**Decision Trees Classification**

Consider a decision (to play or not) which depends on three variables (outlook, humidity, wind).

Table

Description automatically generated

It seems in this case that Play = f(outlook, humidity, wind), i.e., it is a function of those three variables. There are only 3×2×2 = 12 possible variable combinations, points in the Domain. And each gets mapped to a single value of Play. I suppose in some other scenarios, the decision output would only be probablistic, not determinant, i.e., the same domain points gets mapped to a certain percentage of Yes’s and No’s.

**Information/Entropy of Distribution, Predicted Outcome, and Information Gain**

First we’ll consider the entropy of the entire set of outcomes. For a probability distribution of a set of outcomes Yi, the entropy is defined as (evidently using base N log when there are N outcomes?):



This is kind of a measure of how spread out the probability distribution is. In that sense, it’s kind of like variance. Might note a completely determinant, and completely random distribution would give us, respectively:



So a completely determinant distribution gives us 0 entropy, as makes sense. And a completely random distribution gives us the ‘max’ entropy, 1. In our situation, the outcomes are those of Play, and there are 8 Yes and 6 No outcomes. So the P(play = yes) = 8/14, and P(play = no) = 6/14. And the entropy associated with this is:



So play is a pretty random random variable. For Decision Tree purposes, the predicted outcome, or fit, of our distribution would be the most probable value:



and so the predicted outcome above would be f = Yes. While the outcome of Play is dependent on all three variables, some of the variables have a greater determining factor than others in the outcome. To start, we’ll consider the conditional entropy of Y, given one of the values of A. This is:



That is to say, we consider all the rows with value A = Ai. Then out of this reduced set, we calculate the probability distribution and entropy of the Y values. FWIW, we can define a predicted outcome for this reduced set of Y values. That would be:



i.e., the most common of the values Y can assume when A = Ai. But moving on for now….we can further define an average conditional entropy for Y, assuming the variable A. This is:



where Ai enumerates the possible values that variable A may take on, P(Ai) is the associated probability of it taking on those values. The point of calculating this quantity is to ascertain whether knowledge of A improves knowledge of Y, in which case we should find S(Y|A) < S(Y). This will be the case if A is correlated with Y to some degree. For instance, consider the case where A is perfectly oppositely correlated with Y:

|  |  |
| --- | --- |
| **A** | **Y** |
| No | Yes |
| No | Yes |
| Yes | No |
| Yes | No |

Then,



So knowledge of A gives us perfect knowledge of Y. But it is possible that knowledge of A obfuscates knowledge of Y too, in which case S(Y|A) > S(Y). For instance, consider the following table,

|  |  |
| --- | --- |
| **A** | **Y** |
| Yes | Yes |
| No | Yes |
| No | Yes |
| No | Yes |
| Yes | No |
| No | No |

Then,



So we define the information gain of the variable as the difference between the outcome’s entropy, and the outcome’s entropy given the A variable.



As we noted the information gain can be negative. When will IG(A) = 0? This will happen if S(Y|A) = S(Y), which will happen if S(Y|Ai) = S(Y). And *this* would happen if P(Yj|Ai) = P(Yj). And since P(Yj|Ai) = P(Yj∩Ai)/P(Ai), this equality would imply P(Yj∩Ai) = P(Yj)P(Ai), which would mean the events are completely independent. Or in other words, the Y and A variables are completely uncorrelated. And so it makes sense in that case that the information gained by knowledge of A is zero. And the max information gain we can have is IG(A) = S(Y). This will happen if S(Y|Ai) = 0. And *this* will happen if P(Yj|Ai) = δYY´, i.e., if there is only one outcome Yj associated with Ai, and its probability is 1. By the way, we can rewrite the information gain as:



So we can say,



where we define the Mutual Information of two variables,



Again, this evinces IG(A) as a measure of how correlated Y is with A.

**Gini Impurity of a Distribution, Predicted Outcome, and Gini Impurity Loss**

Another way to quantify the spread of outcomes of a variable is the Gini impurity value. This is:



Might note a completely determinant, and completely random distribution would give us, respectively:



In the case where N = 2, the most impure outcome would be ½. In our situation, the outcomes are those of Play, and there are 8 Yes and 6 No outcomes. So the P(play = yes) = 8/14, and P(play = no) = 6/14. And the entropy associated with this is:



For Decision Tree purposes, the predicted outcome, or fit, of our distribution would again be the most probable value:



and so the predicted outcome above would be f = Yes. Like before, we’d be interested in measuring how correlated another column, say, A, is with the outcomes Y. So let’s restrict ourselves to the case A = Ai. Then out of these Y’s we can construct a (relative) probability distribution for them, and (relative) Gini Impurity.



