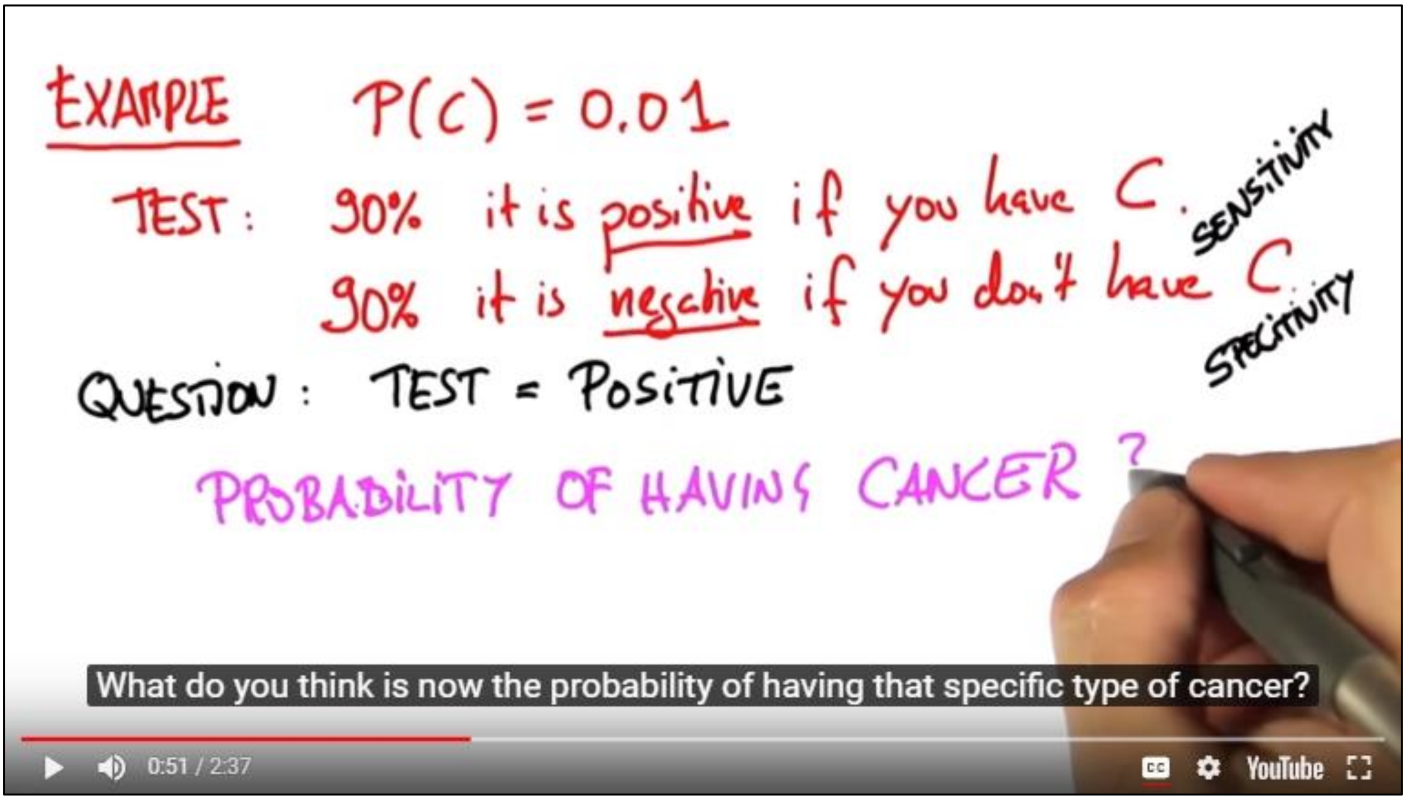
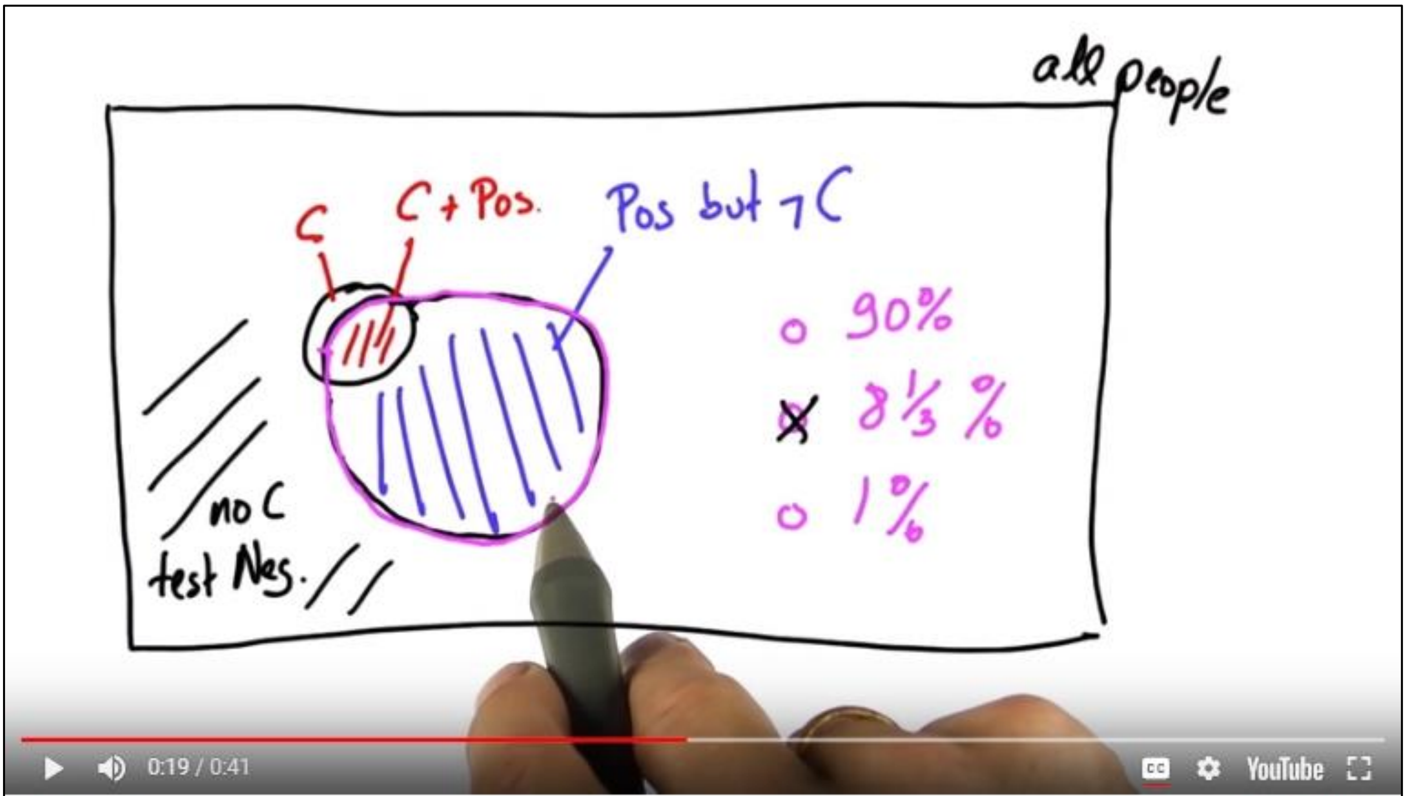
