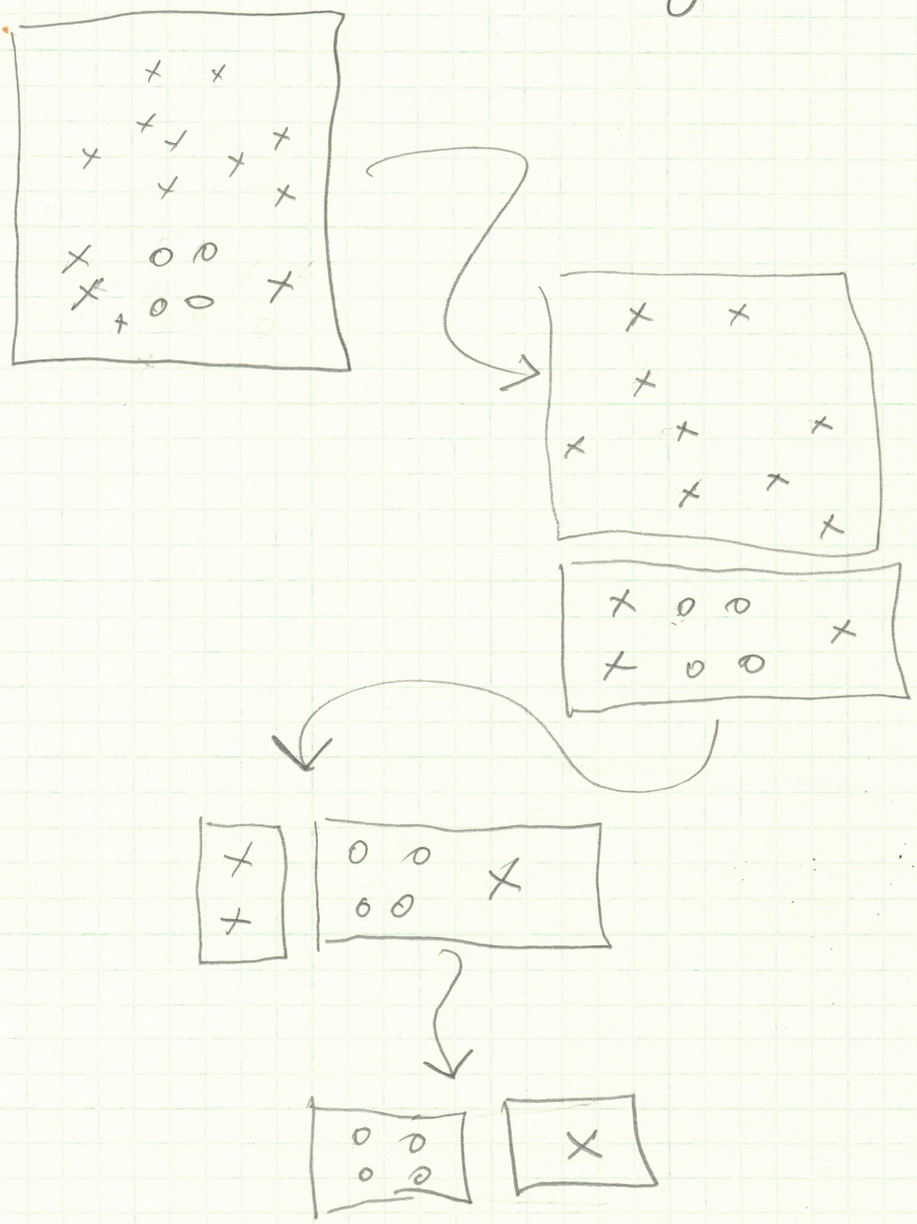
Notes based on article by Robert Kubler towards Data Science

Decision Trees

- Decision Trees seperate / splits data into groups based on "information gain"
- One really good way to implement the algorithm is to do it recursively



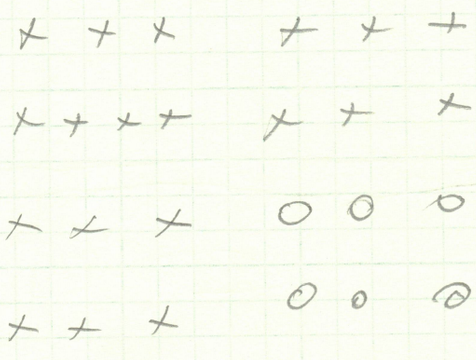
We stop splitting when:

- the split result in a part being empty
- A certain recursion depth is reached
- not much data is in a split, making more splits un-necessary

