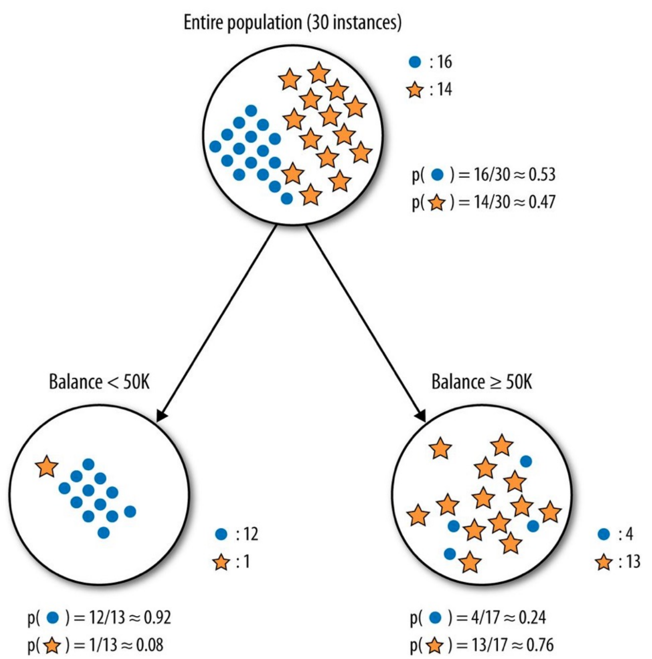
Entropy, KL Divergence all that jazz

Definition?

- H(X) = $-\Sigma\, p_i * \log_2 p_i$
  - How is that helpful?
- Entropy "is a measure of how many bits it takes to represent an observation of X on average"
  - Uh….

- All the definitions above are correct but honestly, say what now?

Motivating Example (simple decision tree) - Part 1



$$E(Parent) = -\frac{16}{30}\log_2\left(\frac{16}{30}\right) - \frac{14}{30}\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13}\log_2\left(\frac{12}{13}\right) - \frac{1}{13}\log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17}\log_2\left(\frac{4}{17}\right) - \frac{13}{17}\log_2\left(\frac{13}{17}\right) \approx 0.79$$

Weighted Average of entropy for each node:

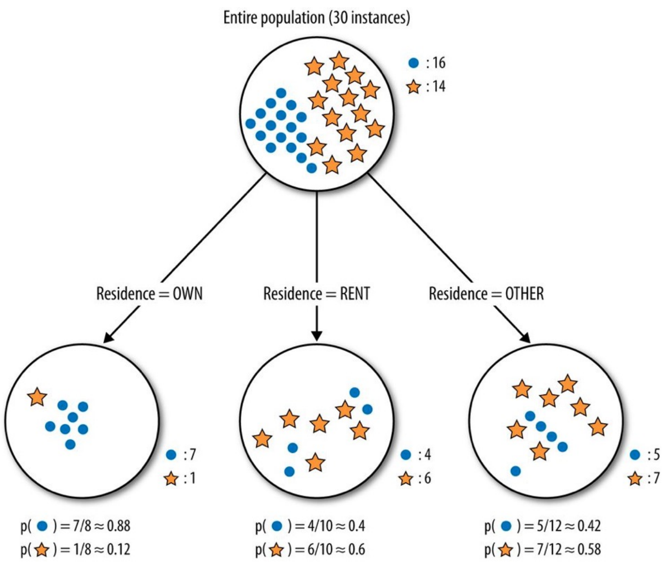
$$E(Balance) = \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79 = 0.62$$

Information Gain:

$$\begin{aligned} IG(Parent, Balance) &= E(Parent) - E(Balance) \\ &= 0.99 - 0.62 \\ &= 0.37 \end{aligned}$$

