LAB 11DECISION TREES

Solving Classification Problems with Decision Trees

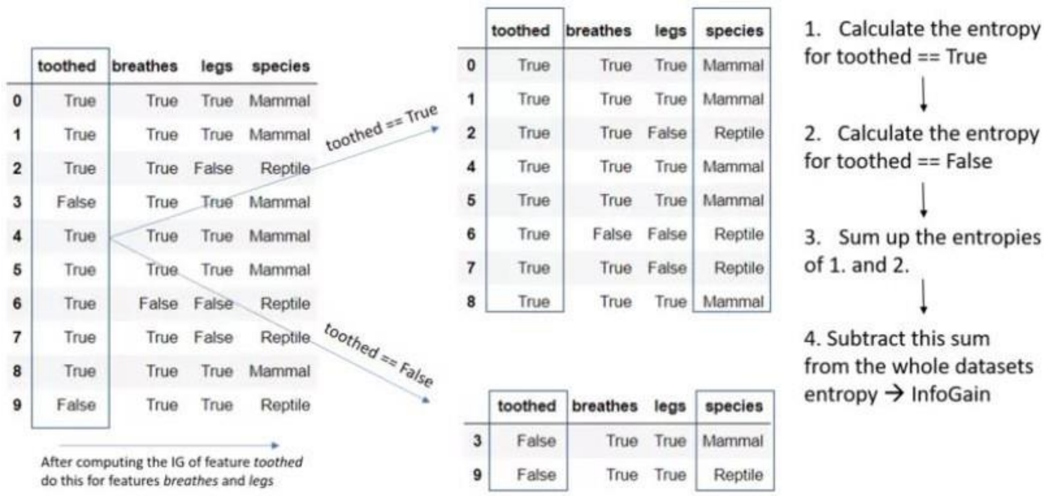
In the last lab we implemented some bits and pieces to decide that out of a given number of features how we can decide which of our **Feature** should go into the root of our Tree**.**In this lab we will implement ID3 algorithm

Information Gain

In the class you would have studied **information gain** as a measure of how gooda descriptive feature is suited to split a dataset on. To be able to calculate the information gain, we have to first implement the **entropy** of a dataset. The entropy of a dataset is used to measure the impurity of a dataset. 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 and Variance. The information gain of a feature is calculated by

$$InfoGain(feature_d) = Entropy(D) - Entropy(feature_d)$$

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.



$$H(toothed) = \underbrace{(\frac{8}{10}(-((\underbrace{\frac{5}{8} * log2(\frac{5}{8})}_{\text{toothed = True ; Mammal}})) + (\underbrace{\frac{3}{8} * log2(\frac{3}{8})}_{\text{toothed = True ; Reptile}}))))}_{\text{toothed = True}} + \underbrace{\frac{2}{10}(-((\underbrace{\frac{1}{2} * log2(\frac{1}{2})}_{\text{toothed = False ; Mammal}})) + (\underbrace{\frac{1}{2} * log2(\frac{1}{2})}_{\text{toothed = False ; Reptile}})))))}_{\text{toothed = False}}$$

Like this we will calculate the entropy and information gain of each attribute and select the best feature.

$InfoGain(toothed) = 0.971 - 0.963547 = \mathbf{0.00745}$

breathes:

$H(breathes) = (\frac{9}{10} * -((\frac{6}{9} * log_2(\frac{6}{9})) + (\frac{3}{9} * log_2(\frac{3}{9}))) + \frac{1}{10} * -((0) + (1 * log_2(1)))) = \mathbf{0.82647}$

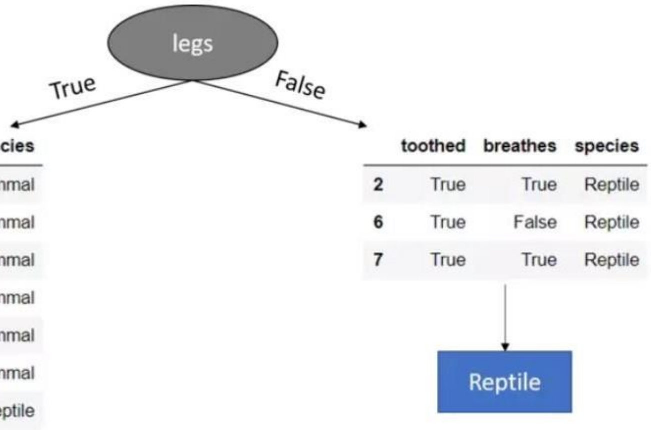
$InfoGain(breathes) = 0.971 - 0.82647 = \mathbf{0.1445}$

legs:

$H(legs) = \frac{7}{10} * -((\frac{6}{7} * log_2(\frac{6}{7})) + (\frac{1}{7} * log_2(\frac{1}{7}))) + \frac{3}{10} * -((0) + (1 * log_2(1))) = \mathbf{0.41417}$

$InfoGain(legs) = 0.971 - 0.41417 = \mathbf{0.5568}$

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



	toothed	breathes	species
0	True	True	Mammal
1	True	True	Mammal
3	False	True	Mammal
4	True	True	Mammal
5	True	True	Mammal
8	True	True	Mammal
9	False	True	Reptile

	toothed	breathes	species
2	True	True	Reptile
6	True	False	Reptile
7	True	True	Reptile

We see that for **legs == False**, the target feature values of the remaining dataset areall *Reptile* 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 *Reptile*). Additionally, you see that the feature *legs* is no longer included in theremaining datasets. Because we already has used this (categorical) feature to split the dataseton it must not be further used.Until now we have found the feature for the root node as well as a leaf node for ***legs == False***. The same steps for information gain calculation must now be accomplished also for the remaining dataset for ***legs == True*** since here we still have a mixture of different target feature values. Hence:

Information gain calculation for the features *toothed* and *breathes* for the remaining dataset *legs == True*:

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

$H(D) = -((\frac{6}{7} * log_2(\frac{6}{7})) + (\frac{1}{7} * log_2(\frac{1}{7}))) = \mathbf{0.5917}$

toothed:

$H(toothed) = \frac{5}{7} * -((1 * log_2(1)) + (0)) + \frac{2}{7} * -((\frac{1}{2} * log_2(\frac{1}{2})) + (\frac{1}{2} * log_2(\frac{1}{2}))) = \mathbf{0.285}$

