Data Mining - tool-based, R-based

Summary: In this Project, you are going to use three UI-based tools (no coding!), to carry out data mining: **WEKA, KNIME, RapidMiner.** There are 3 questions you need to answer - linear regression, using each tool. PLUS, you'll use a couple of **'R'** libraries ('packages') to calculate, and plot, the centroid of the convex hull in your Project_3

Description

WEKA

Start by downloading WEKA, from https://ml.cms.waikato.ac.nz/weka. Note - you can use an older 32-bit version, if your laptop is unable to run the latest 64-bit one. FYI WEKA is written in Java, so you need to install Java [most likely you already have it] prior to installing WEKA. WEKA is powerful and capable - you can continue using WEKA long after this course, and in the future, even consider extending it by writing plugins for it. Here is a short tutorial to bring you up to speed: https://www.tutorialspoint.com/weka/index.htm.

Here is a famous (in the ML/DM community) dataset called the 'Boston Housing Dataset'. As you can read from the description, it is a dataset that contains data regarding houses in several Boston suburbs, published in 1993. It has 506 rows (records) of data, and 14 columns (attributes). For this HW, we'll use the 'MEDV' (median home price) attribute as the "class" (the output to predict). In other words, using existing data from the other 13 columns, we want to be able to learn to predict MEDV for a new record (ie. row) that contains known values for those 13 'input' columns. Note that Zillow, TopHap, etc. routinely carry out such an analysis.

As you can see, the data is in a WEKA-native format called ARFF [https://www.cs.waikato.ac.nz/ml/weka/arff.html], which resembles, but is more descriptive than, CSV.

Q1 (2 points). Build a **linear regression** equation, to predict MEDV. Include a screenshot that shows the linear equation. How many terms are in the equation, and 'why'? In other words, discuss the resulting equation.

KNIME

Here is another dataset to use (scientists go out 'in the field' to painstakingly collect such data! ML might be able to automate some/all of it). It consists of 4177 rows of data regarding abalone shells, where each row resulted from measuring 9 parameters/features/values for each shell. The data is in text format (.arff format, for input to WEKA, like above), do take a look at it. The idea is to be able to predict the 9th value, number-of-rings, given the other 8 values, using the existing dataset to learn how to predict.

Next, download and install KNIME ("nime"), and work through the quickstart tutorial. KNIME is also UI-driven, like WEKA; additionally, it's also visual-dataflow-driven, which means we can do data mining with it, by 'connecting the boxes' (where each box reads data or does mining or writes data, etc).

Q2 (2 points). Use KNIME to perform **linear regression** [on all parameters, not a subset]. You need these nodes: AARF Reader, Linear Regression Learner. Create and connect the nodes, and execute each. What is the linear equation? Include a screenshot.

RapidMiner

Download RapidMiner Studio, and play with it for a bit - it is also dataflow-based, just like KNIME.

Q3 (1.5 points). Bring in the shells.arff data (in the operators list, look under Data Access -> Files -> Read), and only work with these 4 params: length,diameter,height,num_rings (use a 'Select Attributes' node, and type in a regular expression that specifies length,diameter,height,num_rings, or use the 'subset' attribute filter to pick the ones we want - search the documentation for how (additionally, this will help: https://www.youtube.com/watch?v=tQ7oDnQXhmQ). Do a **linear regression** to predict num_rings, from length,diameter,height. Question: what is the equation? Include a screenshot. Note that you need a 'Set Role' node where you would set num_rings to be a "label", before doing the regression (to let the regression node know which attribute to predict, using the other non-label ones). The regression itself would be done using a 'Linear Regression' operator.

If you can't get RapidMiner to work, you can use 'Orange' instead: https://orangedatamining.com/. Note - you'd need to convert .arff to .csv like so [make the edit that's shown, save, rename the file extension to .csv].

'R'

Download and install R first, then RStudio: https://www.r-project.org/, https://rstudio.com/products/rstudio/download/

Bring up RStudio. Do File -> New File -> R Script. In the empty editor window, copy and paste this:

```
##############################################################################################################
```

```
# geosphere is a really cool library for spatial calcs, see
```

```
# https://cran.r-project.org/web/packages/geosphere/vignettes/geosphere.pdf
```

```
install.packages('geosphere')

# use the library to compute polygon stats such as area, perimeter, centroid

library('geosphere')

# the pairs of values are long,lat...

pol <- rbind(c(-180,-20), c(-160,5), c(-60, 0), c(-160,-60), c(-180,-20))

areaPolygon(pol)

perimeter(pol)

# store the centroid in 'c' because we'll need it below, for plotting

c <- centroid(pol)

c

c[1][1]

c[2][1]


################################################################################
##########################################

# leaflet is an R port of the excellent leaflet.js library (https://leafletjs.com/)

# see https://cran.r-project.org/web/packages/leaflet/leaflet.pdf

install.packages("leaflet")

# use leaflet to plot the convex hull coords, plus centroid

library("leaflet")

m <- leaflet()

m <- addTiles(m)

# our centroid - right now, it's the centroid of the 'pol' polygon above

m <- addMarkers(m, lng=c[1][1], lat=c[2][1], popup="Hull centroid")

# our convex hull - right now it's a piece of Venice Beach!!

m <- addCircleMarkers(m, lng=-118.473386, lat=33.985156,label="Ocean Front Walk", radius=2,
fillOpacity=1.0,fill = TRUE, fillColor ="red")
```

```
m <- addCircleMarkers(m, lng=-118.472590, lat=33.985405,label="Muscle Beach", radius=2, fillOpacity=1.0,fill = TRUE, fillColor ="red")

m <- addCircleMarkers(m, lng=-118.473176, lat=33.986269,label="Drum Circle", radius=2, fillOpacity=1.0,fill = TRUE, fillColor ="red")

# time to see the results

m

##############################################################################################################
```

Run the above code, in parts. First select the two install.packages lines, click on the 'Run' button at the top of the editor. If a popup in the console asks you if you want compile from sources, pick 'n'.