*Courtesy of: https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Motivating Example (simple decision tree) - Part 2



$$E(\,Residence\ =\ OWN)\ =\ -\frac{7}{8}\log_2\left(\frac{7}{8}\right)-\ \frac{1}{8}\log_2\left(\frac{1}{8}\right)\ \approx 0.54$$

$$E(\,Residence\ =\ RENT\,)\ =\ -\frac{4}{10}\log_2\left(\frac{4}{10}\right)-\ \frac{6}{10}\log_2\left(\frac{6}{10}\right)\ \approx 0.97$$

$$E(\,Residence\ =\ OTHER)\ =\ -\frac{5}{12}\log_2\left(\frac{5}{12}\right)-\ \frac{7}{12}\log_2\left(\frac{7}{12}\right)\ \approx 0.98$$

Weighted Average of entropies for each node:

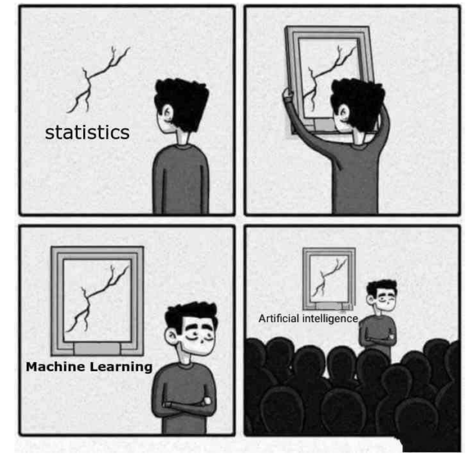
$$E(\,Residence)\ =\frac{8}{30}\times 0.54\ +\ \frac{10}{30}\times 0.97\ +\frac{12}{30}\times 0.98\ =\ 0.86$$

Information Gain:

$$\begin{aligned} IG(\,Parent,\ Residence) &= E(\,Parent)\ -\ E(\,Residence) \\ &= 0.99\ -\ 0.86 \\ &= 0.13 \end{aligned}$$

*Courtesy of: https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Is that it?



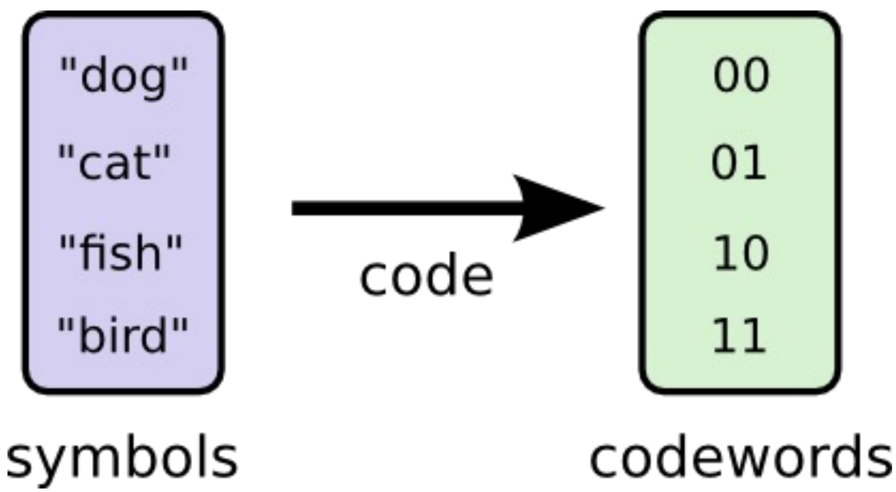
- So is entropy a measure of node purity?
  - Yes and no.
- Entropy and related quantities are pervasive in every area of
  - machine learning
  - information theory
  - compression
  - Etc.
- Believe it or not you already used it!
  - Where?

Entropy 101

- Intuitively, Entropy is a measure of “disorder”.
  - The higher the entropy the more disorder in the system.
- Let’s define this in terms of codes:

This is your R.V.

This is your “encoded” word



Assume every word is equally likely.

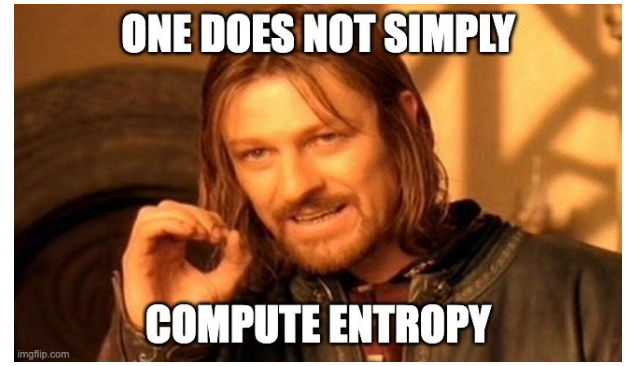
What is the number of bits you need
to represent every “message” or “random variable”?