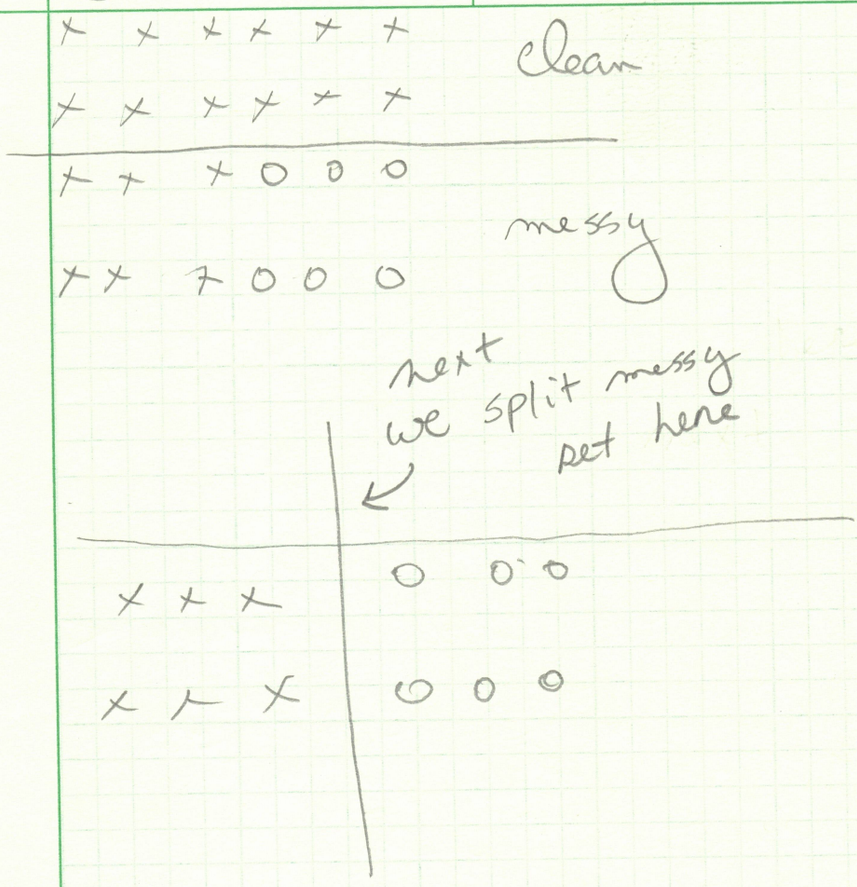
> However, looking at the example, notice we split data into 2 parts:

- a homogeneous part
- a non-homogeneous part

The recursion, or splitting follows the non-homogeneous part of the data.



- The idea idea is too split the data into two parts - these parts should be "cleaner" than the complete data before



- We need a measure of cleaness

  here are 3 arrays

  $x = [0\,0\,0\,0\;\;0\,0\,0\,0]$ cleanest

  $y = [1\,0\,0\,0\,0\;\;0\,0\,1\,0]$ meh!

  $z = [1\,0\,1\,1\,0\;\;0\,1\,0]$ messyest

- A simple measure (for 2 class data)

$$\frac{|\text{number class}_0 - \text{number class}_1|}{\text{total set size}} = \frac{|n_0 - n_1|}{n}$$

(3)

$$\frac{|n_0 - n_1|}{n} = \frac{|(n - n_1) - n_1|}{n} = \frac{|n - 2n_1|}{n}$$

(n_0)

$$= \left|\frac{n}{n} - \frac{2n_1}{n}\right| = \left|1 - 2\frac{n_1}{n}\right|$$

let $p_1 = \frac{n_1}{n}$; so $|1 - 2p_1|$

p_1 is the percentage of n_1 labels

let $n = n_0 + n_1$

- This is set up for 2 class data

$X = [0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 0]$

$|1 - 2p_1| = |1 - 2 \cdot 0| = \boxed{1}$

$p_0 = \frac{8}{8} = 1$; $p_1 = \frac{0}{8} = 0$

$y = [1\ 0\ 0\ 0\ \ 0\ 0\ 1\ 0]$

$p_0 = \frac{6}{8} = \frac{3}{4}$; $p = \frac{2}{8} = \boxed{\frac{1}{4}}$

$|1 - 2 \cdot \frac{1}{4}| = .5$

$z = [1\ 0\ 1\ 1\ \ 0\ 0\ 1\ 0]$

$p_0 = \frac{4}{8} = \frac{1}{2}$; $p_1 = \frac{4}{8} = \frac{1}{2}$

$|1 - 2 \cdot \frac{1}{2}| = \boxed{0}$

A more general formula is the Gini impurity measure:

$$gini(p_0, p_1) = p_0(1-p_0) + p_1(1-p_1)$$

The above is the 2 class case, below is the multi class case

$$gini(p) = \sum_{i=1}^{n} p_i(1-p_i)$$

here $p_0 = \frac{n_0}{n}$, $p_1 = \frac{n_1}{n}$

and

$$p_i = \frac{n_i}{n}$$

> looking at x, y, and z

x $gini(1,0) = \frac{8}{8}(1-1) + \frac{0}{8}(1-0) = 0$

y $gini(\frac{6}{8}, \frac{2}{8}) = \frac{6}{8}(1-\frac{6}{8}) + \frac{2}{8}(1-\frac{2}{8})$