Naïve Bayes

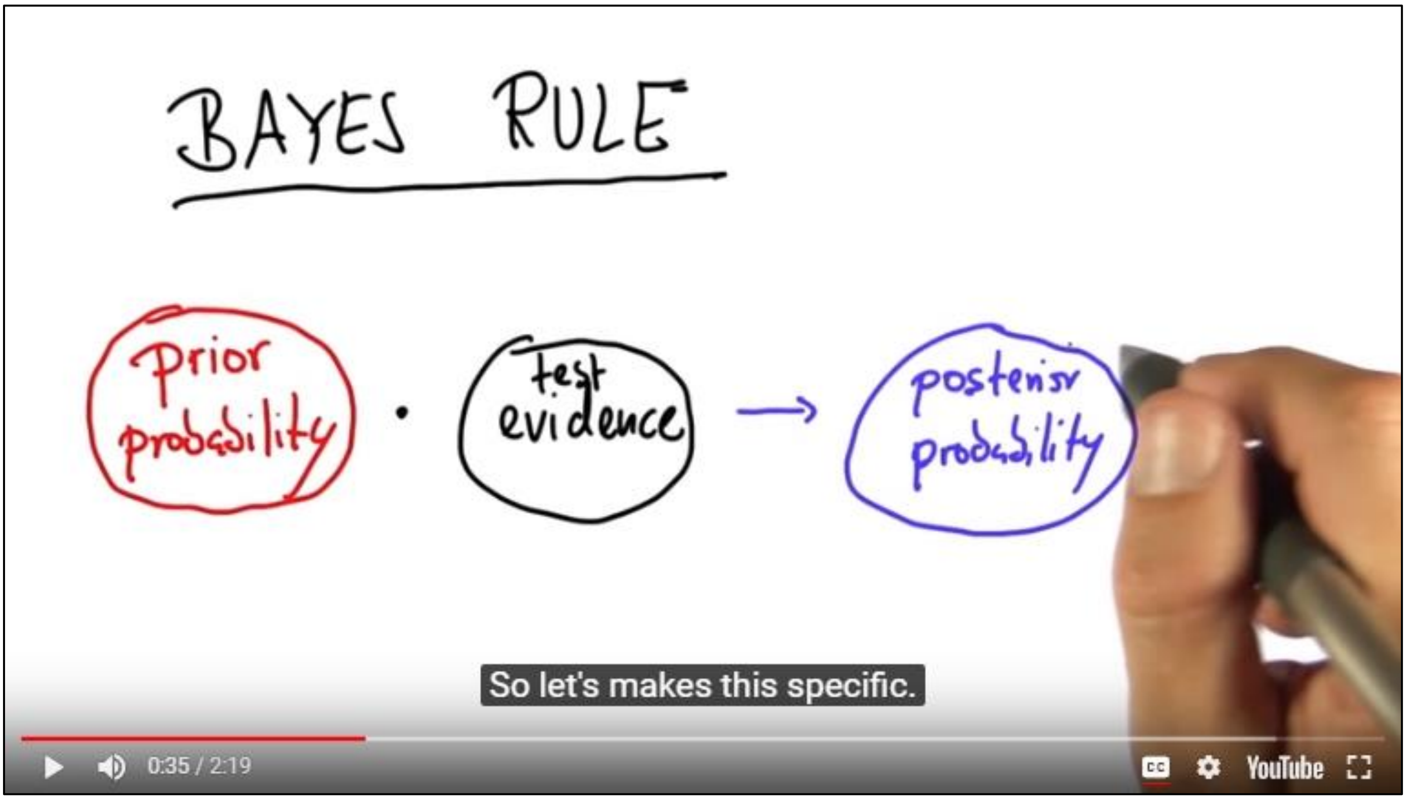


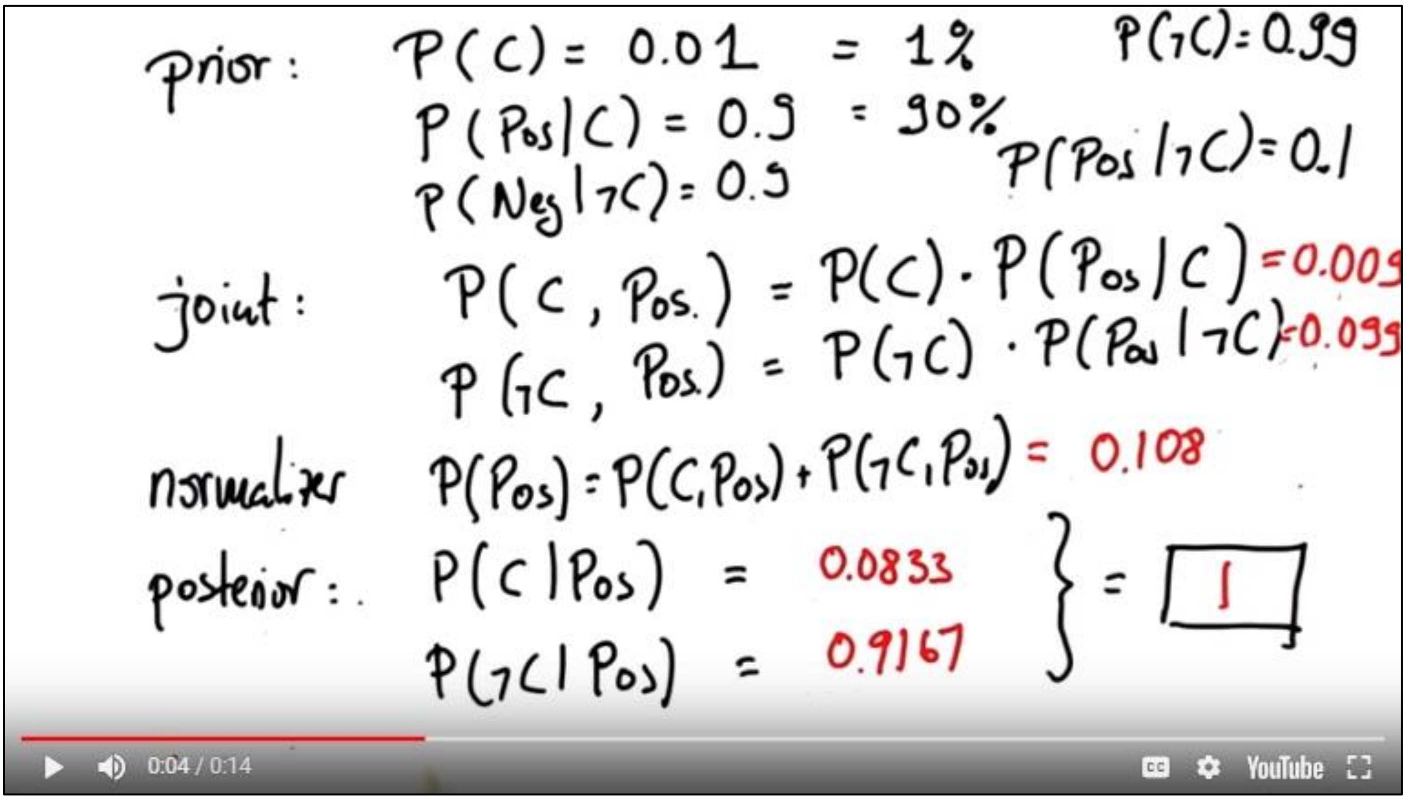


