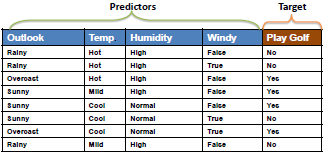
# Quiz 2 Decision Tree Classification and Entropy/Information Gain Calculations

Use the following data set to determine which input feature should be the first to use to divide the data set (and grow the decision tree into two or more branches) based on the Maximum Information Gain Principle. Type the needed calculations and arguments in the space below and rename the file using your name. Discuss the reasons that decision tree is one of the most popular predictive analytics models for decision making. Bonus points would be given to divide the data set (and grow the tree) further.



**Answer**

Decision trees are widely used models for classification and regression tasks. Essentially, they learn a hierarchy of if/else questions, leading to a decision. Decision trees are assigned to the information-based learning algorithms which use different measures of **information gain** for learning.

The main idea of decision trees is to find those descriptive features which contain the most "information" regarding the target feature and then split the dataset along the values of these features such that the target feature values for the resulting subsets of data are as pure as possible. This process of finding the "most informative" feature is done until we accomplish a stopping criterion where we then finally end up in so called **leaf nodes**. The leaf nodes contain the **predictions** we will make for new query instances presented to our trained model.

A decision tree mainly contains of a **root node**, **interior nodes**, and **leaf nodes** which are then connected by **branches**.

**Splitting** is the key to the decision tree models. The information gain is a measure of how good a descriptive feature is suited to split a dataset on. The entropy of a dataset is used to measure the impurity of a dataset and we will use this kind of informativeness measure in our calculations. There are also other types of measures which can be used to calculate the information gain. The most prominent ones are the: Gini Index, Chi-Square, Information gain ratio, Variance.

Descriptive Features of our Dataset –

* Outlook
* Temp
* Humidity
* Windy

Target Value of our Dataset – Play Golf (Yes | No)

Overall Entropy (Purity/Impurity) of our Data set based on Shannon’s definition of Entropy which serves as the basis for Information Gain calculation –



where we say that P(x=k) is the probability, that the target feature takes a specific value k. Hence applying this formula to our example with the two Target Values (Play Gold = Yes | No) we get:

Play Golf = Yes: H(x = Yes) = 0.5 \* log2(0.5) = -0.5 | Play Gold = No: H(x = No) = 0.5 \* log2(0.5) = -0.5

|  |  |
| --- | --- |
| **Target Value** |  |
| Yes | -0.5 |
| No | -0.5 |
| Overall Entropy | 1 |

**H(x) = -((0.5 \* log2(0.5)) + (0.5 \* log2(0.5))) = 1**

**Overall Entropy of the Dataset = 1**

We have now determined the total impurity/purity (≈≈ entropy) of our dataset which equals to approximately 1. Now our task is to find the best feature in terms of information gain i.e. the feature which splits the data most accurately along the target values which we should use to first split our data on (which serves as root node).

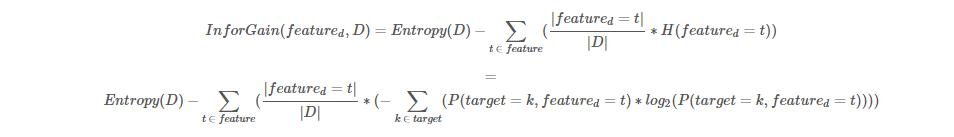
How to choose the best feature which splits the dataset to have lowest Impurity/Entropy?

We use each descriptive feature and split the dataset along the values of these descriptive feature and then calculate the entropy of the dataset once we have split the data along the feature values. This gives us the remaining entropy after we have split the dataset along the feature values. Next, we subtract this value from the originally calculated entropy of the dataset to see how much this feature splitting reduces the original entropy.

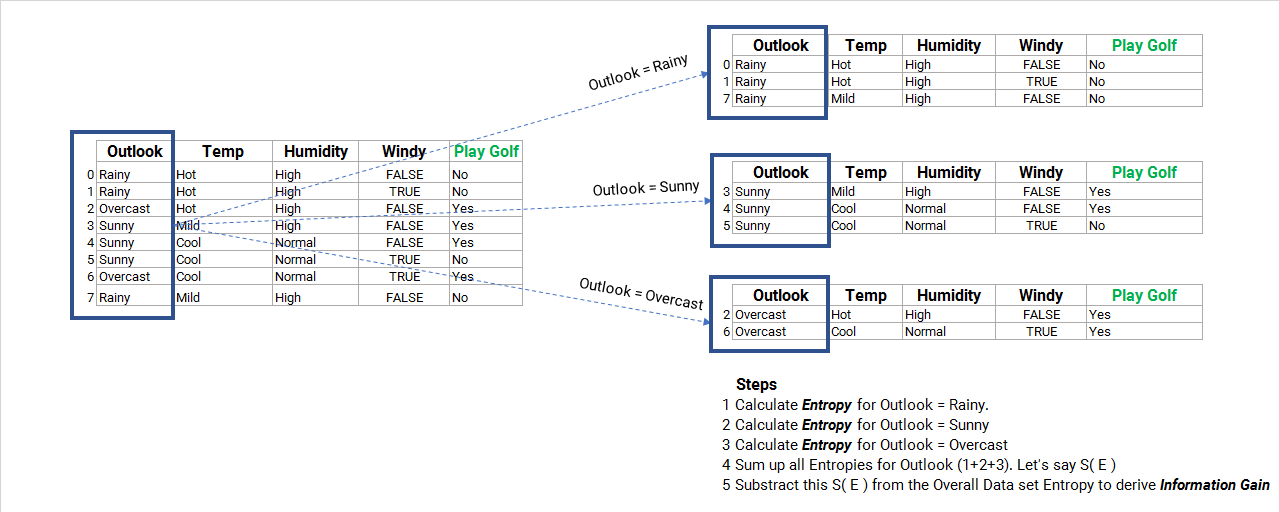
The information gain of a feature is calculated with:



