## [CAP4611-21Spring](https://webcourses.ucf.edu/courses/1369384/calendar_events/2158980)

# Variant 1

# Day 6 (Thursday, Jan 28):

STONKS LECTURE

If youre in the UCF CS discord, go to bot\_commands and do a bot command !register CAP4611\_hollander

Quiz today:

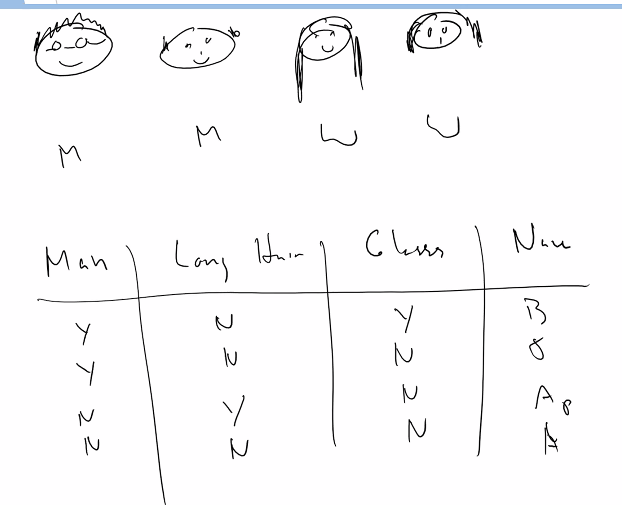
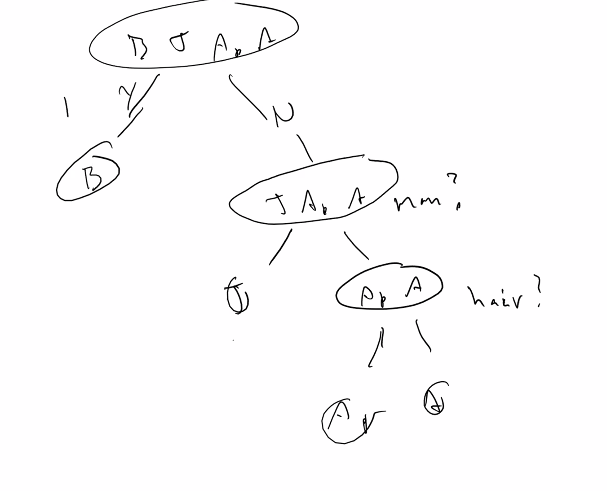
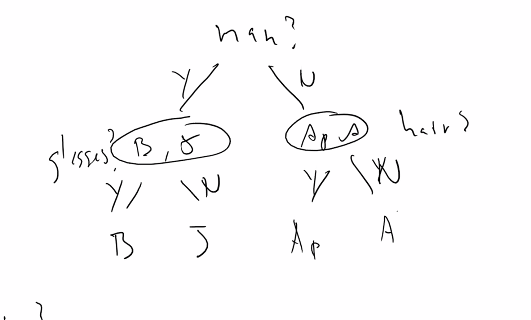
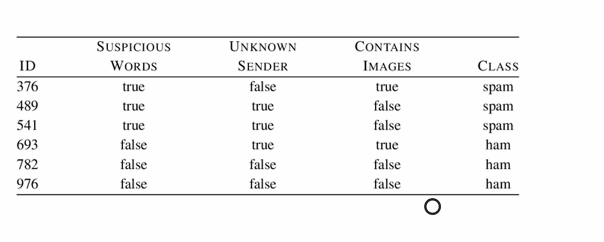
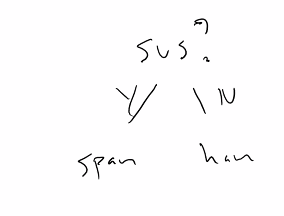
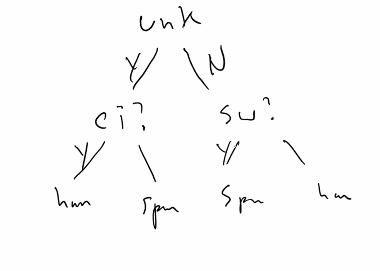
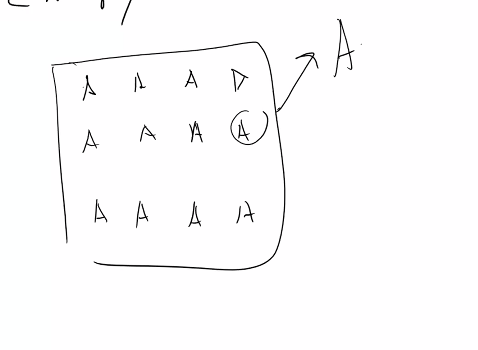
* “I have decided to make quizzes during the start of the class, due to this whole pandemic situation, think of it as an in person pop quiz… gives you an incentive to study material”
* Quizzes will be done every time we finish a topic (?)
* Generally plan for one quiz a week
* We might get sidetracked sometimes, then it’ll get pushed back
* 20% of final grade is from quizzes
* 60% from projects