And we can define a predicted outcome for this reduced set of Y values. That would be:



i.e., the most common of the values Y can assume when A = Ai. And if we consider all outcomes of A at once, then we can construct an average Gini Impurity, assuming knowledge of A. This is:



Let’s reprise our two prior examples. Consider the case where A is perfectly oppositely correlated with Y:

|  |  |
| --- | --- |
| **A** | **Y** |
| No | Yes |
| No | Yes |
| Yes | No |
| Yes | No |

Then,



So knowledge of A gives us perfect knowledge of Y as reflected by its impurity value of 0. And we’ll note GI(Y|A) < GI(Y). But it is possible that knowledge of A obfuscates knowledge of Y too, in which case we expect GI(Y|A) > GI(Y). For instance, consider the following table,

|  |  |
| --- | --- |
| **A** | **Y** |
| Yes | Yes |
| No | Yes |
| No | Yes |
| No | Yes |
| Yes | No |
| No | No |

Then,



Well GI(Y|A) < GI(Y) still, but whatever. In principle, we can have GI(Y|A) > GI(Y). And I guess we could define a quantity – the Gini impurity loss, as:



And if GIL(A) > 0, then that means A is correlated with Y. And the larger the better. When is GIL(A) = 0? If events Y and A are completely uncorrelated, then P(Yj|Ai) = P(Yj). And then we have GI(Y|A) = 1 – ΣjP(Yj)2, whatever that is, and so GIL(A) = 0. If on the other hand, P(Yj|Ai) were something like δYY´, we’d get GI(Y|A) = 0, and then GIL(A) = GI(Y), it’s largest possible value. Let’s do an example,

**Log-Loss, Predicted Outcome, and Information Gain**

A third option is Log-Loss. Things work a little differently in this case. Let’s consider a set of events {xi} with outcomes yi, which can take on 0, 1. And then consider a logistic probability function of outcomes f(xi) = 1/(1+e-x\_i) of these outcomes. As in the logistic regression file, we define the log-loss of the probability distribution function to be:



And observe that the more closely f(xo) tracks yi, i.e. the more closely f(xi) = 1 when yi = 1, and f(xi) = 0 when yi = 0, the closer to 0 LLf(Y) will be. But actually, for decision tree purposes, we consider a simpler logistic regression function, f(xi) = p, which is just a constant. And p is the constant which minimizes LLf(Y). This happens to be p = n1/(n0 + n1) where n1 is the number of Yi = 1 occurences, and n0 is the number of Yi = 0 occurences. Can see this via:



So there we are. So given f(xi) = p for all xi, we can say that the log-loss of a set of outcomes, is:



and the probability of a positive outcome, is:



Again, we’re interested in measuring how correlated another column, say, A, is with the outcomes Y. So let’s restrict ourselves to the case A = A1, say. Then out of these Y’s we can construct a (relative) overall Log Loss.



and a relative probability output,



So the value A = A1 would be considered to adhere closely to the outcomes yA1,j if pA1 adheres closely to yA1,j. Since pA1 is just the average of the values of yA1,j, this means that A = A1 will adhere closely to yA1,j if yA1,j adheres closely to its average, which would basically only be if yA1,j is mostly 0’s or mostly 1’s. Finally, we can calculate a log-loss over the entire variable A = {Ai}.



So the variable A will do a good job tracking the outcome Y if LLf(Y|A) is small. As before, we can define an information gain,



that will be correspondingly large when LLf(Y|A) is small. And we’d split the decision tree by the variable, A, that maximizes the information gain. And our leaves would just be outputing probabilities. But we would customarily choose a cutoff pcut = 0.5 to delineate classifications of 1/0, yes/no.

**Example**

So let’s recall our table above,

Table

Description automatically generated

And let’s calculate some entropies, relative entropies, information gains, etc.

**Outlook**

For instance, consider Outlook. Let’s calculate the Play entropy associated with Outlook:



Looking at the table above, the P(Outlooki) probabilities and associated entropies S(Play|Outlooki) are:



Therefore the Play entropy associated with Outlook is:



So the Outlook information gain would be:



Okay. And next,

**Humidity**

Can do similarly for the other variables…let’s do Humidity. The Play entropy associated with Humidity is:



Looking at the table above, the P(Humidityi) probabilities and associated entropies S(Play|Humidityi) are:



So Play entropy associated with Humidity is:



and information gain is:



So there.

**Wind**

And last we’ll do Wind. The Play entropy associated with Wind is:



Looking at the table above, the P(Windi) probabilities and associated entropies S(Play|Windi) are:



So Play entropy associated with Wind is:



and information gain,



So looks like Outlook has the largest information gain.