So, the only thing we have to do is to split the dataset along the values of each feature and then treat these sub sets as if they were our "original" dataset in terms of entropy calculation. The formula for the Information Gain calculation per feature is:



Calculate the “Information Gain” for each Descriptive Feature. Here we start with “Outlook” and proceed to Temp, Humidity and Windy.



H(**Outlook**) = H(x = Rainy) + H(x = Sunny) + H(x = Overcast)

H(**Outlook**) = (3/8 \* -((3/3 \* log2(3/3))) + (3/8 \* -((2/3 \* log2(2/3) + (1/3 \* log2(1/3))) + (2/8 \* -(2/2 \* log2(2/2))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rainy | Sunny | | Overcast | |
| 0.3750 | 0.3750 | | 0.2500 | |
| 0 | -0.918295834 | | 0 | |
| 0 | 0.344360938 | | 0 | |
| **Entropy (Outlook)** | | **0.344360938** | |



**Infogain(Outlook) = (1 – 0.344360938) = 0.655639062**

H(**Temp**) = H(x = Hot) + H(x = Mild) + H(x = Cool)

H(**Temp**) = (3/8 \* -((2/3\*Log2(2/3) + 1/3\*Log2(1/3)) + (2/8 \* ((1/2\*Log2(1/2) + 1/2\*Log2(1/2)) + (3/8 \* ((2/3\*Log2(2/3) + 1/3\*Log2(1/3))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hot | Mild | | Cool | |
| 0.3750 | 0.2500 | | 0.3750 | |
| -0.918295834 | -1 | | -0.918295834 | |
| 0.344360938 | 0.25 | | 0.344360938 | |
| **Entropy (Temp)** | | **0.938721876** | |

**Infogain(Temp) = (1 – 0.938721876) = 0.061278124**

H(**Humidity**) = H(x = High) + H(x = Normal)

H(**Humidity**) = (5/8 \* (3/5\*LOG(3/5,2) + 2/5\*LOG(2/5,2)) + (3/8 \* (2/3\*LOG(2/3,2) + 1/3\*LOG(1/3,2)))

|  |  |  |
| --- | --- | --- |
| High | Normal | |
| 0.6250 | 0.3750 | |
| -0.970950594 | -0.918295834 | |
| 0.606844122 | 0.344360938 | |
| **Entropy (Humidity)** | | **0.951205059** | |

**Infogain(Humidity) = (1 – 0.951205059) = 0.048794941**

H(**Windy**) = H(x = False) + H(x = True)

H(**Windy**) = (5/8 \* ((3/5\*Log2(3/5)) + 2/5\*Log2(2/5)) + (3/8 \* (2/3\*Log2(2/3) + 1/3\*Log2(1/3))

|  |  |  |
| --- | --- | --- |
| FALSE | TRUE | |
| 0.6250 | 0.3750 | |
| -0.970950594 | -0.918295834 | |
| 0.606844122 | 0.344360938 | |
| **Entropy (Windy)** | | **0.951205059** | |
|  | |  | |

**Infogain(Windy) = (1 – 0.951205059) = 0.048794941**

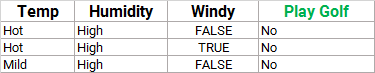
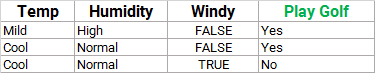
Hence the splitting the dataset along the feature **Outlook** results in the largest information gain and we should use this feature for our root node.  
The decision tree model looks like:

Create Root Node based on highest Information Gain

Outlook = Rainy

Outlook = Overcast

Outlook = Sunny

Yes

No

We see that for **Outlook = Rainy**, the target feature values of the remaining dataset are all **No** and hence we set this as leaf node because we have a pure dataset (Further splitting the dataset on any of the remaining two features would not lead to a different or more accurate result since whatever we do after this point, the prediction will remain No).