Administrative stuff

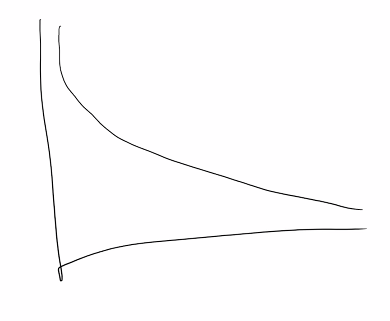
* Notes for decision trees will be posted after we fully cover it
* Webcourses updated
* Once we go about predictions, we will talk about evaluation, then decision trees
* If I can get the assignments on Kaggle, I will probably get you an assignment every two weeks
* Midterm stuff will be talked about later, theoretical questions and some math problems.
* It may/maynot be during class time

Talking about our first Machine Learning algorithm today:

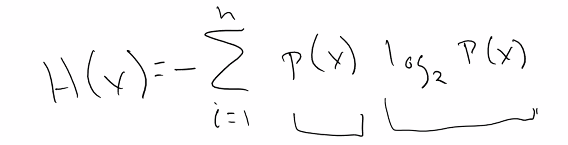
Decision trees:

* Basic idea
  + Assume you have some collection of people
  + 
  + They have different hair styles, (bald, short, long, medium (?))
  + Might have glasses
  + They’re men || women
  + There are a few questions we can ask
    - Man?
    - Long Hair?
    - Glasses?
    - Names
  + How many questions do we have to ask to figure out who we’re talkin about?
    - We ask one of the questions, this will rule out some of the individuals
    - We keep asking questions until we reach one of the possible options.
    - Mapping this out in a tree diagram looks like this:
    - 
    - Note that it takes a different amount of questions to figure out who we need to select
    - For the diagram above, it takes an average of 2.5 (?) questions to figure out who the person is.
    - It is possible to reduce the amount of questions we need to ask by asking the questions in a different order.
    - 
    - Thats great, but what does this fundamentally lead us to?
      * What is the optimal way we can ask the questions?
    - This structure is called a decision tree, they usually take the form of a binary tree, but don’t have to
    - One of the really powerful things about a decision tree is that you can explain how we got that prediction
    - This is different than using **Neural Networks** (as you will have no idea)
    - [Benefit] decision tree is easy to do
    - Q: why do we use NN instead of decision trees?
      * NN give us better answers
    - [Example] Now looking at this dataset:
    - 
    - How can we tell if email is spam or ham? (ham == !spam)
    - What is the first question we should ask?
      * Suspicious wording?
        + Yes/No
        + Notice that everything that is spam has sus wording
        + So we will need to only ask this question for this dataset to figure it out
        + 
      * Unknown Sender?
        + Yes/No
        + Yes, then does it contain images?
        + No, is it sus?
        + 
      * Which of the two trees above would we prefer?
        + The smaller one.
      * **Decision trees are greedy algorithms** (this correct statement?), they prefer smaller trees
      * Decision trees only care about the data you have
      * If we chose to go with the suspicious words question first,
        + Any suspicious wording will toss the message into the spam folder
      * If you have a decision tree with only one node, it probably won’t generalize well. (But it’d be fast)
      * This might lead you to think the deeper the better,
        + But this is false, deeper trees lead to overfitting
        + The most amount of leafs we can have in a decision tree is the amount of entries in the data.
      * Entropy:
        + The measurement of uncertainty in a set when you give a random point
        + [Example] you have a bunch of playing cards, all but one are Aces
        + 
        + If we take one of the cards out, and its an Ace, does that tell anything about the data?

