# **Decision Tree-Entropy and Information Gain**

Krishna Kumar Veeraputhiran

Grand Canyon University

DSC 540 – O500 – Machine Learning for Data Science

Dr. Aiman Darwiche

September 8,2021

# **Decision Tree-Entropy and Information Gain**

In this technical report we will discuss on the dataset that has been used for this weeks assignment and we will see on how the Information gain has been calculated for each attribute using the Entropy. We will also discuss on the decision tree that has been built for the dataset.

**Dataset**

The dataset used for this week’s assignment has 14 entries with each entry depicting a day along with the temperature, wind speed and traffic (input parameters) for the given day along with whether an individual will decide to drive his car or not (output parameter) based on the input parameters. In order for us to build a decision tree based on the data we need to factorize the data of the input fields and use them in the algorithm. From the 14 day observation, we can see that the for 9 days the individual has decided to drive the car whereas for the remaining 5 days he/she has not decided to drive the car.

**Entropy**

The decision tree uses various criteria to select a variable and do a split at a given node and one such criteria is using the entropy. Entropy in data science is a way to measure how mixed a variable is. Entropy is specifically used to measure the disorder in a given column in a dataset. Using the dataset, we will calculate the entry for the target variable and the input variables. The generic entropy for any given variable can be given as,

-∑ c i=1 P(xi) logbP(xi)

We can calculate the entropy for the target variable based on the probabilities. The probability of driving the car is 9/14 which equates to 0.64 and the probability of not driving is 5/14 and can be written as 0.36. Entropy of decision to drive is

-(Probability to drive \* log2(Probability to drive) + Probability not to drive \* log2(Probability not to drive))

-(0.64 \* log2(0.64) + 0.36 \* log2(0.36))

-(0.64 \* (-0.6439) + 0.36 \* (-1.474))

-(-0.412 – 0.531)

0.943

*The entropy of the dataset based on the target value is 0.943*

**Information Gain**

The goal of decision tree is to lower the entropy on both side of the tree. Ultimately the overall entropy of the decision tree need to be lowered with each classification. Information Gain, or IG for short, measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. It can be given as,

**Gain(D,xi) = Entropy (D) – Entropy(D,xi)**

For attribute temperature, we see that they have three classifications, hot, mild and cool. We have 4 days of hot days out of which 2 days the individual has used the car and 2 days without the car. Similarly there are 6 mild days and 4 days have driven and 2 days have not driven the car. For cool days, there are a total of 4 days out of which 3 days the car has been used and one it has not been used. The information gain for temperature can be calculated as,

Info(hot) = -(2/4 \* log2(2/4) + 2/4 \* log2(2/4) ) = 1.0

Info(mild) = -(4/6 \* log2(4/6) + 2/6 \* log2(2/6) ) = -(0.667 \*(-0.584) + 0.333\*(-1.586)) = 0.917

Info (cool) = -(3/4 \* log2(3/4) + 1/4 \* log2(1/4)) = -(0.75\*(-0.415) + 0.25\*(-2)) = 0.81

Info(D, temperature) = 4/14 \* 1 + 6/14\*0.917 + 4/14\*0.81 = 0.286 + 0.393 +0.231=0.91

***The information gain of temperature is 0.943 -0.91 = 0.033***

For wind, there are two classifications namely weak and strong. 8 days have weak winds with 6 days driven and 2 days not driven. Similarly there are 6 days of strong winds with 3 days driven and 3 days not driven.

Info (weak) = -(6/8\* log2(6/8) + 2/8 \* log2(2/8)) = -(0.75\*(-0.415) + 0.25\*(-2)) = 0.81

Info(Strong) = -(3/6\* log2(3/6) + 3/6 \* log2(3/6)) = -(0.5\*(-1.0) + 0.5\*(-1.0)) = 1.0

Info(D,wind) = 8/14\*0.81 + 6/14\*1 = 0.463 + 0.429 = 0.892

***The information gain of wind is 0.943-0.892 = 0.0851***

For traffic, we have again two classifications namely long and short. 7 days have longer traffic jams out of which 3 days the car has been driven and for 4 days the car has not been driven. Similarly there are 7 dyas of shorter traffic jam with 6 days of car being driven and one day not being driven.