Answer: 2. Why?

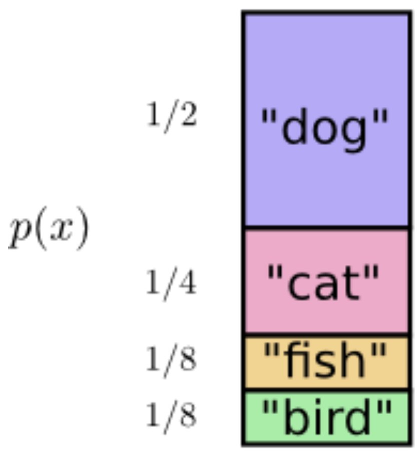
*Courtesy of: https://colah.github.io/posts/2015-09-Visual-Information/

Entropy 101

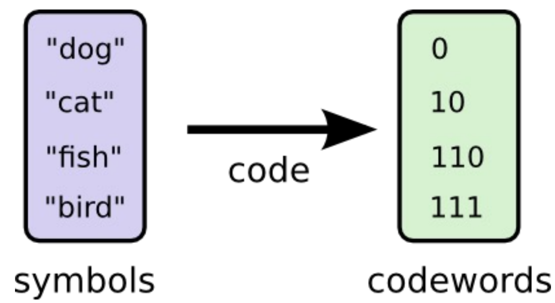
- But can we do better?
  - Not if we use the simple code.
- But what if every message was not equally likely.
  You're much more likely to say "dog" than "bird" in your made up world.



Well if this is how you communicate you might want a short hand.



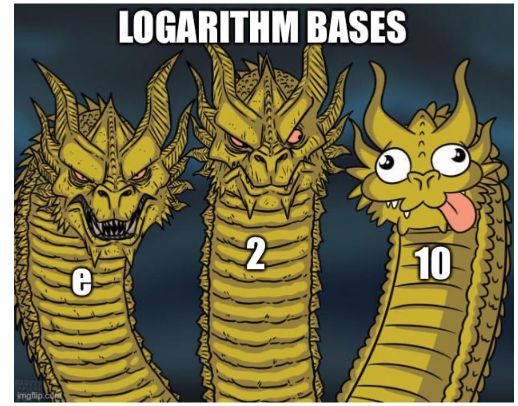
Dog Lover's
Word Frequency



Question: Note that once you use "0" for "dog"
Why can't you use "01" for "cat"?

*Courtesy of: https://colah.github.io/posts/2015-09-Visual-Information/

Entropy 101



- What's the entropy here?

$$H(p) = -\left(\frac{1}{2} * \log_2 \frac{1}{2} + \frac{1}{4} * \log_2 \frac{1}{4} + \frac{1}{8} * \log_2 \left(\frac{1}{8}\right) + \frac{1}{8} * \log_2 \frac{1}{8}\right) = 1.75$$

- We need *LESS* bits to represent the typical "average" message. Wow!

Entropy 101

Learnings so far:

- If we use a “variable length” code we can decrease the average number of bits use by shortening the code used for more likely words!

- “Entropy” is a measure of disorder OR average number of bits needed.
  - What happens if one message becomes very very likely? Like with probability .99? .999? .9999….
  - The disorder goes down. The entropy goes down. The average number of bits needed to communicate the message goes down.
  - In fact, if I know EXACTLY what you’re going to say, why even say it? You don’t need to. The number of bits needed to communicate is 0. (!). The entropy is 0.
  - For example: For instance, if you’re entire vocabulary consisted of spaceships that made the “kessel run in less than 12 parsecs”

