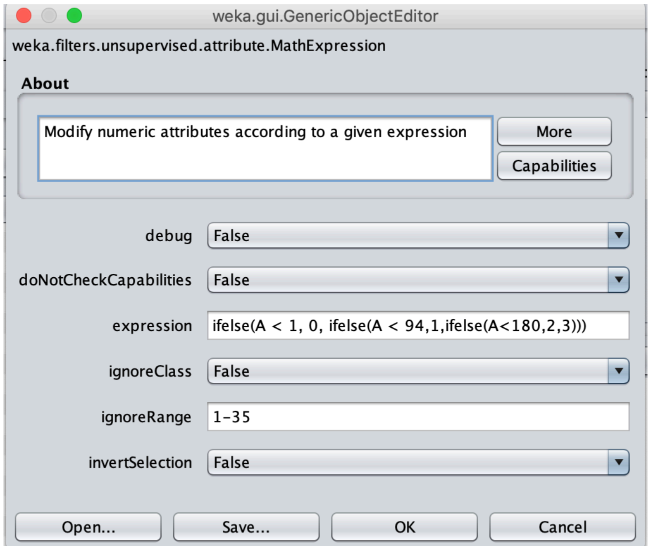
First we are going to discretize the BW counts into classes A through D. We will address conditions 1 through 2 after that. Please follow the following Weka filter steps.

STEP1: In the Preprocessor execute **UNDO** so that BW returns to being a number.

[4] Note on this notation in Weka trees: “'(-inf-0.5]' (1308.0/24.0)” The 1308.0 refers to how many instances arrived at this class via the tree, and 24.0 is the number of those that were incorrect. Numbers may include fractions where there are ‘?’ unknown values involved.

STEP2: Execute filter **unsupervised -> attribute -> Copy** with an attribute index of **last**. This step creates attribute **Copy of BW**. We are doing this for two reasons. First, the next filter actually changes its attribute values. Unlike AddExpression, MathExpression does not create a new attribute, so we do that with **Copy**. Second, I cannot get MathExpression to work with the target class, even if I set ignoreClass to True. Therefore, we will mutate BW, leaving Copy BW as unaltered.

STEP3: Execute precisely this **unsupervised -> attribute -> MathExpression** to mutate BW. The values are based on the goal of separating 0 BWs from the others, and using the mean of 8.289 and the standard deviation of 85.826 reported by Weka's Preprocessor for numeric BW.



weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.MathExpression

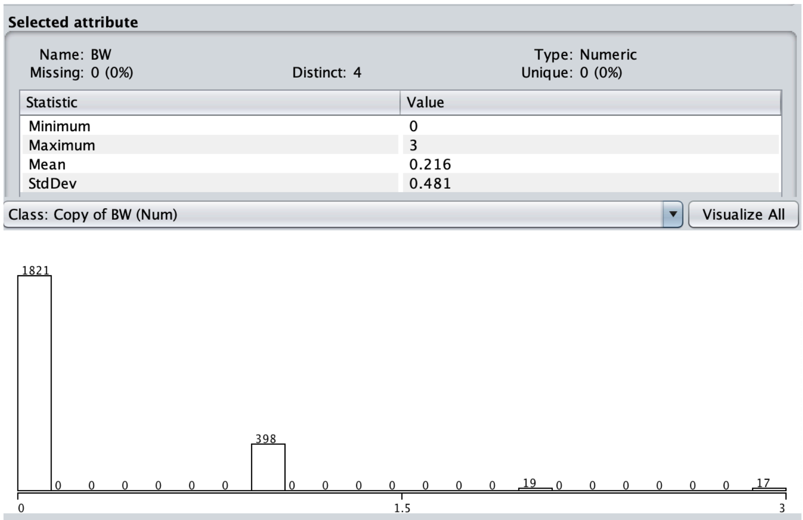
About

Modify numeric attributes according to a given expression

debug	False
doNotCheckCapabilities	False
expression	ifelse(A < 1, 0, ifelse(A < 94,1,ifelse(A<180,2,3)))
ignoreClass	False
ignoreRange	1-35
invertSelection	False

Figure 6

STEP4: Check to ensure that attribute BW has the distribution of Figure 7. The attributes preceding BW must not be altered by MathExpression; undo and fix ignoreRange if you changed them by accident. Also, do not Apply MathExpression more than once, since each Apply alters BW further.



