Let's imagine you are playing a game of [Twenty Questions](http://en.wikipedia.org/wiki/Twenty_Questions).  Your opponent has secretly chosen a subject, and you must figure out what she chose.  At each turn, you may ask a yes-or-no question, and your opponent must answer truthfully.  How do you find out the secret in the fewest number of questions?  
  
It should be obvious some questions are better than others.  For example, asking "Is it a basketball" as your first question is likely to be unfruitful, whereas asking "Is it alive" is a bit more useful.  Intuitively, we want each question to significantly narrow down the space of possibly secrets, eventually leading to our answer.  
  
That is the basic idea behind decision trees.  At each point, we consider a set of questions that can partition our data set.  We choose the question that provides the best split (often called maximizing information gain), and again find the best questions for the partitions.  We stop once all the points we are considering are of the same class (in the naive case).  Then classifying is easy.  Simply grab a point, and chuck him down the tree.  The questions will guide him to his appropriate class.