P(x) = 1

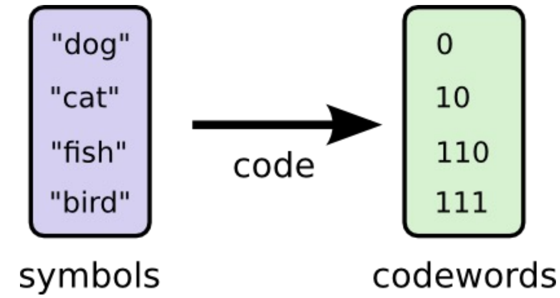
P(x) = 0

P(x) = 0

….

$$H(p) = -(1 * \log_2 1) = 0$$

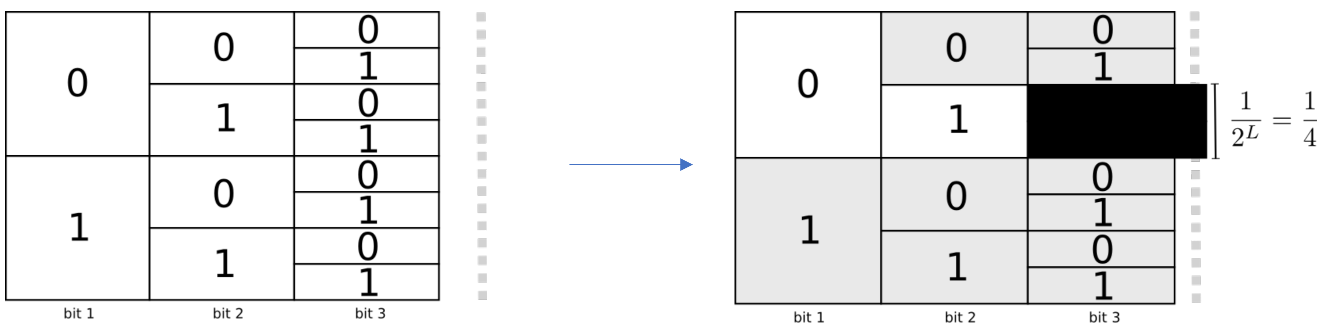
Entropy 101



- Ok great, but how did you come up with the length of the codeword?

  See here: https://colah.github.io/posts/2015-09-Visual-Information

  - The intuition though is fairly straightforward:

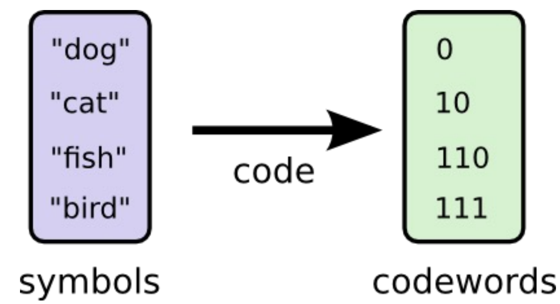


So by using a codeword "01" we block off ¼ of all possible future code words.

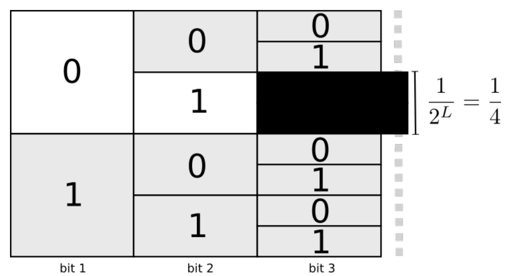
- So what do we do? We match the length of the word to the "cost"

$$Cost(L) = \frac{1}{2^L} \rightarrow 2^L = \frac{1}{cost} \rightarrow \log_2 2^L = \log_2 \frac{1}{cost} \rightarrow L\, Log_2(2) = \log_2\left(\frac{1}{cost}\right) \rightarrow L = Log_2\left(\frac{1}{cost}\right)$$

Entropy 101



- So $L(X) = \log_2 \frac{1}{Cost}$
- But how much is the cost? Exactly the probability of all future words that are taken up!