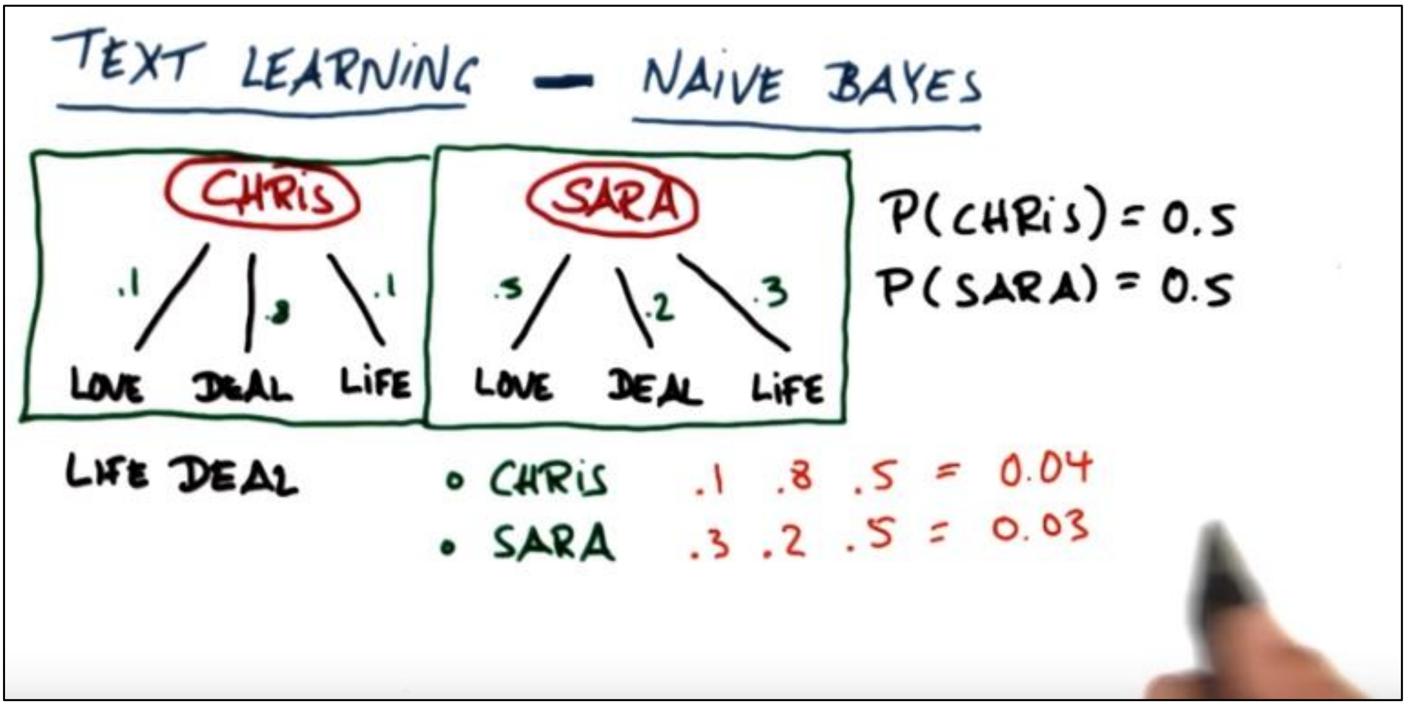
VIEW INTRO

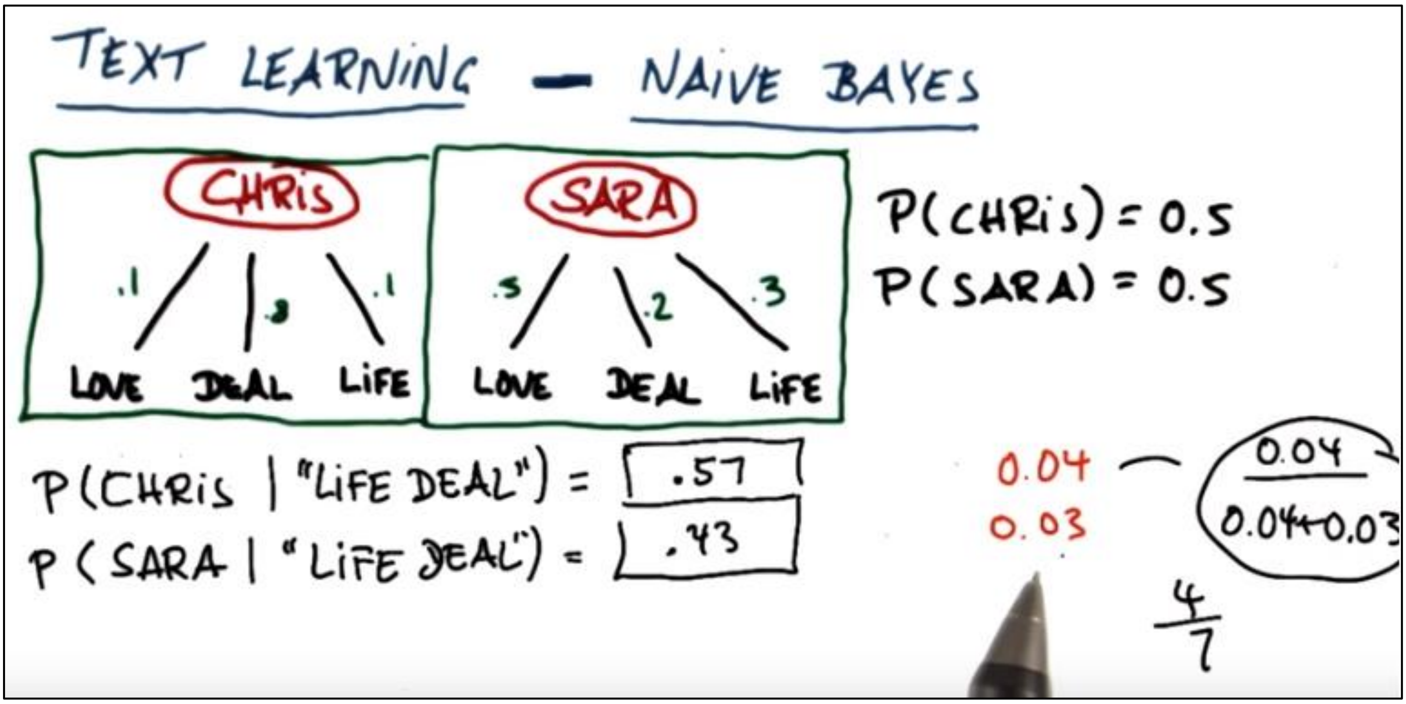