Selected attribute

Name: BW
Missing: 0 (0%)
Distinct: 4
Type: Numeric
Unique: 0 (0%)

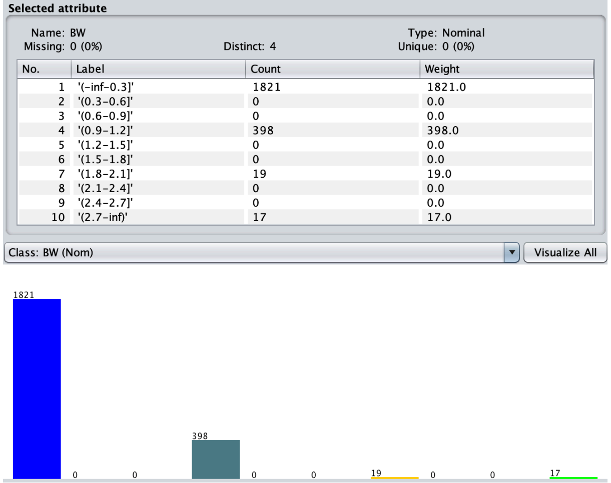
Statistic	Value
Minimum	0
Maximum	3
Mean	0.216
StdDev	0.481

Class: Copy of BW (Num)

Figure 7

STEP5: Run filter **Unsupervised -> Attribute -> Discretize** filter to break ONLY the BW attribute into **10** bins according to the BW unit of measure, i.e., linear count subranges. Leave the **useEqualFrequency**

configuration parameter of **Discretize** to **False**, and **ignoreClass** to **True**. You should wind up with a colored nominal histogram that looks like Figure 8. We are not splitting into 4 bins because we would wind up with the distribution of Figure 9 that redistributes BW values into bins unlike our custom discretization. Make sure not to accidentally discretize any other attributes.



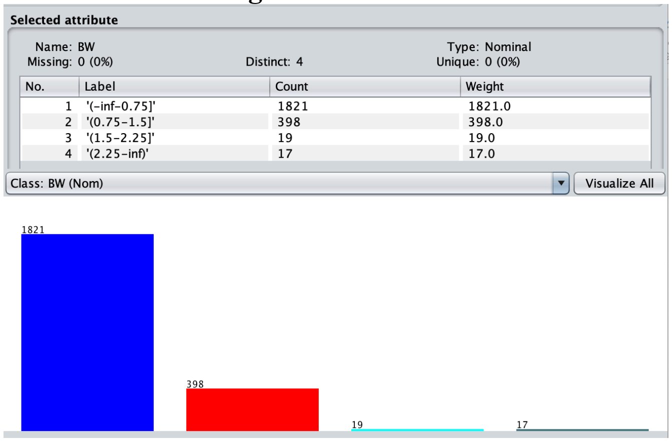
Selected attribute

Name: BW
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-0.3]'	1821	1821.0
2	'(0.3-0.6]'	0	0.0
3	'(0.6-0.9]'	0	0.0
4	'(0.9-1.2]'	398	398.0
5	'(1.2-1.5]'	0	0.0
6	'(1.5-1.8]'	0	0.0
7	'(1.8-2.1]'	19	19.0
8	'(2.1-2.4]'	0	0.0
9	'(2.4-2.7]'	0	0.0
10	'(2.7-inf)'	17	17.0

Class: BW (Nom)

Figure 8 is correct



Selected attribute

Name: BW
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-0.75]'	1821	1821.0
2	'(0.75-1.5]'	398	398.0
3	'(1.5-2.25]'	19	19.0
4	'(2.25-inf)'	17	17.0

Class: BW (Nom)

Figure 9 is incorrect

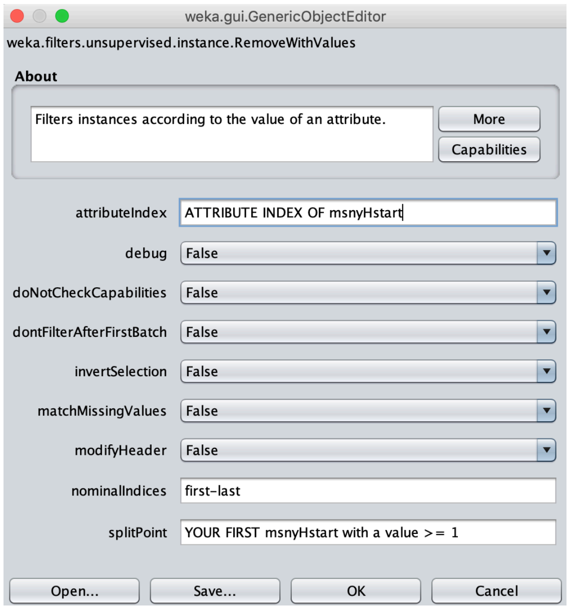
Q12: Remove attribute **Copy of BW**; **BW** of Figure 8 is then the target attribute. Rerun the **OneR**, **J48,** and **RandomTree** classifiers on this dataset. Report their kappa values. How do they compare to kappa values in previous steps?