The same applies for **Outlook = Overcast,** the target feature values of the remaining dataset are all **Yes** and hence we set this as leaf node because we have a pure dataset. No further splitting is needed as it would not result in a different or more accurate result.

Additionally, you see that the feature Outlook is no longer included in the remaining datasets. Because we already had used this (categorical) feature to split the dataset on it must not be further used.

The same steps for information gain calculation must now be accomplished also for the remaining dataset for Outlook = Sunny since here we still have a mixture of different target feature values. So –

**Information gain calculation for the features Temp, Humidity and Windy for the remaining dataset Outlook = Sunny**

**Overall Entropy of the (new) sub data set after first split:**

H(D) = -((2/3 \* log2(2/3)) + (1/3 \* log2(1/3))) = **0.918295834**

|  |  |
| --- | --- |
| **Target Value** |  |
| Yes | -0.389975 |
| No | -0.528320834 |
| Overall Entropy | 0.918295834 |

**H(Temp) = -((1/3 \* (1\*log2(1,2)) + (2/3 \* (1/2 \* Log2(1/2) + 1/2\*Log2(1/2))**

|  |  |  |
| --- | --- | --- |
| Mild | Cool | |
| 0.3333 | 0.6667 | |
| 0 | -1 | |
| 0 | 0.666666667 | |
| **Entropy (Temp)** | | **0.666666667** | |

**Infogain(Temp) = (0.918295834 – 0.666666667) = 0.251629167**

**H(Humidity) = -((1/3 \* (1\*log2(1,2)) + (2/3 \* (1/2 \* Log2(1/2) + 1/2\*Log2(1/2))**

|  |  |  |
| --- | --- | --- |
| High | Normal | |
| 0.3333 | 0.6667 | |
| 0 | -1 | |
| 0 | 0.666666667 | |
| **Entropy (Humidity)** | | **0.666666667** | |

**Infogain(humidity) = (0.918295834 – 0.666666667) = 0.251629167**

**H(Windy) = -((1/3 \* (1\*log2(1,2)) + (2/3 \* (1/2 \* Log2(1/2) + 1/2\*Log2(1/2))**

|  |  |  |
| --- | --- | --- |
| True | False | |
| 0.3333 | 0.6667 | |
| 0 | -1 | |
| 0 | 0.666666667 | |
| **Entropy (Windy)** | | **0.666666667** | |

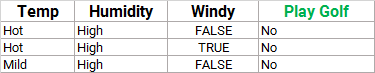
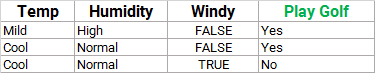
**Infogain(Windy) = (0.918295834 – 0.666666667) = 0.251629167**

Based on the **first** split dataset information gain, the tree can be grown on “Outlook = Sunny” and Temp. The Tree looks like below –

Outlook = Rainy

Outlook = Overcast

Outlook = Sunny

Create First Split Node based on highest Information Gain

Yes

No

Temp = Mild

Temp = Cool

** **

Yes

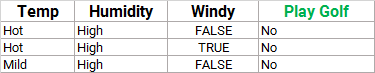
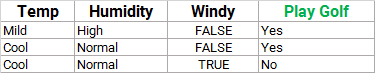
Based on the **second** split dataset information gain, the tree can still be grown on “Temp = Cool” and Windy. The fully-grown Tree to the Leaf Nodes as Target Values looks like below –

Create Root Node based on highest Information Gain

Outlook = Rainy

Outlook = Overcast

Outlook = Sunny

No

Create First Split Node based on highest Information Gain

Yes

Temp = Mild

Temp = Cool

** **

No

Windy = True

Yes

Windy = False

Yes

\*\* All are leaf nodes with Target Values/Predictions of the Model.

**Advantages –**

There are several advantages of using decision trees for predictive analysis and decision making -

1. Decision trees can be used to predict both continuous and discrete values i.e. they work well for both regression and classification tasks.
2. Easily Visualized, simple to interpret, understand and explain to Higher Management/Decision Makers (at least for smaller trees)
3. They require relatively less effort for training the algorithm.
4. They're very fast and efficient compared to [KNN](https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/) and other classification algorithms.
5. They can be used to classify non-linearly separable data and non-linear relationships between parameters do not affect Tree performance.
6. Decision Trees are invariant to the scalability of data and there is no need for pre-processing like Normalization or Standardization of Features.
7. The important variables for a Decision are automatically emphasized through the process of developing the tree. The top nodes of the tree are the most important because they determine the subsequent decisions to be made.
8. Every possible scenario from a Decision finds its representation by a clear fork and node, enabling viewing all possible solutions clearly in a single view.

The biggest drawback to decision trees though is that the split it makes at each node will be optimized for the dataset it is fit to. This splitting process will rarely generalize well to other data. Decision Trees tend to overfit even with the use of pre-pruning and provide poor generalization performance. However, we can generate huge numbers of these decision trees, tuned in slightly different ways, and ensemble their predictions to create some of our best models today.