After the packages are installed, COMMENT OUT the two lines!

Next, highlight the geosphere section and run it, you'll see the area, perimeter, centroid being printed out (on the console). Cool!

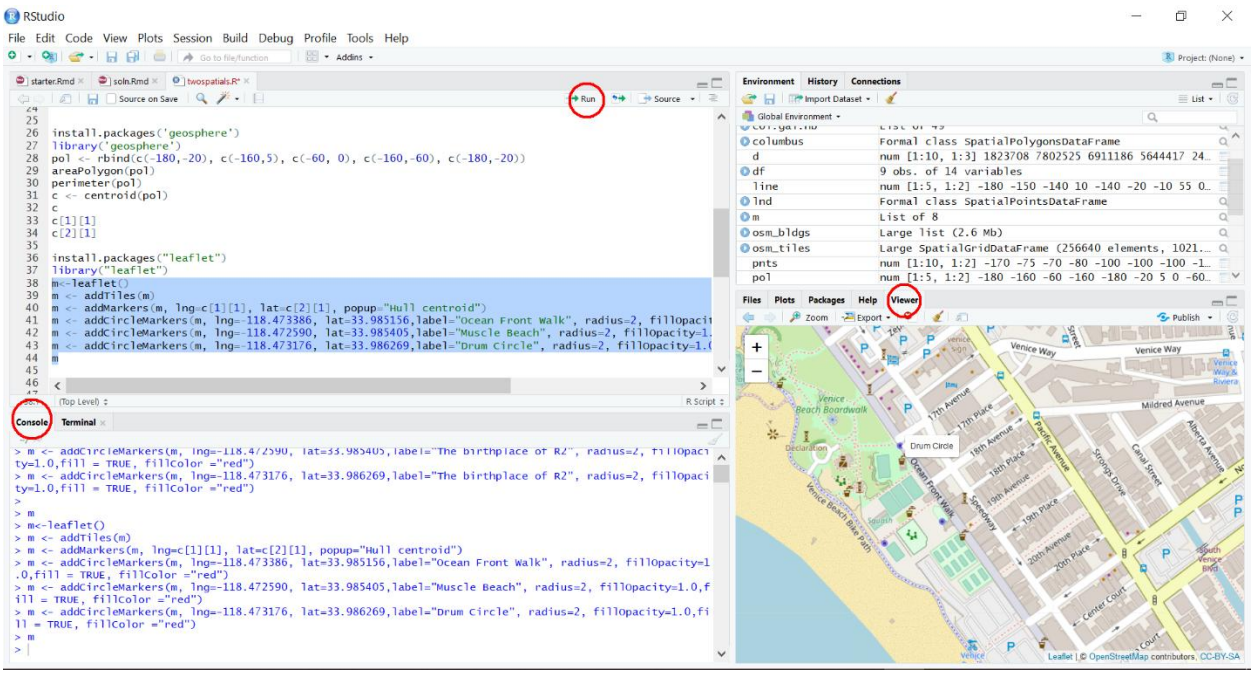
Now select the leaflet code and run it - you'll see a map of SoCal, zoom in to see the three Venice Beach (VB) locations I put in. Pass the mouse pointer over the locations... If you zoom (way) out, you'll see the blue marker from the addMarkers() call, way deep in the ocean to the southwest of VB - click on it.

REPLACE the long,lat pairs in the geosphere section, and the circle marker locations in the leaflet section, with YOUR convex hull USC coords :)

Run the centroid code again, to get the centroid of YOUR hull. Next, run the leaflet code - you should see your convex hull coords (circles), AND their centroid (darkblue flag icon). Cool!

Q4 (0.5 point). Take a screenshot of the entire RStudio IDE that shows the code, console output and the map (sufficiently zoomed in) with the hull and centroid markers visible, for submission.

Here is a screenshot of RStudio, with the editor, console, viewer (tops circled in red):



What to submit: a single .zip, named HW4_<yourname>.zip, with:

• screenshots, named Q1.{jpg,png} etc

• a single README.txt file, with answers for the questions (ie. regression equations)

It's highly worth knowing how to use such tools for analysis, in addition to only knowing how to do so using Python or R code - **the interface-driven tools are just as powerful**, because they encapsulate, with point-and-click UI, a variety of data-mining algorithms/code - resulting in a product that (even) non-programmers, eg. business analysts, managers etc. can use.

That said, R is a wonderful language for data analysis and plotting - if you have never used it before, you can start now, now that you've gotten your feet wet :) Here is a little reference card (not mine) to tell you more. Fun fact: in R, the assignment operator can be flipped! In other words, a <- 5 and 5 -> a are both legal syntax [wow]. Also, what you did above was to calculate your convex hull's area, perimeter and centroid - rather minimal analysis. Much more spatial data analysis/mining is possible using R, and this gives you a taste of it. And there's a little mo'R'e, Me Hea'R'ties - YaRrr/rrrr!!