1a) Explain the decision tree algorithm.

What is a Decision Tree?

A decision tree is a tree-based supervised learning method used to predict the output of a target variable. Supervised learning uses labeled data (data with known output variables) to make predictions with the help of regression and classification algorithms. Supervised learning algorithms act as a supervisor for training a model with a defined output variable. It learns from simple decision rules using the various data features. Decision trees in Python can be used to solve both classification and regression problems—they are frequently used in determining odds.

The following is an example of a simple decision tree used to classify different animals based on their features. We will be using the color and height of the animals as input features.

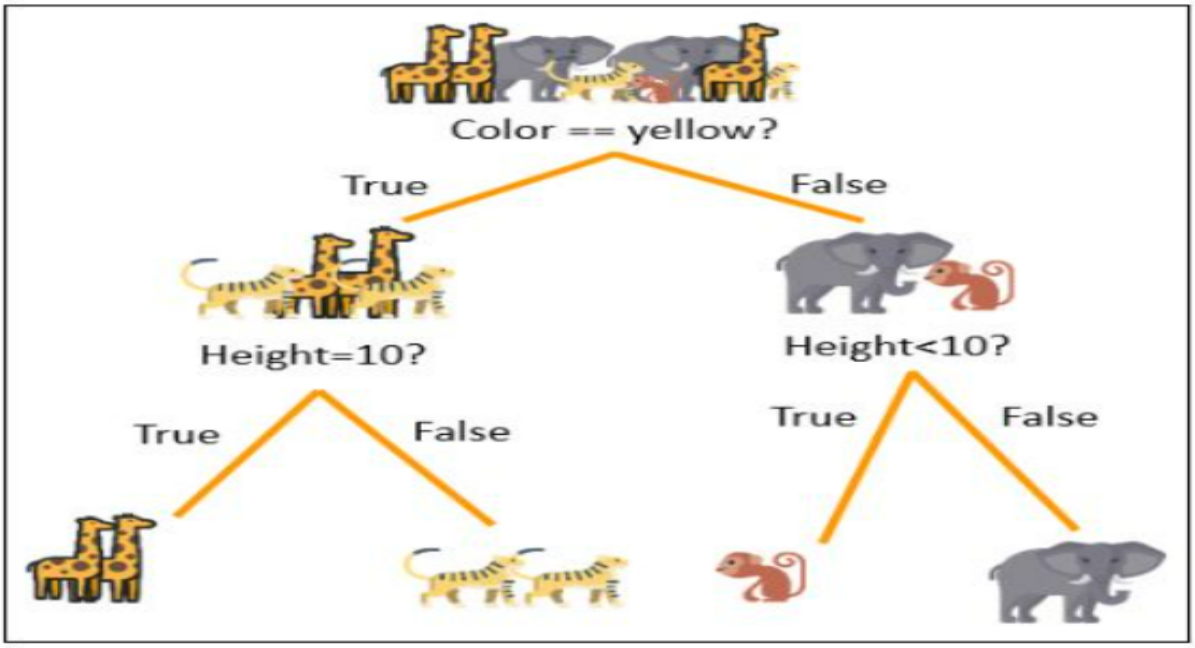

Fig: Decision tree to classify animals

Advantages of Using Decision Trees

- **Decision trees are simple to understand, interpret, and visualize**
- **They can effectively handle both numerical and categorical data**
- **They can determine the worst, best, and expected values for several scenarios**
- **Decision trees require little data preparation and data normalization**

- **They perform well, even if the actual model violates the assumptions**

Decision Tree Applications

1. **A decision tree is used to determine whether an applicant is likely to default on a loan.**
2. **It can be used to determine the odds of an individual developing a specific disease.**
3. **It can help ecommerce companies in predicting whether a consumer is likely to purchase a specific product.**
4. **Decision trees can also be used to find customer churn rates.**

How Does a Decision Tree Algorithm Work?

Suppose there are different animals, and you want to identify each animal and classify them based on their features. We can easily accomplish this by using a decision tree.
The following is a cluttered sample data set with high entropy:

Training Dataset

Color	Height	Label
Grey	10	Elephant
Yellow	10	Giraffe
Brown	3	Monkey
Grey	10	Elephant
Yellow	4	Tiger

We have to determine which features split the data so that the information gain is the highest. We can do that by splitting the data using each feature and checking the information gain that we obtain from them. The feature that returns the highest gain will be used for the first split.

We'll use the information gain method to determine which variable yields the maximum gain, which can also be used as the root node.

Suppose Color == Yellow results in the maximum information gain, so that is what we will use for our first split at the root node.