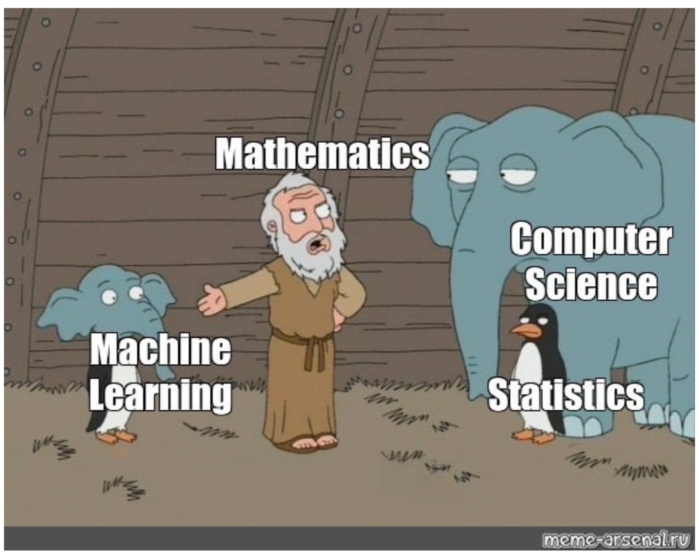
- So we rewrite $L(X) = \log_2 \frac{1}{p(x)}$

Exactly what we started with!

- And Entropy is the AVERAGE Length: Literally $H(p) = \Sigma p(x) * L(x) = \Sigma p(x) * log_2\left(\frac{1}{p(x)}\right) = -\Sigma \mathrm{p(x)} * \log_2(\mathrm{p(x)})$

Entropy 201

- Ok, ok. Entropy is neat. I can do cute things like reduce the number of bits needed to send a message.
- I can even play with decision trees.
- But this is a class on Data Mining and Machine Learning. Why should I care?

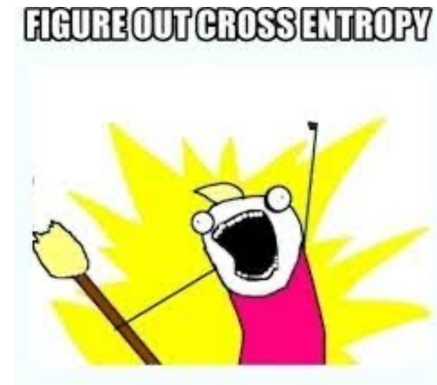


That's why!

Actually, there's some concepts like cross-entropy, KL-divergence and mutual information that forms the basis of most optimization algorithms.

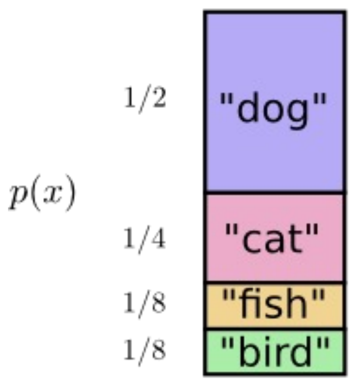
Let's define what they are and then look at logistic regression and cross-entropy.

Cross-Entropy (Eg Entropy 201)

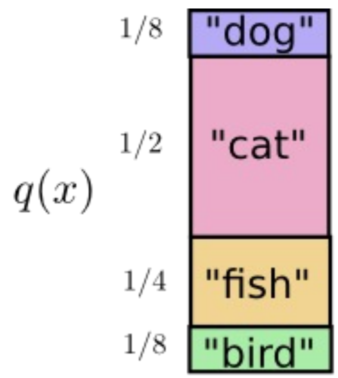


- Suppose now you have two potential people (Bob and Alice – because it's a CS class)
- They both speak the same language but Bob is a dog lover and Alice is a Cat lover.
  - Somehow they are still in a relationship.
  - AND they are still talking to each other.

