

**Dr. Dale E. Parson, Assignment 4, Python datetime lagging for time series analysis.**

**DUE By 11:59 PM on Thursday December 3, 2020 via make turnitin on mcgonagall or acad. The standard 10% per day deduction for late assignments applies.**

This assignment is a new project using Hawk Mountain Sanctuary raptor observation data from 2017 and 2018. CSC458 used this overall dataset in fall 2019, but we are using some different attributes and doing a completely different time-series analysis.

From acad it will be necessary to **ssh mcgonagall** from acad in order to **time make test**. You can also [download and run the VPN](#) and then putty or ssh into mcgonagall.kutztown.edu directly. Testing must occur on mcgonagall because of increasing CPU load in these projects. If you are on campus, you can log directly into mcgonagall.kutztown.edu without going through acad. Also, you can perform file editing via acad, since acad and mcgonagall share the same student and faculty networked file systems. You can **make turnitin** at the end of the project from either machine.

Perform the following steps to set up for this project. Start out in your login directory.

```
cd $HOME
mkdir DataMine # This should already be there.
cd ./DataMine
cp ~parson/DataMine/TimeRaptors4CSC523.problem.zip TimeRaptors4CSC523.problem.zip
unzip TimeRaptors4CSC523.problem.zip
cd ./TimeRaptors4CSC523
```

This is the directory from which you must run **make turnitin** by the project deadline to avoid a 10% per day late penalty. **Please do not change the name of this directory**, since my test scripts depend on it. Some of the larger data files for this project are in directory ~parson/DataMine, with symbolic links installed to our project directory by **make test** for ease of examining them with an editor.

```
~parson/DataMine/ csc558wn10Ksp2020.arff # The input to your script, copied to your directory.
~parson/DataMine/csc558wn10Ksp2020.arff.ref # Edited output ARFF ref file, a symbolic link.
```

You will see the following files in this **TimeRaptors4CSC523** directory:

TimeRaptors4CSC523.py        The Python program you must complete.

F20172018\_Date\_WS\_Temp\_Vis\_Total\_Num\_Rnd.arff

Your program's initial input file. It contains these attributes.

@attribute datetime string

@attribute WindSpdKmh numeric

@attribute Temp numeric

@attribute Visibility numeric

@attribute TOTAL numeric

**TOTAL** is the target, total number of raptors observed in that observing session, typically one hour.

F20172018\_Deltas\_WS\_Temp\_Vis\_Total\_Num.arff.ref

The reference file that your return result from preprocess(...) must match. It contains these attributes.

- @attribute datetime string
- @attribute WindSpdKmh numeric
- @attribute Temp numeric
- @attribute Visibility numeric
- @attribute TOTAL numeric
- @attribute TOTAL24 numeric
- @attribute Temp24 numeric
- @attribute TOTAL48 numeric
- @attribute Temp48 numeric

TOTAL24 is the fractional change in TOTAL from the observing session 24 hours earlier, e.g. 2.0 for a doubling of TOTAL and 0.5 for a halving. TOTAL48 is the same time-lagged fraction for 48 hours. Both look back to the closest observation that matches 24 or 48 hours previously,  $\pm 4$  hours for inexact matches. Most matches are exact, others within 15 or 30 minutes.

Temp24 is the rise or fall in degrees Celsius since the observing session 24 hours earlier. Temp48 is the rise or fall in degrees Celsius since the observing session 48 hours earlier. These are not fractional changes, they are temperature subtractions (current-previous).

TimeRaptors4CSC523.out.ref Reference file for sys.stdout and print statements from printResults(...).  
makefile & makelib Files needed to **time make test** and **make turnitin**.  
arfflib\_3\_1.py ARFF attribute & data I/O and manipulation module.  
This module is also useful for manipulating in-memory CSV data.

A successful **time make clean test** looks like this. The **bold-highlighted** test will pass after your complete STUDENT requirements in function preprocess(...). The **red-highlighted test** will pass after you complete STUDENT requirements in function analyze(...).