OneR kappa:
J48 kappa:
RandomTree:

STEP6: The final Preprocessing step is to get rid of all 0 BW records that precede the first non-0 BW observation for the combined years, as well as all 0 BW record that follow the final non-0 BW record. We do **not** want to remove BW == 0 records mixed into the middle. The goal is to find out what other attributes than time of year are significant only during the BW fly-over time period. Q13 has the actual edits.

Q13: Go into the Preprocess -> Edit window and sort on **msnyHstart** to get the observation's start hour since New Year (not since midnight). Click the **msnyHstart** heading and make sure the records sort on it. Scroll right and find the first BW count that is NOT less than 1, then scroll back and find its msnyHstart value. What is that msnyHstart value? Run Weka filter **unsupervised -> instance -> RemoveWithValues**

using Figure 10's parameters. After Applying it, check the msnyHstart resulting range and the resulting number of instances to ensure that you have removed only instances with instances before your split point. msnyHstart. How many instances remain?



weka.filters.unsupervised.instance.RemoveWithValues

Filters instances according to the value of an attribute.

attributeIndex: ATTRIBUTE INDEX OF msnyHstart
debug: False
doNotCheckCapabilities: False
dontFilterAfterFirstBatch: False
invertSelection: False
matchMissingValues: False
modifyHeader: False
nominalIndices: first-last
splitPoint: YOUR FIRST msnyHstart with a value >= 1

Figure 10

Q14: Go into the Preprocess -> Edit window and sort in descending order on **msnyHstart** to get the observation's start hour since New Year (not since midnight). Shift-Click the **msnyHstart** heading and make sure the records sort on it. Scroll right and find the last BW count that is less than 1, then scroll back and find its msnyHstart value. What is that msnyHstart value? Run Weka filter **unsupervised -> instance -> RemoveWithValues** using Figure 10's parameters, except setting **invertSelection to True** so you remove the later dates without BWs. After Applying it, check the msnyHstart resulting range and the resulting number of instances to ensure that you have removed only instances with instances at or after your split point. msnyHstart? How many instances remain?

Q15: Rerun the **OneR**. **J48** and **RandomTree** classifiers on this dataset. Report their kappa values. How do they compare to kappa values in previous step, before removing 0-BW instances at the start and end of their North Lookout transit time period?
OneR kappa:
J48 kappa:
RandomTree:

STEP7: Hit Undo twice in the Preprocess tab to restore all instances in the dataset. It should be twice unless you have applied RemoveWithValues more times. Just watch the "Instances:" count in the Preprocess tab until it goes back to the original value. Do not Undo further.

Q16: Run the Select Attributes tab with its default parameters and list the selected parameters here:

Selected attributes: (INDICES)
LIST THEM HERE

STEP8: Remove all attributes that are not in Q16's list except BW. Keep only BW and the listed Q16 attributes.

Q17: Rerun the **OneR**. **J48** and **RandomTree** classifiers on this dataset after removing the non-selected attributes. Report their kappa values. How do they compare to kappa values in previous steps Q12 and Q15?
OneR kappa:
J48 kappa:
RandomTree:

Q18: **Remove** all msny* and SunMinutes attributes to get rid of any correlations with day of year (look at the previous OneR run's non-target attribute in the rule). Rerun the **OneR**. **J48** and **RandomTree** classifiers on this dataset after removing the non-selected attributes. Report their kappa values. They will have gotten a little lower than Q17. We are doing this step to look at non-day-of year physical attributes' correlation to BW. Which kappa decreased the least out of the three classifiers going from Q17?
OneR kappa:
J48 kappa:
RandomTree:

Inspect Q18's OneR rule and J48 and RandomTree trees to see how the attributes are used.

When you have completed all of your work and double-checked the assignment requirements, make sure that both FilteredCSC458assn2.arff saved in a previous step, and your **README.txt** that answers Q1 through Q18, are sitting in your **csc458fall2019assn2**/ directory, then run **make turnitin** by the due date. Late assignments lose 10% per day late, and I will not accept an assignment after I go over its solution in class.