(Not really)

* + - * + Think of Entropy as the heterogeneity of a set
        + So the general notion of entropy is that, the higher the probability that something is going to occur, the less entropy it has
        + The less likely it is to happen the more entropy.
        + High probability ~~~~ low entropy
        + Low probability ~~~~ high entropy
        + The more disorder there is in a set, the more entropy
        + Remember, probability ranges from [0-1]
        + Is there a way for (?)
        + What we would like is a function that will gives us a high value for 0, and a low value for 1. Something that looks like this:
        + If we take the -log() of probability, we will get something in between 0 and 1.
        + From here, we know that P(x) ---> [0,1]
        + Then -log(P(x)) will give us the entropy of it
      * Entropy and Columns
        + How do we find entropy of a column?
        + Think of a column as a set of numbers
        + We would take the average column.
        + In this situation, we would:

For every value in the column, the probability that we actually observe that value times the log of that probability negated

* 

log2{(x)} is the magnitude, p(x) is the expected occurrence (??)

Why is it in log2()?

We are dealing with computers, also has something to do with probability, but it doesn’t matter

With entropy, we can figure out how much of a variable we can split off (???)

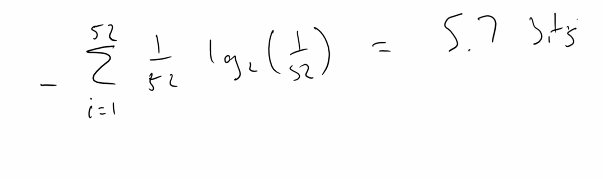
The smaller the probability of occurrence, the larger the log.

* + - * + Calculator example [Entropy Example]:

Think of a deck of cards, how many cards are in a deck? 52

If we wanted to know the entropy of a deck of playing cards, what is the probability of picking any card at random?

1/52

* 

5.7 bits **ENTROPY IS MEASURED IN BITS**

Instead if we wanted to select a card, and didn’t care about the suit, how will that change the outcome?

1/13 instead of 1/52

H(x) = entropy equation

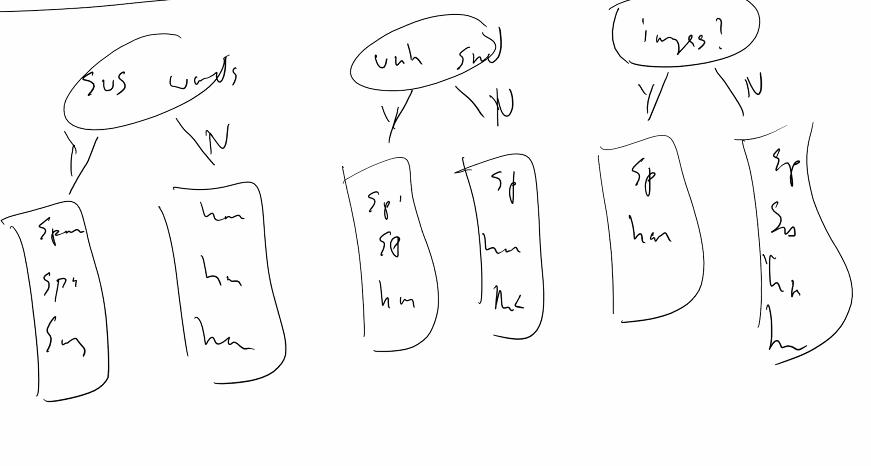
H(1/13) = ~2 bits

What constitutes high entropy? It depends.

* + - * + How is Entropy Useful?

We can go back to our decision tree.

