For this assignment, we are given a sample dataset of the weather and traffic conditions for 14 days and the resulting decision of whether to drive to work or not. We are interested in determining if we can create a decision tree to follow to decide whether the person will choose to drive or not. To begin with, we must first introduce our decision criterion for this assignment, which is entropy. Much like in physics, this term refers to the "instability" or "chaos" that occurs within a system and varies from 0 to 1, only here it specifically refers to the

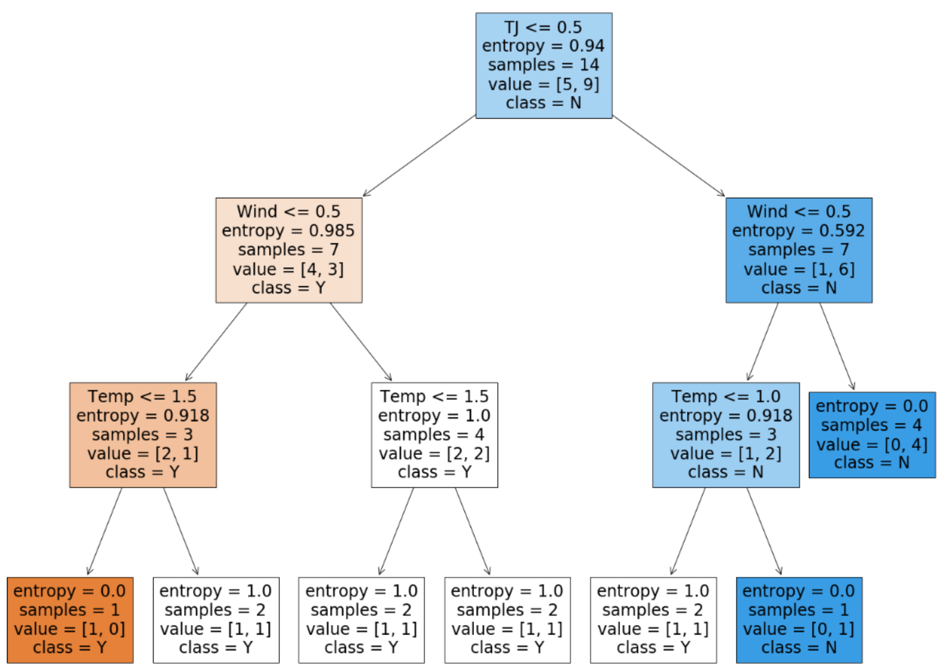
We calculate the information gain from the entropy, which is defined as: Where P_i is the ratio of class C_i in the set of examples S. Thus, $H(S) = 0$ indicates all examples are of the same class, and $H(S) = 1$ indicates the examples are evenly distributed amongst all i classes. Information gain is calculated from this total entropy definition, and as its name implies, it represents the final information content that is given by a specific input feature X_i and is mathematically given by: Where $\frac{|S_v|}{|S|}$ is the ratio of the data with attribute v in S. More simply, the gain can be said to be found individually for each class in an attribute, and then multiplied by the amount of occurrences of that class. The gain for each class is then subtracted from the entropy of S to get a final value.

$$H(S) = \sum_i^N -P_i * \log_2(P_i)$$

$$G(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

Since this value depends on $H(S)$, we must calculate that first. In the Jupyter notebook, this is found to be 0.9403. Next, for the traffic jam attribute, we find the gain of both classes: $TJ_{long} = 0.4926$, $TJ_{long} = 0.2958$. When multiplied by the occurrences of each class in S (in this case, both 7) and then subtracted from $H(S)$, we arrive at: $G(S,TJ) = 0.1519$.
The gains of temperature and wind are found in the same fashion, and were calculated to be: $G(S,W) = 0.0481, \quad G(S,T) = 0.0292$.

The root node of the decision tree is determined by the attribute with the highest information gain. Therefore, we choose traffic jam as the root. The full decision tree is implemented in the corresponding Jupyter Notebook and reproduced here to the right:



The nodes will not be all pure in this tree for two reasons. Firstly, since the splitting criterion is entropy, when the entropy of a node is null (zero), there is nothing left for the model to do because all the examples belong to the same class. These are represented above as the dark orange and dark blue boxes. Secondly, the function that creates this tree has a built in minimum threshold for when to stop splitting (designed to prevent overfitting) and therefore cannot split the white nodes at the bottom of the tree above.

