

Data Representation and Models

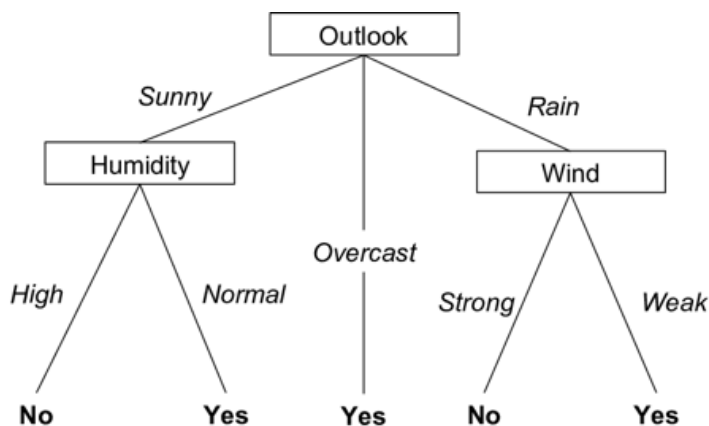
(find and work with a partner)

1. Consider the tennis dataset shown below. What is n (number of data points)? What is p (number of features)?

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis (y) |
|----------|----------|-------------|----------|--------|--------------------|
| x_1 | Sunny | Hot | High | Weak | No |
| x_2 | Sunny | Hot | High | Strong | No |
| x_3 | Overcast | Hot | High | Weak | Yes |
| x_4 | Rain | Mild | High | Weak | Yes |
| x_5 | Rain | Cool | Normal | Weak | Yes |
| x_6 | Rain | Cool | Normal | Strong | No |
| x_7 | Overcast | Cool | Normal | Strong | Yes |
| x_8 | Sunny | Mild | High | Weak | No |
| x_9 | Sunny | Cool | Normal | Weak | Yes |
| x_{10} | Rain | Mild | Normal | Weak | Yes |

Data and model from Machine Learning by Tom Mitchell (Table 3.2)

2. How would you *featurize* this data? In other words, if you needed each feature to be numerical, how would you map the current feature values to numerical values?
3. Using your response from the previous question, what would the *feature vector* become for x_1 ?
4. In the model below, the children of each node divide the data into partitions. Label each node (both internal nodes and leaves) with the counts of “No” and “Yes” labels based on the partition. For example, the counts for the node labeled *Outlook* would be [4,6] (correction!) Does this model perfectly classify all examples?



5. What label (i.e. play tennis or not) would you *predict* for the feature vector [Rain, Hot, High, Strong]?