Due on Feb 7 2017, beginning of class (Online students can email me a PDF/MS Word file, by that time).

Note: We can measure *accuracy-rate* or *error rate*. To convert between them we just subtract from 1.0. You can use either one in this report, just be clear and consistent.

1. Review the Decision\_Tree\_Matlab.pptx briefing.
2. Visit the UCI Data Archive [b]. Grab the leaf dataset [c] again.
3. Using two-fold cross-validation, compute the **nearest neighbor** error-rate. Report only the best of three possibilities {no normalization, zero-one normalization, z- normalization} (you can reuse code from the last assignment), but tell me which you used.
4. Using two-fold cross-validation, compute the **Decision Tree** error-rate. Here normalization should make no difference (do you see why?). Report error-rate number averaged over 30 runs (and report the standard deviation). Do not use the build in cross-validation tools that Matlab has, just adapt my code, put it in a loop that runs 30 times….
5. Plot one tree. Take the time to type in the feature labels, like this “ 'PredictorNames',{'Aspect-Ratio'…”, so the tree is self contained.
6. Plot a bar chart showing the feature’s importance.
7. Revisit the **nearest neighbor** classifier, suppose you delete a few features that the **Decision Tree** finds to be unimportant, and you recompute the **nearest neighbor** error-rate. Does the error-rate improve, stay the same, get worse?

Hand in a two to three page report, with tables, and any figures your think might be useful for me to see (if any).

Figures and tables need to have meaningful captions. The text should have a self-contained narrative, explain what you did, to a person other than me.

[a] http://www.cs.ucr.edu/~eamonn/time\_series\_data/

[b] https://archive.ics.uci.edu/ml/datasets.html

[c] https://archive.ics.uci.edu/ml/machine-learning-databases/00288/