$InfoGain(toothed) = 0.5917 - 0.285 = \mathbf{0.3067}$

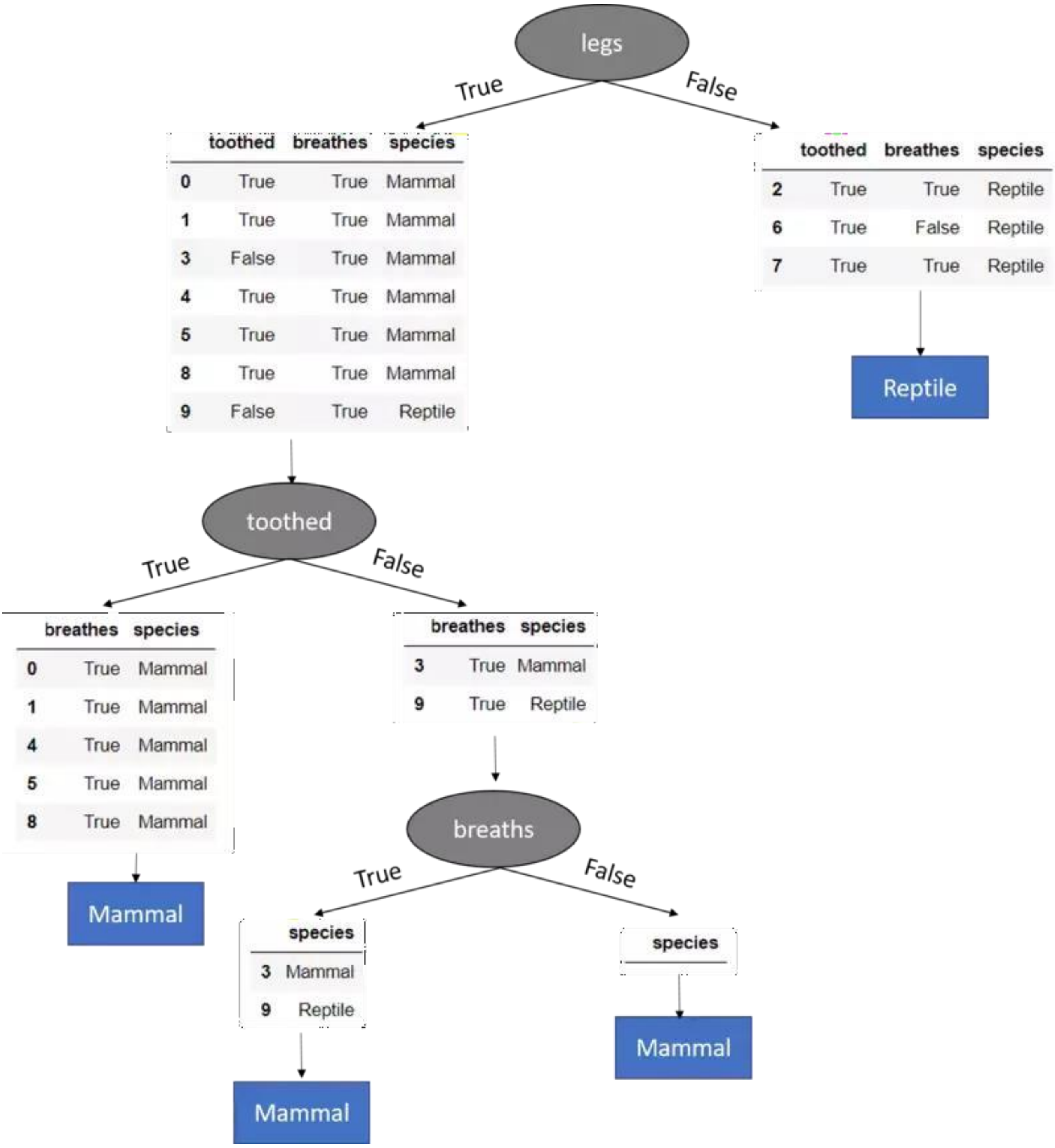
breathes:

$H(breathes) = \frac{7}{7} * -((\frac{6}{7} * log_2(\frac{6}{7})) + (\frac{1}{7} * log_2(\frac{1}{7}))) + 0 = \mathbf{5917}$

$InfoGain(toothed) = 0.5917 - 0.5917 = \mathbf{0}$

The dataset for toothed == False still contains a mixture of different target feature values why we proceed partitioning on the last left feature (== breathes)

Now our tree looks like this.



Mind the last split (node) where the dataset got split on the **breathes** feature. Here the breathes feature solely contains data where breaths == True. Hence for breathes == False there are no instances in the dataset and therewith there is no **sub-Dataset** which can be built. In that case we return the most frequently occurring target feature value in the original dataset which is **Mammal**. This is an example how our tree model generalizes behind the training data.

If we consider the other branch, that is breathes == True we know, that after splitting the Dataset on the values of a specific feature (breathes {True,False}) in our case, the feature must be removed. Well, that leads to a dataset where no more features are available to further split the dataset on. Hence we stop growing the tree and return the mode value of the direct parent node which is "**Mammal**".