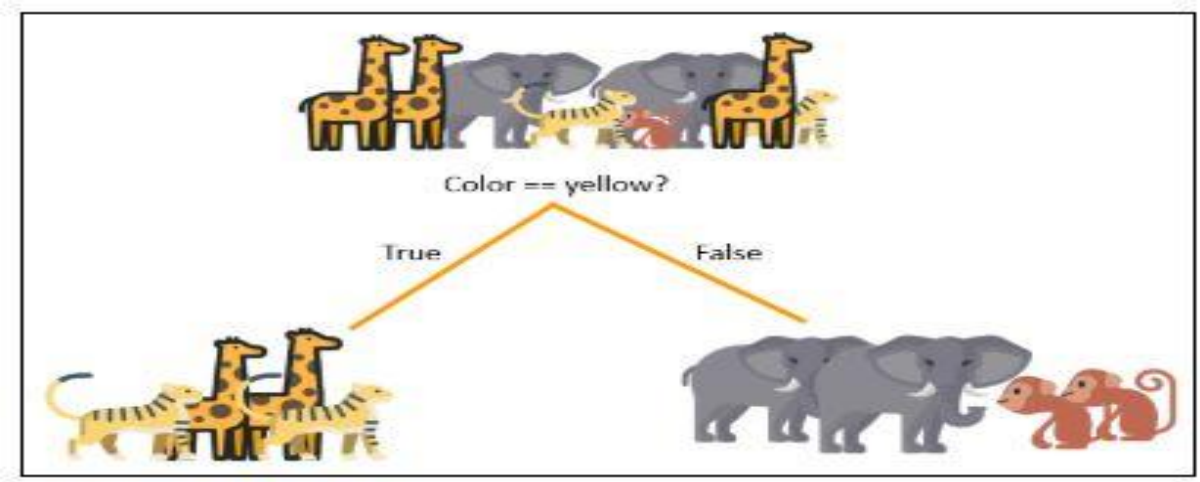

Fig: Using Color == Yellow for our first split of decision tree

The entropy after splitting should decrease considerably. However, we still need to split the child nodes at both the branches to attain an entropy value equal to zero.

We will split both the nodes using ‘height’ variable and height > 10 and height < 10 as our conditions.

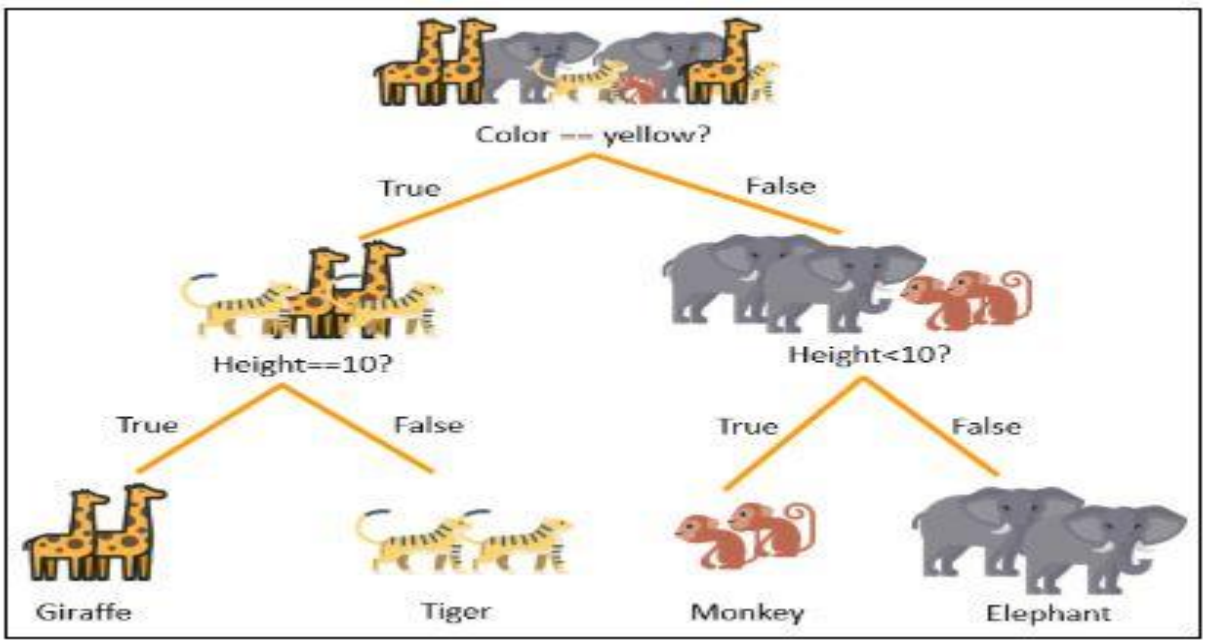

Fig: Slitting the decision tree with the height variable

The decision tree above can now predict all the classes of animals present in the data set.