Info(long) = -(3/7\* log2(3/7) + 4/7 \* log2(4/7))= -(0.429\*(-1.222) + 0.571\*(-0.807)) =0.985

Info(short) = -(6/7\* log2(6/7) + 1/7 \* log2(1/7))= -(0.857\*(-0.222) + 0.143\*(-2.807)) = 0.591

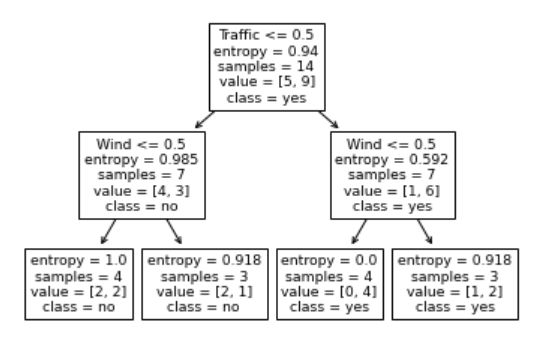
Info(D,traffic) = 7/14\*0.985 + 7/14\*0.591 = (0.5\*0.985 + 0.5\*0.591) = 0.788

***The information gain of traffic jam is 0.943-0.788 = 0.155***

The attribute with the highest Information gain is selected as the root node for the split. In our case the traffic attribute has the highest information gain (0.155). *Hence, the traffic jam variable is selected as the root node.*

**Partial Decision Tree**

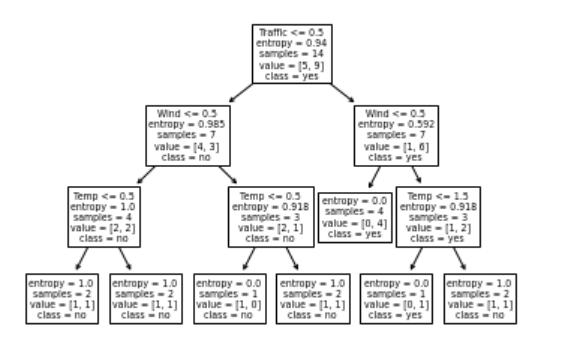
Below plot shows the partial decision tree plotted for the dataset.



We can see that the tree is not completely grown and is pre-pruned. In python, a decision tree can be built using the DecisionTreeClassifier function available in the sklearn library and tree package. The parameter that is used to prune the tree here is the max\_depth and in our case it has been set at 2. Setting the max\_depth will help in regularizing the tree and prevent the tree from overfitting.

**Fully Grown Decision Tree**

Below is the picture of a fully-grown Decision tree. Usually when parameters like max\_depth or max\_leaf\_nodes the tree is regularized. If these parameters are not set then the tree continues growing till all the leaf nodes are pure. This can also lead to overfitting the decision tree, meaning the model completely fits the training data and fails to generalize the test data, leading to increase test data error.



We can see that the leaf node is fitted for all of the training data (14 records) and hence the training data is not generalized. Also when we allow a decision tree to grow it tries to achieve maximum purity at the leaf node. A node is 100% pure when the data in a single node belongs only one class. We do see that in the above decision tree there are 4 leaf nodes where all the samples do not belong to the same class. It is not possible in our case to achieve 100% purity as there are some inconsistency within our data. For example consider day 1 and day 3, the conditions are same for both these days, however the outcome is different. This is the reason why we cannot achieve 100% purity in our node split.

# **References**

Gopal, M. (2019). *Applied machine learning*. McGraw-Hill Education.

Lutes, J. (2020, November 15). *Entropy and information gain in decision trees*. Medium. https://towardsdatascience.com/entropy-and-information-gain-in-decision-trees-c7db67a3a293.