Information Gain and Entropy.

* Describe what **Information Gain** is and how to compute it.
* Describe what **Entropy** is and how to compute it.
* Explain why **Information Gain** is useful in building **Decision Trees**.

**Information Gain & Entropy 🡪** Are Information Theory terms.

* **Information Gain** 🡪 Is the evaluation function we use in building Decision Trees.
* **Entropy** 🡪 It is a very important underlying concept.

|  |  |
| --- | --- |
| Information Gain && Entropy | |
| GOAL: We want to find the **threshold** that gives us the **best** split.  i.e. Find the threshold over a particular Feature’s values.  (Numeric Features) | |
| What does it mean for a **split** to be ***best***?  ==  What does it mean for the data to be **split** the ***best***?  In order to address this question and figure out what threshold will give us the best split, we need the following concepts.  This is not the only way to find one of these splits. | |
|  | **Entropy** is something that measures Uncertainty and Chaos. |

* **Entropy** is something that measures Uncertainty and Chaos.
  + A dataset with an equal ratio of Class Labels is Completely Uncertain 🡪 It has **High** **Entropy**.
    - (**High** **Entropy** = High Uncertainty).
    - Because there is NO Majority Label 🡪 There is nothing about the data that tells us what we should use to predict the Class.
    - This is a Chaotic dataset.
    - Ex. A dataset with 50/50 Split between 2 Classes has **High** **Entropy**.
  + A dataset consisting of a single Class Label is Completely Certain 🡪 It has **Low** **Entropy**.
    - Ex. If we have 100 observations and they all have the same Label.
  + Goal: Come up with a **Low** **Entropy** **Split**.
* Information Gain 🡪 Gives us a measure of the change in **Entropy** before and after splitting.
  + It is an Information Theoretic measure.
  + 🡪 Considers the difference in **Entropy** of data before and after splitting.
    - It helps us answer: “Is it helpful to Split on this Feature (and this particular Threshold)?”
    - This is what we really want to know as we are trying to choose which Feature to use and which Threshold to use, for a particular Internal Node.
      * 🡪 We want to pick the Feature and Threshold that is going to help us narrow down our dataset, in such a way that we can zero-in on a kind of High Confidence, Low Entropy decision to Classify a New observation.
  + We want to look at what the **Entropy** is of our entire dataset.
  + Then, we look at the **Entropy** of the subsets created by splitting.
  + What we want to know is 🡪 Do we get 2 subsets that give us better **Entropy** overall if we Split it with this Threshold, than if we kept the original Split?
  + Information Gain 🡪 Bigger is better.
    - This is one of the reasons why we want to optimize our evaluation in our pseudo code for the Decision Tree Induction Phase.

|  |  |  |  |
| --- | --- | --- | --- |
| Information Gain. | | | |
| In our Decision Tree, we are looking at 1 node, and we are picking a Feature for that internal node. And we want to know what Threshold to use for that Feature. | | | |
|  | For a particular Threshold, we are going to Split the data, giving us S1 and S2.  It takes 2 data Splits.  The top half and the bottom half.  We want the Information Gain for these 2 subsets. | | |
|  | We first calculate the **Entropy** of our unsplit dataset.  We take the 2 subsets and we glue them back together.  U (union) takes 2 Sets and joins them together. Like addition. | | |
|  | We are going to compute the same formula, but for our 2 different subsets. | | |
| p(S1) | Is the probability of our first data Split.  It is just the size of that Split, divided by the size of our Unsplit data.  🡪 9/22 | |
| H(S1) | Where we compute the **Entropy** for S1. | |
|  |  | |  |
| p(S2) | | The probability of S2 is:  🡪 13/22 |
|  | | Where we compute the **Entropy** for S2. |
|  |  | | |
|  |  | | |
|  |  | | |

|  |  |
| --- | --- |
| Entropy Formula | |
| For a particular subset of observations | |
|  |  |
|  | “For each Class Label x in S, we will compute the following”  ∑ 🡪 Looping over all of our Labels. |
|  | Our whole dataset has 3 different Classes.  X 🡪 We are going to figure out what Classes this given Set of data has.    It is going to keep track of the Set of distinct Class Labels we have (3).  For the case of S1, we only have 1 Class : Setosa.  Thus, X = 1. |
| x ∈ X x is an element of X.  “For each Class x in our Set of Classes X”  We are going to compute: |
| d | (The negated probability of x) \* (the log base 2 of p of x)  We are computing p(x) in 2 places (d , g). |
| e | The reason for this negative is that log is going to give us a negative number.  Because g is going to be between 0 and 1.  If you log anything that is 1 or under, you are going to get some kind of negative value. |
|  | So we compute these 2 p(x)  All p(x) is:  The proportion of observations that have that particular Label. |
|  |  |
|  |  |