$\frac{6}{8}(\frac{2}{8}) + \frac{2}{8}(\frac{6}{8})$

$\frac{12}{64} + \frac{12}{64} = \frac{24}{64} = \frac{3}{8}$

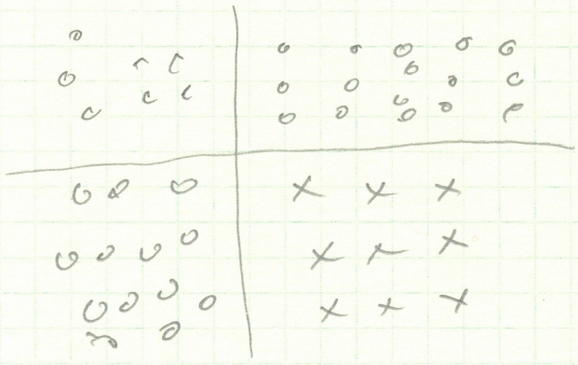
z $gini(\frac{1}{2}, \frac{1}{2}) = \frac{1}{2}(1-\frac{1}{2}) + \frac{1}{2}(1-\frac{1}{2})$

$\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$

(5) So unlike our previous calculations, the more "disordered" or "impure" the set is, the higher the gini value

> Finding where to split the data.

We will split the data along either the x or y axis using a gini measure



suppose 75% of the points are O
25% of the points are X

$$\text{gini}(.75, .25) = .25(1-.25) + .75(1-.75)$$
$$= .375$$

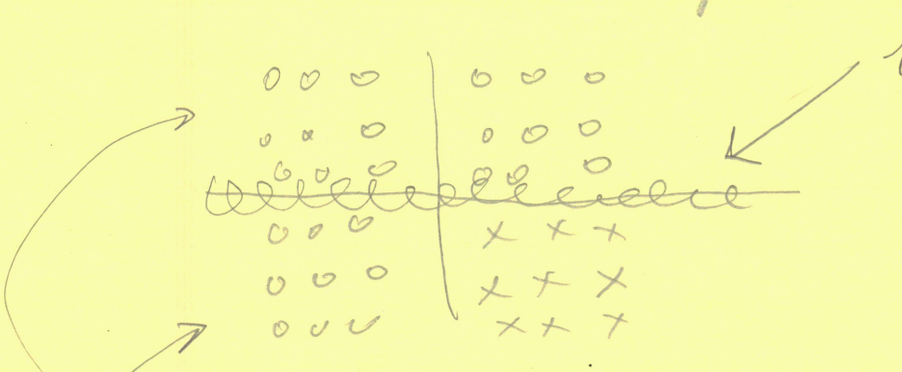
↑
over all ginni

We want to find a split where the gini on 1 side of split shows

- very pure

and the other side

- not so pure



$$gini(1,0) = 1(1-1) + 0(1-0) = 0$$

$$gini(.5,.5) = .5(1-.5) + .5(1-.5) = .25 + .25 = .5$$

So using this split, we can average the gini values

$$\frac{(0 + .5)}{2} = .25$$

- Remember, the gini value without the split was .3075

  .25 < .375 so this is a better measure

  - the lower the gini number, the better

→ we could have done a split on $y = 0$ and that would have been good too.

! last thing, we should use a weighted average when calculating our average gini #'s

Suppose part 1 has 50 points
part 2 has 450 points

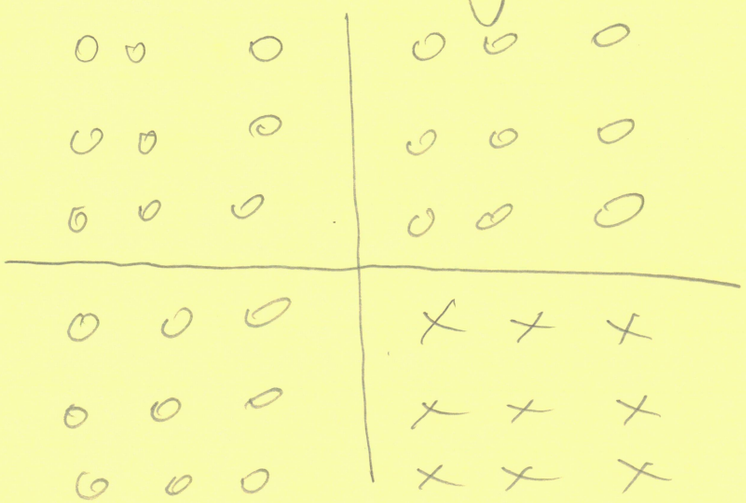
instead of calculating

$$\text{gini Average} = \frac{(0 + .5)}{2} = .25$$

we should do:

$$\text{gini Average} = \frac{50}{50+450}(0) + \frac{450}{50+450}(.5) = .45$$

How do we find the split?



While (splitting) {

find the y range of the data

for every y value

find the weighted gini average

y value with lowest gini average is y split

keep the data in the impure side of the split

if both sides have gini values <= pure enough

splitting = false

(9)

if (splitting) {

find the x range of the data

for every X value

find the weighted gini average

x value with the lowest gini average is the y split

keep the data in the impure side of split

if both sides have gini values <= pure enough

splitting = false

}

} /* end while */