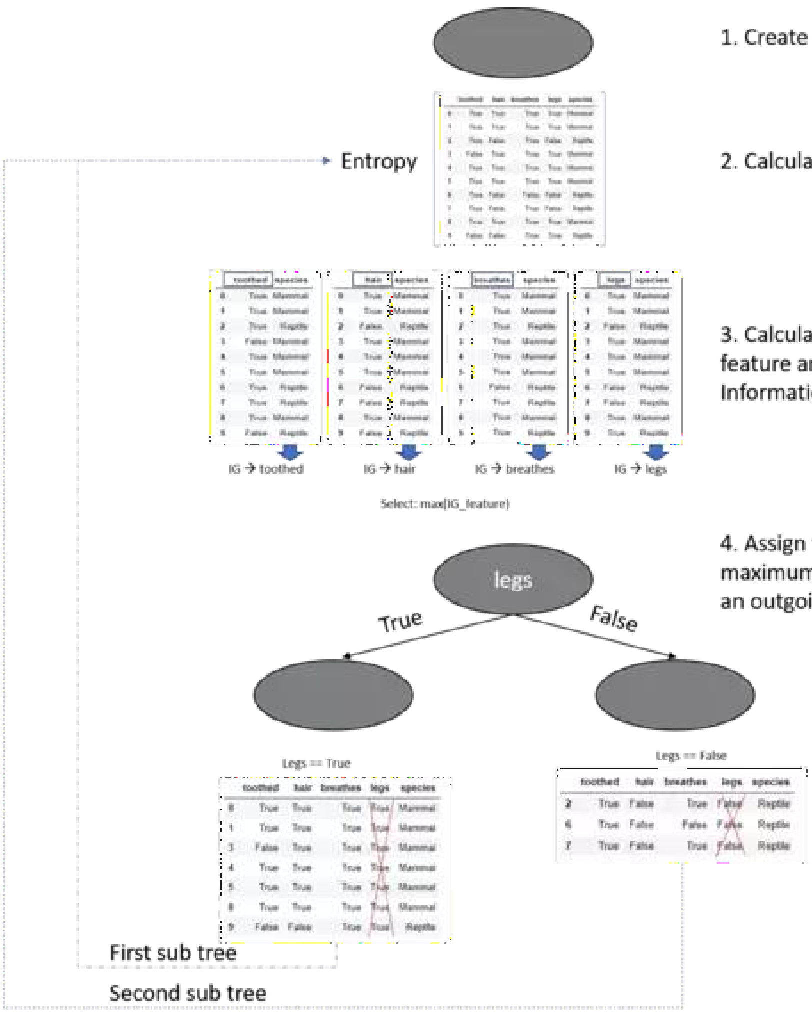
That leads us to the introduction of the ID3 algorithm which is a popular algorithm to grow decision trees, published by Ross Quinlan in 1986. Besides the ID3 algorithm there are also other popular algorithms like the C4.5, the C5.0 and the CART algorithm.
Let's come back to the stopping criteria of the above grown tree. We can define a nearly arbitrarily large number of stopping criteria. Assume for instance, we say a tree is allowed to grow for only 2 seconds and then the growing process should stop - Well that would be a stopping criteria - Nonetheless, there are mainly three useful cases in which we stop the tree from growing assuming we do not stop it beforehand by defining for instance a maximum tree depth or a minimum information gain value. We stop the tree from growing when:

1. All rows in the target feature have the same value
2. The dataset can be no longer split since there are no more features left
3. The dataset can no longer be split since there are no more rows left / There is no data left

By definition, we say that if the growing gets stopped because of stopping criteria two, the leaf node should predict the most frequently occurring target feature value of the superior (parent) node. If the growing gets stopped because of the third stopping criteria, we assign the leaf node the mode target feature value of the original dataset.

ID3 Pseudo Code

```
ID3(D,Feature_Attributes,Target_Attributes)

  Create a root node r

  Set r to the mode target feature value in D

  If all target feature values are the same:
    return r

  Else:
    pass

  If Feature_Attributes is empty:
    return r

  Else:
    Att = Attribute from Feature_Attributes with the largest information gain value
    r = Att

    For values in Att:
      Add a new node below r where node_values = (Att == values)
      Sub_D_values = (Att == values)

      If Sub_D_values == empty:
        Add a leaf node l where l equals the mode target value in D
      Else:
        add Sub_Tree with ID3(Sub_D_values,Feature_Attributes = Feature_Attributes
without Att, Target_Attributes)
```



1. Create a root node

2. Calculate the entropy of the whole (sub) dataset

3. Calculate the Information gain of each single feature and pick that feature with the larges Information gain

4. Assign the (root) node the label of the feature with the maximum information gain. Grow for each feature value an outgoing branch and add unlabelled nodes at the end

5. Split the dataset along the values of the maximum information gain feature and remove this feature from the dataset

6. For each of the sub_datasets, repeat steps 2 to 5 until a stopping criteria is satisfied →Here the recursion kicks in

Task:
You need to implement the ID3 algorithm. Open the provided notebook and complete the solution.