NAME: Nachiket Erlekar
Roll no,: 41434 Batch: Q4

ML Assignment No. 2

2.1 Title

Assignment based on Decision Tree Classfier

2.2 Problem Definition:

A dataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future. Find the root node of decision tree. According to the decision tree you have made from previous training data set, what is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?

2.3 Prerequisite:

Basic of Python, Data Mining Algorithm, Concept of Decision Tree Classifier

2.4 Software Requirements:

Anaconda with Python 3.7

2.5 Hardware Requirement:

PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

2.6 Learning Objectives:

Learn How to Apply Decision Tree Classifier to find the root node of decision tree. According to the decision tree you have made from previous training data set.

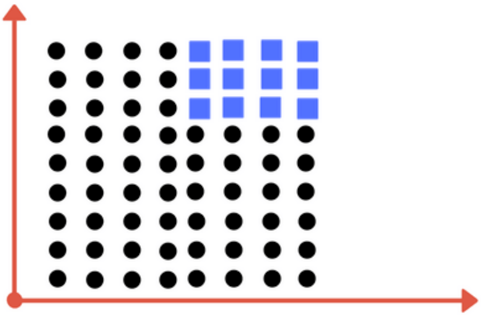
2.7 Outcomes:

After completion of this assignment students are able Implement code for Create Decision tree for given dataset and find the root node for same based on the given condition.

2.8 Theory Concepts:

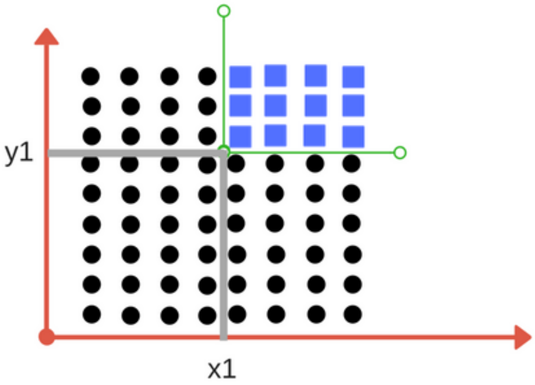
2.8.1 Motivation

Suppose we have following plot for two classes represented by black circle and blue squares. Is it possible to draw a single separation line ? Perhaps no.



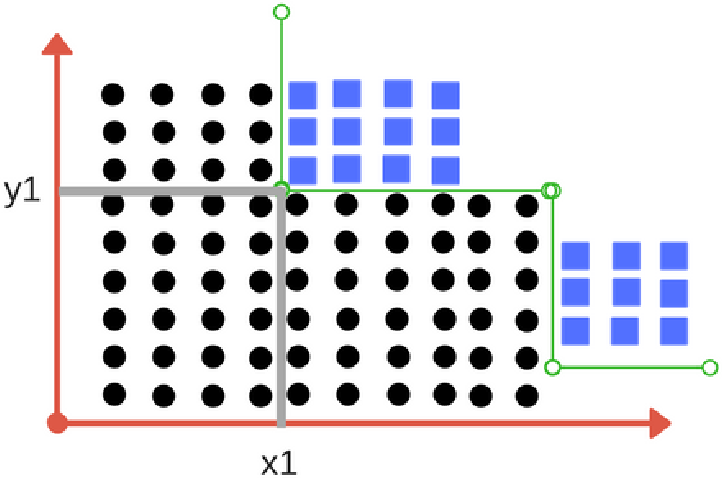
Can you draw single division line for these classes?

We will need more than one line, to divide into classes. Something similar to following image:



We need two lines one for threshold of x and threshold for y.

We need two lines here one separating according to threshold value of ***x*** and other for threshold value of ***y.*** *Decision Tree Classifier, repetitively divides the working area(plot) into sub part by identifying lines.* (repetitively because there may be two distant regions of same class divided by other as shown in image below).

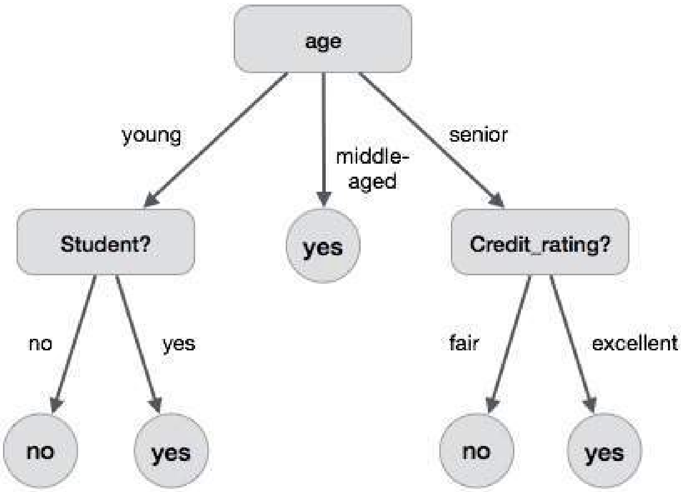


So when does it terminate?
Either it has divided into classes that are pure (only containing members of single class)
Some criteria of classifier attributes are met.