Optimal Code
0
10
110
111



Dog Lover's Word Frequency

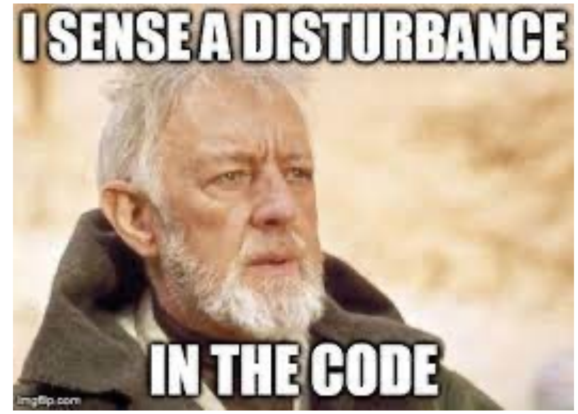


Cat Lover's Word Frequency

Optimal Code
111
0
10
110

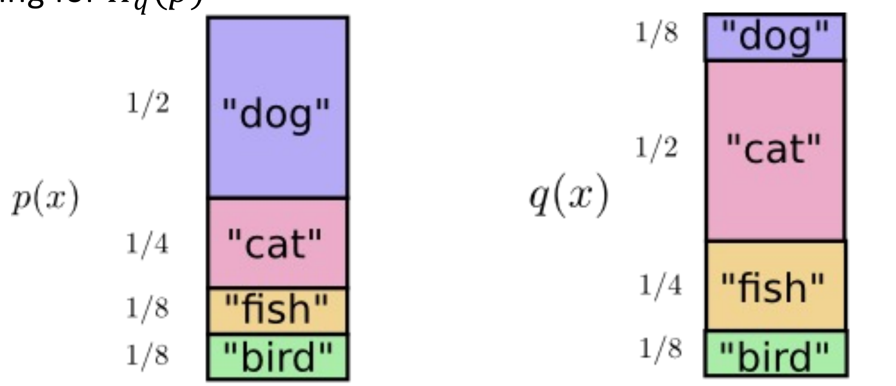
*Note, if we use Bob's probabilities on Bob's encoding (or Alice's probabilities on Alice's encoding) we get entropy of 1.75

Cross-Entropy (Eg Entropy 201)



- What if Bob now uses Alice's code?
  - Notation, let Bob's probabilities be "p" and Alice's probabilities be "q".
  - This means "111" is used to represent "dog" which occurs ½ the time!
  - We're looking for $H_q(p)$

Optimal Code
0
10
110
111



Dog Lover's Word Frequency

Cat Lover's Word Frequency

Optimal Code
111
0
10
110

$H(p,q)$

$$H_q(p) = -\left(\frac{1}{2} * \log_2\left(\frac{1}{8}\right) + \frac{1}{4} * \log_2\left(\frac{1}{2}\right) + \frac{1}{8} * \log_2\left(\frac{1}{4}\right) + \frac{1}{8} * \log_2\left(\frac{1}{8}\right)\right) = 2.375$$

Similarly, $H_p(q) = 2.25$ (try this on your own!)

Cross-Entropy (Eg Entropy 201)

- Note: $H_q(p) \neq H_q(p)$.
  - Cross-entropy is NOT symmetric. Some codes are better than others.

- Enter (stage left) “Kullback–Leibler divergence” or just KL divergence.
- $D_{KL}(P \mid\mid Q) = D_q(p) = H_q(p) - H(p)$
  - Mathematically equivalent to: $\mathrm{D}_q(p) = \ \Sigma p_{\mathrm{i}} \ * \log\left(\frac{\mathrm{p_i}}{q_i}\right)$
    - Log of the ratio of the two probabilities and weighted by the probability of the reference distribution.
- Reformulating $H_q(p) = D_{\mathrm{q}}(\mathrm{p}) + \mathrm{H(p)}$
  - The cross entropy decomposes into the optimal encoding plus how many additional bits needed to account for the fact that you’re using the wrong length code!
  - Or put another way, it’s a non-symmetric measure of how apart two distributions are from each other.

- This is used everywhere where you want to minimize the the distance between two distributions?
  - Your neural net predicts that picture X is 85% dog, 10% cat, 5% fish. You want to match it up to 100% dog, 0% cat, 0% fish.

Mutual Information (Entropy 301)

- Ok, but what if I don't so much are about the difference in distributions but how much one variable tells me of another?
  - Shall we just use correlation? Well No. Why?
  - Enter Information. The better modern day way to measure variable relationships!