Now, we have the process of fuzzification. This is motivated by the nature of our input variables not being strictly categorical: temperature, wind speed, and the size of a traffic jam are all continuous variables and thus it might be more appropriate to designate them as fuzzy variables. This implies that what constitutes a "cold day" is not as simple as saying it is any day below 32^oF, since 33 or 34 degrees is also still quite cold. Therefore, we utilize membership functions that assign values to classes progressively instead of a vertical jump. These functions map the input variable from 0 to 1, not technically a probability (as there is no "chance" going on) but instead a "suitability" measure.

For our problem, we assign membership functions to each class of each variable in our dataset, which take on the form of piecewise functions:

Attribute *Temperature:*

$$\mu_c(x) = \begin{cases} 1 & x < 0 \\ 1 - x/15 & 0 <= x <= 15 \\ 0 & x > 15 \end{cases}$$

$$\mu_m(x) = \begin{cases} 0 & x < 5 \\ x/15 - 1/3 & 5 <= x < 20 \\ 1 & 20 <= x < 30 \\ -x/5 + 7 & 30 <= x < 35 \\ 0 & x > 35 \end{cases}$$

$$\mu_h(x) = \begin{cases} 0 & x < 25 \\ x/10 - 2.5 & 25 <= x <= 35 \\ 1 & x > 35 \end{cases}$$

Attribute *Wind:*

$$\mu_w(\mathrm{x}) = \begin{cases} 1 & \mathrm{x} < 3 \\ 2.5 - x/2 & 3 <= \mathrm{x} <= 5 \\ 0 & \mathrm{x} > 5 \end{cases}$$

$$\mu_{st}(\mathrm{x}) = \begin{cases} 0 & x < 3 \\ x/5 - 0.6 & 3 <= x <= 8 \\ 1 & x > 8 \end{cases}$$

Attribute *Traffic-Jam:*

$$\mu_{sh}(\mathrm{x}) = \begin{cases} 1 & x < 3 \\ 1.5 - x/6 & 3 <= x <= 9 \\ 0 & \mathrm{x} > 9 \end{cases}$$

$$\mu_{l}(\mathrm{x}) = \begin{cases} 0 & x < 5 \\ x/10 - 0.5 & 5 <= x <= 15 \\ 1 & \mathrm{x} > 15 \end{cases}$$

These are implanted in the corresponding notebook. Once again, we choose our root node for our decision tree based on the information gain value. Since we calculate the gains as below, we therefore conclude that Traffic Jam should serve as our root node in the fuzzy decision tree, since it presents us with the highest information gain.

$$G_f(S, W) = 0.0425, \qquad G(S, T) = 0.0671 \qquad G(S, TJ) = 0.1158$$

The paper by Banakar et al. (2016) discusses the attempt to perform classification learning on dried figs, and the authors utilized the same fuzzy decision tree logic as above to achieve the best classifier, with an accuracy of 91.74%. The fruits can be grouped into a number of classes and there were found to be 5 key features with sufficient impact in the precursory analysis.

The application of fuzzy logic is introduced due to the fact some of these predictor variables are not categorical and therefore attributes attain membership to a particular class through a function, mapping onto [0,1]. Unlike in our example, where we assumed the data to be crisp, in this paper (and in many practical scenarios) we do not have that strict requirement and thus one record can belong to more than one class. For example, if we had adopted this approach for the driving example, a traffic jam that was 4km long would have been classified as both long and short, since it lies in the metaphorical “grey area” and could reasonably be defined as either. The overall fuzzy decision tree model in the fig quality classification outperformed its competitors not only for classification accuracy but also the other metrics of sensitivity and specificity.

References

Janikow, C. (2010). *Fuzzy Decision Trees: Issues and Methods*. University of Missouri – St. Louis Department of Mathematics and Computer Science. http://www.cs.umsl.edu/~janikow/fid/papers/fid.ieeesmc.pdf

Liang, G. (2005, August). *A comparative study of three Decision Tree algorithms: ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3*. Erasmus University Rotterdam. https://d1rkab7tlqy5f1.cloudfront.net/TBM/Over%20faculteit/Afdelingen/Engineering%20Systems%20and%20Services/People/Professors%20emeriti/Jan%20van%20den%20Berg/BachelorThesis/candy-bachelor.pdf

T, S. (2019, July 7). *Entropy: How Decision Trees Make Decisions - Towards Data Science*. Medium. https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Thapliyal, T. (2020, December 7). *Decision Trees in Python - Step-By-Step Implementation*. AskPython. https://www.askpython.com/python/examples/decision-trees

Understanding the decision tree structure — scikit-learn 0.24.2 documentation. (2016). Scikit-Learn. https://scikit-learn.org/stable/auto_examples/tree/plot_unveil_tree_structure.html#sphx-glr-auto-examples-tree-plot-unveil-tree-structure-py