Week 5: Decision Trees – Technical Report

Seve Martinez
Grand Canyon University

Abstract

This week we review the decision tree model type and work to implement one using both manual efforts and some machine learning Libraries. Using a dataset of three features and one label, a tree was produced with the presented data. However, given the ambiguity of some observations, full purity was not realized.

Keywords: decision tree, sklearn, machine learning

Table of Contents

Abstract....2
Heading 1....4
Heading 2....4
Heading 3....4
Heading 4....4
Heading 5....4
Reference list....7
Appendix A....8
Appendix B....10

List of Figures

Figure 1. Example figure body text....6
Figure A1. Example figure appendix....9
Figure A2. Example figure appendix....9
Figure B1. Example figure appendix....10

List of Tables

Table 1 Example table body text....6
Table A1 Example table appendix....9

Methods

The decision tree classifier is a great model for both interpretability and as an input for more complex models thanks to its ability to determine feature importances. It can be used both as a regression model and a classifier. Decision trees leverage information gain and entropy in a recursive manner to build out nodes of increasing purity until either all leaves are pure, or there is no other way to discern the labels.

Trees have two main drawbacks: High variability and overfitting of the training set. It is rather easy to build a model with 100% training accuracy that generalizes poorly. However, hyperparameters in the sklearn DecisionTree class enable us to prevent that using the max_depth parameter.

Entropy

Entropy is the measure of uncertainty in data (Kwiatkowski, 2018). Entropy is a number between zero and one and has a arc-shaped graph. This means that in a two-class dataset, if the classes are perfectly represented as 50/50, then we have max entropy of 1. As the data divide becomes more uneven to one class or the other, the entropy will reduce. In other words, the more samples we have of one class, the less uncertainty, or entropy we will have.

Mathematically entropy is as follows:

$$Entropy = -\sum_{j} p_j \log(p_j) \quad \text{(Equation 1)}$$

where p_j is the probability of a class *j*. The logarithm is there for mathematical convenience thanks to it's additive property.

Information Gain

Information gain is the measure of purity within the data. This is the key to decision trees and the formula used recursively to build out the tree. The greater the increase in the information gain, the lower the entropy (Loaiza, 2020).

Information gain is calculated as the prior entropy subtracted by the sum of the child entropies. This is how we know how well the split handled reduction in uncertainty.

$$\text{Information Gain (T,X)} = \text{Entropy(T)} - \sum_{children} \frac{c_1}{T} Entropy(c_1) \quad \text{(Equation 2)}$$

Entropy(T) is the uncertainty of the information fore the split. Next we multiply the probability each child class (split) by the entropy of that class. Then we subtract that sum from the original entropy. This is information gain. The freature with the greatest information gain will be used for the split.