Think about our suspicious words, unknown sender, and images? from the email problem

* 

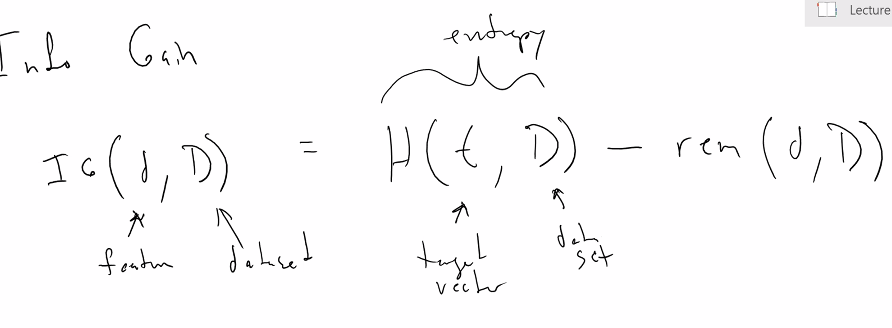
Regarding the questions we ask, we would want to talk about the notion of information gain

* + - * + Information gain:

How much information is gained based on the information provided

If we want to calculate the information gain, the formula is:

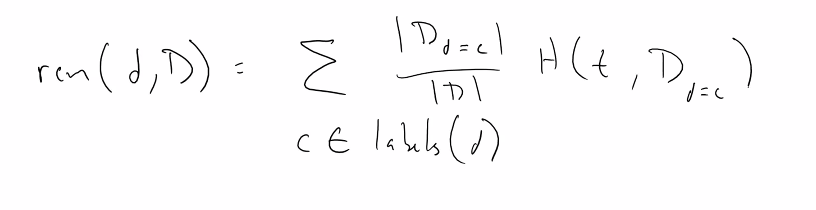
IG(Feature, Dataset) = H(TargetVector, Dataset) - remainder(Feature, Dataset)

* 

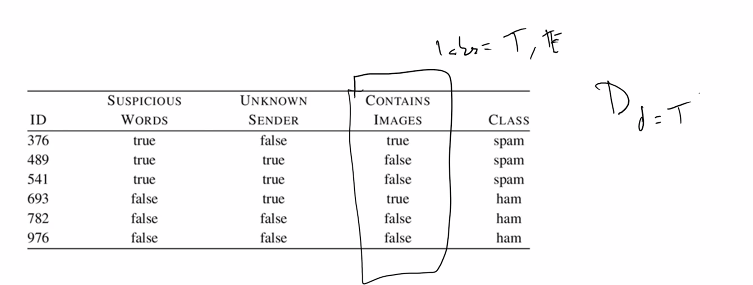
(remember: targetVector -- is the answer you care about)

* + - * + What is the remainder?

Summation of all the labels within that feature, in terms of …

* 
  + - * + When we think about a targetvector or a feature, we can talk about…
        + Feature: Xi :
        + 
        + The labels for Xi will be:

labels(xi) = {cat, rat, dog}

* + - * + In this, we will take the entropy of a target vector (some value in xi) in the dataset, we would need to go through this feature 3 times...
      * Back 2 email question:
* 
* The dataset where D is true (contains images), we will see:
  + CI = T true false (?...)
* The dataset where D is false:
  + CI = T
* Basically, entropy allows us to figure out the information gain for each feature. With this, we can then figure out what questions to ask.
* Chapter 4: is the stuff we just covered
* In terms of sorting, this will come into play when we are doing continuous stuff.
* Building a decision tree is a **recursive algorithm**

Recording ends.

* We will not write a decision tree from scratch, just cover the theory surrounding it
* Regarding performance of a decision tree, its “not to bad”
* The prediction of a decision tree is basically linear
* In python, you can use dictionaries,
* **For your actual assignments, your going to use sci-kit learning api**
* **There are some “issues with it”**
* **It does not handle continuous data appropriately, you will have to mess around with it a bit.**

Quiz recap:

* “This is not a bad distribution, I’m gonna be completely honest with you”
* **I’ve gtg somebody take over**