How did I get this?

Let's suppose I have two NON-independent variables

X	Y	Pr(X)	Pr(Y)	Pr(X&Y)	Pr(X|Y)
T-shirt	raining	.62	.25	.06	=.06/.25 =.24
	sunny		.75	.56	=.56/.75 = .75
Coat	raining	.38	.25	.19	=.19/.25 = .76
	sunny		.75	.19	=.19/.75 = .25

$$H(X\&Y) = \Sigma_{x,y} p(x,y) \times \log_2 \frac{1}{p(x,y)} = .06 * \log\left(\frac{1}{.06}\right) + .56 * \log\left(\frac{1}{.56}\right) + .19 * \log\left(\frac{1}{.19}\right) + .19 * \log\left(\frac{1}{.19}\right) = 1.62$$

$$H(X|Y) = \Sigma_{x,y} p(x,y) \times \log_2 \frac{1}{p(x|y)} = .06 * \log\left(\frac{1}{.24}\right) + .56 * \log\left(\frac{1}{.75}\right) + .19 * \log\left(\frac{1}{.76}\right) + .19 * \log\left(\frac{1}{.25}\right) = .81$$

$$H(Y) = \Sigma_y p(y) \times \log_2 \frac{1}{p(y)} = .25 * \log\left(\frac{1}{.25}\right) + .75 * \log\left(\frac{1}{.75}\right) = .81$$

$$H(X) = \Sigma_x p(x) \times \log_2 \frac{1}{p(x)} = .62 * \log\left(\frac{1}{.62}\right) + .38 * \log\left(\frac{1}{.38}\right) = .96$$

Mutual Information (Entropy 301)

- Now we get Information:
- $I(X,Y)$ = H(X) + H(Y) - H(X,Y) = .81 + .96 – 1.62 = 0.15

.15 bits of extra information when both variables "considered together.

Let's suppose I have two independent variables.

X	Y	Pr(X)	Pr(Y)	Pr(X&Y)	Pr(X|Y)
T-shirt	raining	.62	.25	.155	.62
	sunny		.75	.465	.62
Coat	raining	.38	.25	.095	.38
	sunny		.75	.285	.38

$$H(X\&Y) = \Sigma_{x,y} p(x,y) \times \log_2 \frac{1}{p(x,y)} = .155 \log\left(\frac{1}{.155}\right) + .465 \log\left(\frac{1}{.465}\right) + .095 \log\left(\frac{1}{.095}\right) + .285 \log\left(\frac{1}{.285}\right) = 1.77$$

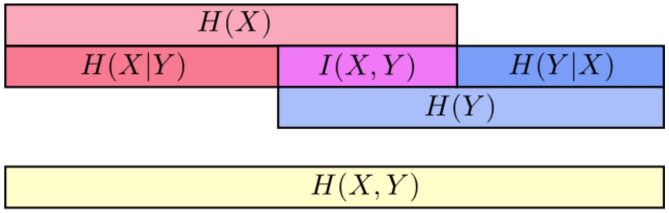
$$H(Y) = \Sigma_y p(y) \times \log_2 \frac{1}{p(y)} = .25 * \log\left(\frac{1}{.25}\right) + .75 * \log\left(\frac{1}{.75}\right) = .81$$

$$H(X) = \Sigma_x p(x) \times \log_2 \frac{1}{p(x)} = .62 * \log\left(\frac{1}{.62}\right) + .38 * \log\left(\frac{1}{.38}\right) = .96$$

- $I(X,Y)$ = H(X) + H(Y) - H(X,Y) = .81 + .96 – 1.77 = 0

LOOK! No extra information by looking at both Variables together under indepence.

All in one picture!



Two additional great references:

https://tungmphung.com/information-theory-concepts-entropy-mutual-information-kl-divergence-and-more/

https://gaussian37.github.io/ml-concept-infomation_theory/

http://www.ece.tufts.edu/ee/194NIT/lect01.pdf

http://www.scholarpedia.org/article/Mutual_information

Time for a logistic regression example? Time to A/B test!