Dataset

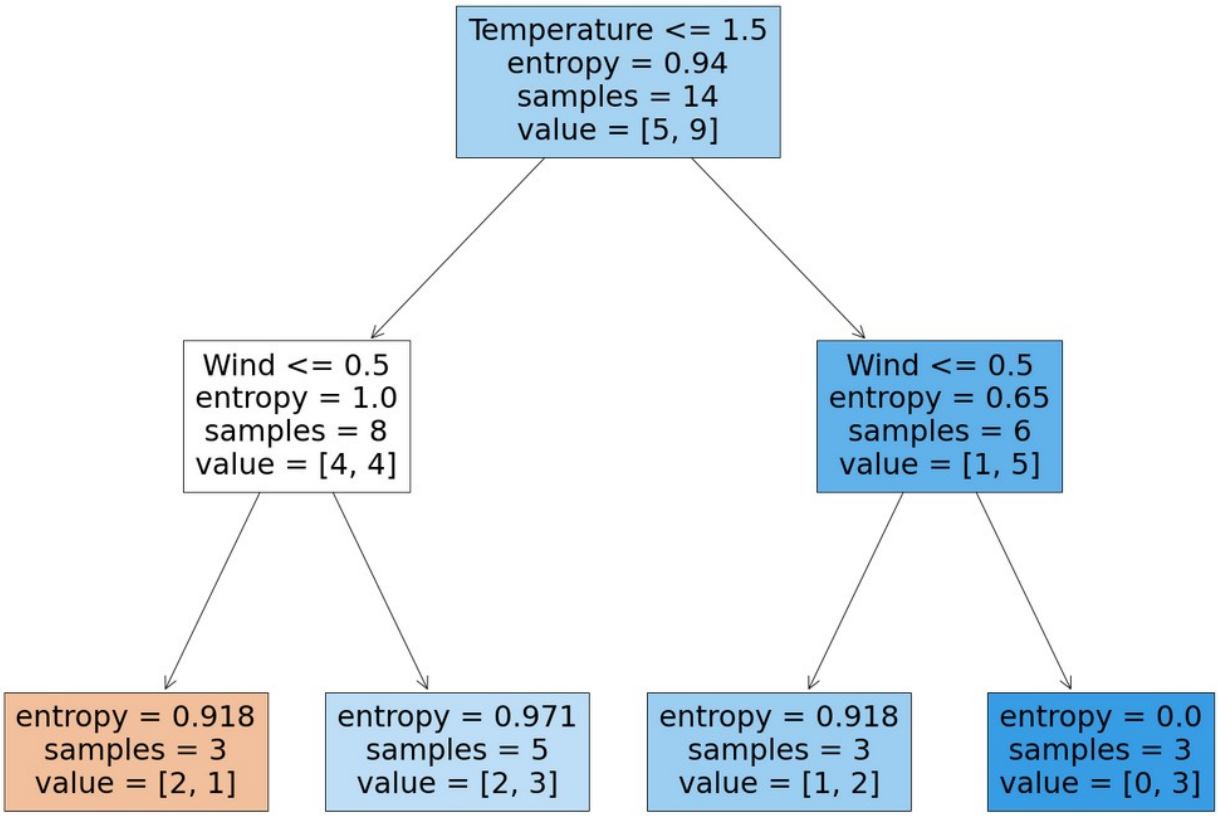
The dataset is a simple set of 14 observations about whether or not a person will drive a car based on three features. The three features are Temperature, Wind, and Traffic-jam. Their examples are as follows:

- Temperature
  - Hot
  - Mild
  - Cool
- Wind
  - Strong
  - Weak
- Traffic-jam
  - Long
  - Short
- Labels
  - yes
  - no

There are some rows that are identical which have different labels. This will create an issue when building the tree regarding purity. Some leaves will not be fully pure. The only way to resolve this would be to have more features that make the rows unique.

The Model

The initial model had a max_depth of 2 to show the first split. The feature with the highest information gain is Wind with 0.30. The other two were 0.09 for Temperature and 0.28 for Traffic-jam. This was used for the initial split. The sklearn model class also came up with the same conclusion as shown here:

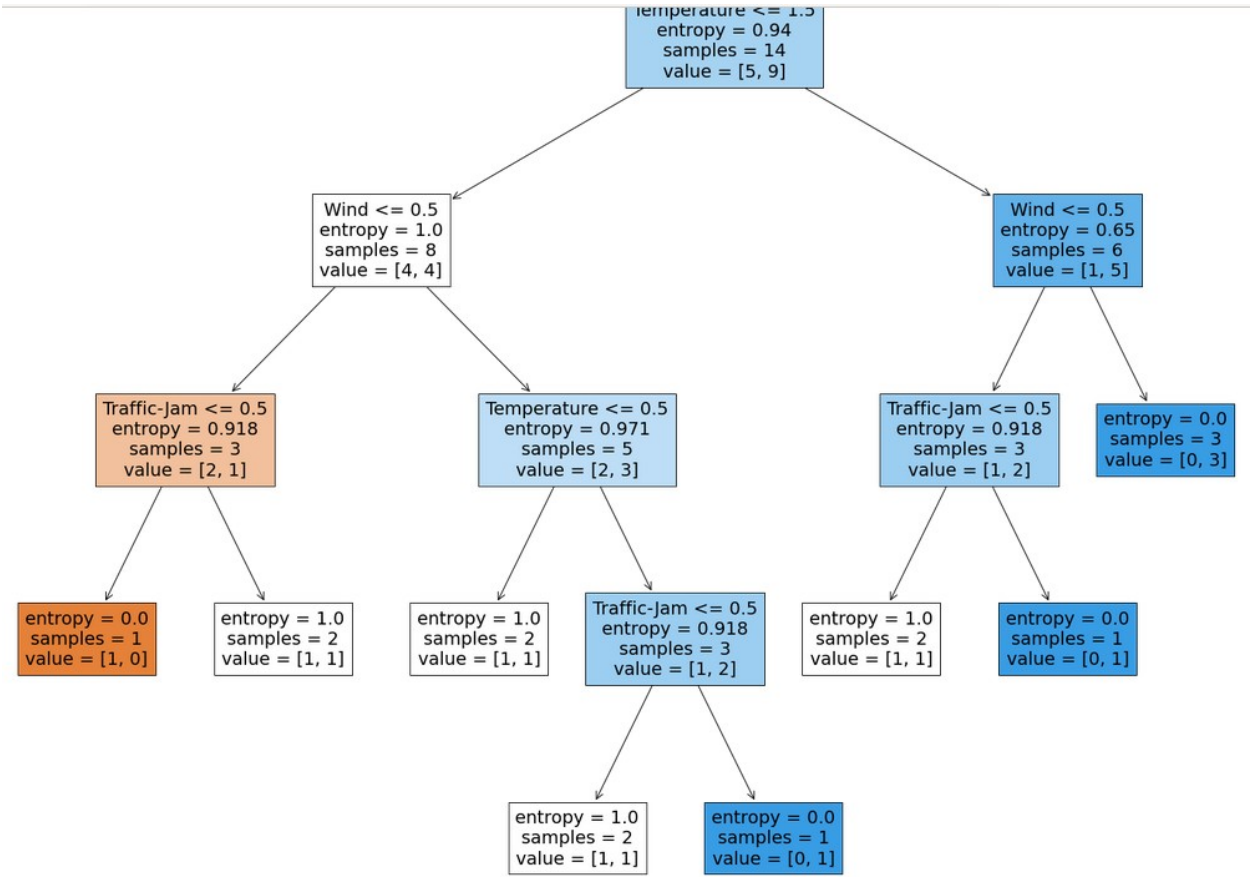


While the model shows “Temperature” at the top, it should be noted that the full dataset is in the root node. So there was no split based on temperature.

The lowest boxes are the leaves. With max_depth of 2, the model only splits based on the initial attribute and then builds the leaves. However, it’s obvious that there is not enough information to discern the classes properly. Only one leaf, the far right, ended up with 0 entropy and only a single class.

The Full Tree

After this example, the tree was built to run with max_depth = None. This will allow the model to do its best to leverage all features to build the best model possible. It’s result is as follows:



Leveraging traffic-jam and temperature, I was able to two additional leaves with full purity. However, note that most of the leaves did not end up in pure leaves and several have max entropy. This is because there was no way to identify differences between two identical observations with differing labels.

Conclusion

The final tree had an accuracy of 71.4% which is alright given the dataset. Had here been more features, surely the accuracy could be increased. However, the model will largely generalize well to a dataset of the same dimensions.

References

Kiwatkowski, S. (2020). Entropy is a Measure of Uncertainty. Towards Data Science. https://towardsdatascience.com/entropy-is-a-measure-of-uncertainty-e2c000301c2c

Loaiza, S. (2020). Entropy and Information Gain. Towards Data Science. https://towardsdatascience.com/entropy-and-information-gain-b738ca8abd2a

Gopal, M. Applied Machine Learning. McGraw Hill. New York.