HW4: Data Mining, tool-based

Total points: 5

Summary: In this homework, you are going to use three UI-based tools (no coding!), to carry out data mining: **WEKA, KNIME, RapidMiner.** There are 6 questions you need to answer.

Description

WEKA

Start by downloading WEKA, from https://www.cs.waikato.ac.nz/ml/weka (https://www.cs.waikato.ac.nz/ml/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.

Take a few hours to go through WEKA's tutorials: https://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/ (https://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/). You can also look at YouTube videos to come up to speed. It is just a matter of getting familiar with the UI, and with the overall workflow (read in data → possibly cleanup data → possibly do EDA → do analysis → possibly export results).

Here (data/housing.arff) is a famous (in the ML/DM community) dataset called the 'Boston Housing Dataset'. (http://tunedit.org/repo/UCI/numeric/housing.arff) 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 (https://www.zillow.com/), TopHap (https://www.tophap.com/), 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 (https://www.cs.waikato.ac.nz/ml/weka/arff.html)], which resembles, but is more descriptive than, CSV.

Q1 (0.5 point). 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.

Q2 (1 point). Create a 'MultilayerPerceptron' **neural network** that learns the data. You can see that you affect the root mean squared error ('RMSE') by setting different values for the learning rate (try to keep this between 0.1 and 0.3), and momentum (again, 0.1 to 0.3). What is the lowest RMSE you are able to achieve? Eg. an RMSE of 5.0 would mean that the MEDV predictions for our 506 rows were off by 5.0 units on the average (the actual values max out at 50, so this represents 10% error on the average). Again, include (two) screenshots that show the NN and the RMSE.

KNIME

Here (data/shells.arff) 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 (https://www.google.com/search?q=abalone+shells&num=100&rlz=1C1CHBF_enUS723US723&source=lnms&tbm=isch&sa=X&ved=0ahUKEwihrcyRq8zXAhWJCpAKHXDdDDUQ_AUICygC&biw=
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 (https://www.knime.com/) ("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).

Q3 (1 point). 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.

Q4 (1 point). Set up a **'Decision Tree Learner'** predictor, where 'sex' is the predicted variable. Note - think "simple" - no need to partition the data into training and test data, etc! Provide a snapshot (.jpg or .png) of the *entire* decision tree [OK if the nodes are too zoomed out and are therefore unreadable] - hint: look at the *right* side of the split-pane window.

RapidMiner

Download RapidMiner Studio (https://rapidminer.com/products/studio/), and play with it for a bit - it is also dataflow-based, just like KNIME.

Q5 (1 point). 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 (https://www.youtube.com/watch?v=tQ7oDnQXhmQ)). Create 6 **clusters** (with all the 4 attrs together, ie. you'd be clustering a 4D dataset) out of the 4177 pieces of data (use a kMeans 'Clustering' node). Question: how many data points are in each cluster? Include a screenshot.

Q6 (0.5 point). Next, 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.

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 (eg. regression equations)

It's highly worth knowing how to use such tools for analysis, as opposed 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.

Please upload your (.zip) submission on to D2L as usual - this (https://piazza.com/class/k53tdrrr6sp62z?cid=1030) helpul note [from Michael and Rittwik] provides details.

ENJOY!!