### \$ make clean test

```
/bin/rm -f *.o *.class .jar core *.exe *.obj *.pyc __pycache__/* .pyc
/bin/rm -f junk* *.pyc F20172018_Deltas_WS_Temp_Vis_Total_Num.arff TimeRaptors4CSC523.out.txt
metrics.txt
/bin/rm -f *.tmp *.o *.dif *.out *.csv __pycache__/*
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/DataMine:... time /usr/local/bin/python3.7
TimeRaptors4CSC523.py F20172018_Date_WS_Temp_Vis_Total_Num_Rnd.arff
F20172018_Deltas_WS_Temp_Vis_Total_Num.arff > TimeRaptors4CSC523.out.txt"
egrep -v '@relation' F20172018_Deltas_WS_Temp_Vis_Total_Num.arff | egrep -v '^%' >
F20172018_Deltas_WS_Temp_Vis_Total_Num.tmp
diff F20172018_Deltas_WS_Temp_Vis_Total_Num.tmp
F20172018_Deltas_WS_Temp_Vis_Total_Num.arff.ref >
F20172018_Deltas_WS_Temp_Vis_Total_Num.arff.dif
OUTPUT F20172018_Deltas_WS_Temp_Vis_Total_Num.arff IS OK
egrep COEF TimeRaptors4CSC523.out.txt | sort --stable -n -k13 > metrics.txt
diff TimeRaptors4CSC523.out.txt TimeRaptors4CSC523.out.ref > TimeRaptors4CSC523.out.dif
```

## OUTPUT TimeRaptors4CSC523.out.txt IS OK

Syntax and related error messages to `sys.stderr` will appear on your terminal without looking into output files. If you receive an error from an output difference from a reference file, apply **less** or **tail -40** to the `.dif` file at the end of the error report.

All project code requirements are documented with upper case **STUDENT** comments in `TimeRaptors4CSC523.py`. Percentage values appear below. When **make clean test** runs and you have rechecked all project requirements, use shell command **make turnitin** and follow the prompts to turn the assignment in to me by the deadline.

### \$ make STUDENT

```
grep "STUDENT.*%" TimeRaptors4CSC523.py | sed -e 's/^[^#]*# //' | sort
CSC523 STUDENT A 10%: Convert datetime string objs to datetime objs.
CSC523 STUDENT B 5%: Sort the dataInstances on their datetime field and
CSC523 STUDENT C LOOP 20%: Make [TOTAL24, Temp24, TOTAL48, Temp48] data.
CSC523 STUDENT D 5%: Call imputeARFF(...) to impute the 'mean' value
CSC523 STUDENT E 10%: Use joinARFF(...) to join numeric attributes
CSC523 STUDENT F 5%: Train (fit()) the regressor to the training data
CSC523 STUDENT G 5%: TEST (.predict()) THE TEST DATASET
CSC523 STUDENT H 5%: Invoke, on separate command lines,
CSC523 STUDENT I 5%: shuffle INPUTARFFDATA to randomize dates,
CSC523 STUDENT J 10%: project TOTAL, TOTAL24, and TOTAL48 as separate
CSC523 STUDENT K 5%: Find integer half the length of nontargetDATA
CSC523 STUDENT L 5%: Write outer for loop over 3 target attr variants.
CSC523 STUDENT M 5%: Use slicing to split first half of targetData
CSC523 STUDENT N 5%: Iterate over regressors in INNER LOOP.
grep "STUDENT.*%" TimeRaptors4CSC523.py | wc -l
14
```

When you have completed coding and **make test** passes, **make sure to insert your name and other project documentation at the top of TimeRaptors4CSC523.py**. Run **make test** a final time, then **make turnitin** and follow the prompt to turn in this project by the due date.

You can use your code for Assignment 3 as a template for many of your steps in this project.