Impurity-
In above division, we had clear separation of classes. But what if we had following case?
Impurity is when we have a traces of one class division into other. This can arise due to following reason
We run out of available features to divide the class upon.
We tolerate some percentage of impurity (we stop further division) for faster performance. (There is

always trade off between accuracy and performance).

For example in second case we may stop our division when we have x number of fewer number of elements left. This is also known as *gini impurity.*



Division based on some features.

2. Entropy

Entropy is degree of randomness of elements or in other words it is measure of impurity. Mathematically, it can be calculated with the help of probability of the items as:

$$H = -\sum p(x) \log p(x)$$

p(x) is probability of item x.

$$H = -\sum p(x) \log p(x)$$

It is negative summation of probability times the log of probability of item x.

For example,

if we have items as number of dice face occurrence in a throw event as 1123,

the entropy is

p(1) = 0.5

p(2) = 0.25

p(3) = 0.25

entropy *= - (0.5 * log(0.5)) - (0.25 * log(0.25)) -(0.25 * log(0.25)*

= ***0.45***

3. Information Gain

Suppose we have multiple features to divide the current working set. What feature should we select for division? Perhaps one that gives us less impurity.

Suppose we divide the classes into multiple branches as follows, the information gain at any node is defined as

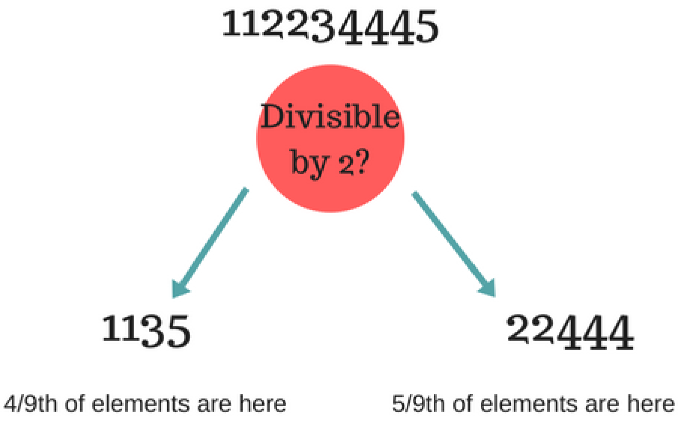
Information Gain (n) = Entropy(x) — ([weighted average] * entropy(children for feature))

This need a bit explanation!

Suppose we have following class to work with intially

112234445

Suppose we divide them based on property: divisible by 2



Entropy at root level : 0.66

Entropy of left child : 0.45 , weighted value = (4/9) * 0.45 = 0.2

Entropy of right child: 0.29 , weighted value = (5/9) * 0.29 = 0.16

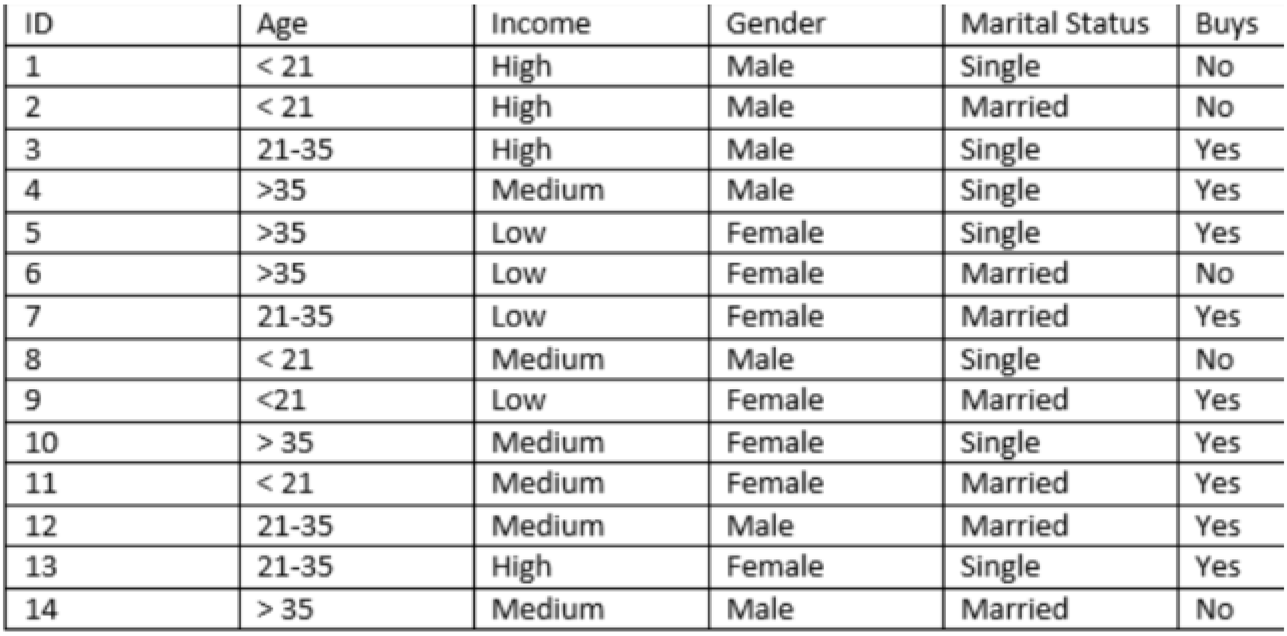
Information Gain = 0.66 - [0.2 + 0.16] = ***0.3***

Check what information gain we get if we take decision as **prime number instead of divide by 2.** Which one is better for this case?

Decision tree at every stage selects the one that gives best information gain.

When information gain is 0 means the feature does not divide the working set at all.

Given Data set in Our Definition



ID	Age	Income	Gender	Marital Status	Buys
1	< 21	High	Male	Single	No
2	< 21	High	Male	Married	No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low	Female	Single	Yes
6	>35	Low	Female	Married	No
7	21-35	Low	Female	Married	Yes
8	< 21	Medium	Male	Single	No
9	<21	Low	Female	Married	Yes
10	> 35	Medium	Female	Single	Yes
11	< 21	Medium	Female	Married	Yes
12	21-35	Medium	Male	Married	Yes
13	21-35	High	Female	Single	Yes
14	> 35	Medium	Male	Married	No

What is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?

Answer is Whether Yes or No??

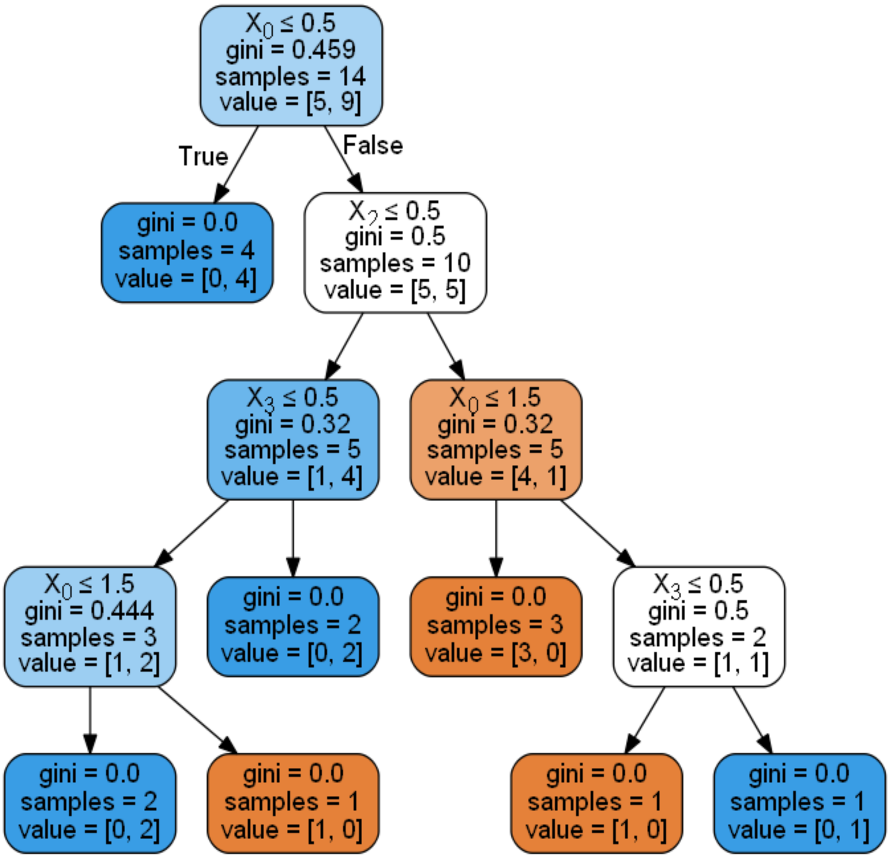
1. Which of the attributes would be the root node.
A. Age
B. Income
C. Gender
D. Marital Status
Solution: C
2. What is the decision for the test data [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?
A. Yes
B. No
Solution: A
[Hints: construct the decision tree to answer these questions]

Algorithm

Import the Required Packages
Read Given Dataset

1. Perform the label Encoding Mean Convert String value into Numerical values
2. Import and Apply Decision Tree Classifier
3. Predict value for the given Expression like [Age < 21, Income = Low, Gender = Female, Marital Status = Married]? In encoding Values [1,1,0,0]
4. Import the packages for Create Decision Tree.
5. Check the Decision Tree Created based on Expression.

Decision Tree Generated after Implementation of Code



2.12 Conclusion

In this way we learn that to how to create Decision Tree based on given decision, Find the Root Node of the tree using Decision tree Classifier.