BACK TO QUIZ

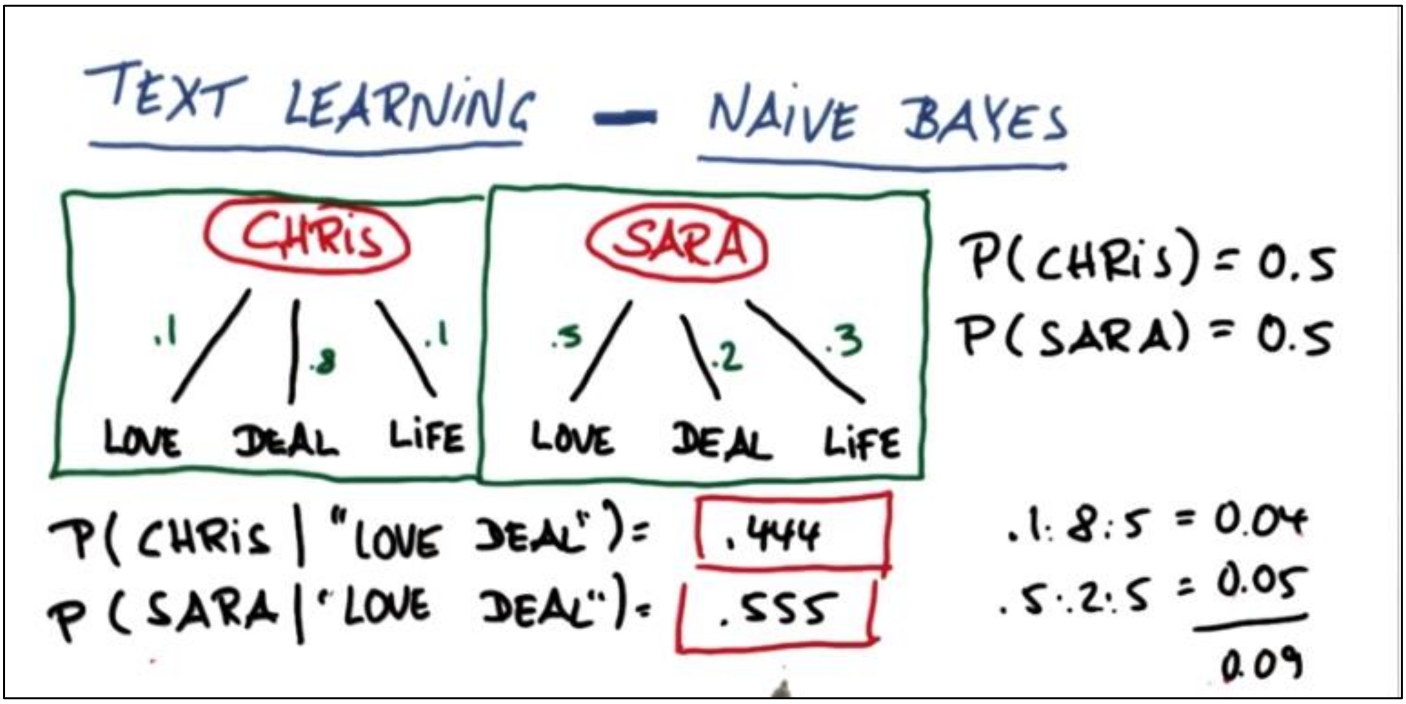
Reminder of the question: the prior probability of cancer is 1%, and a sensitivity and specificity are 90%, what's the probability that someone with a positive cancer test actually has the disease?


BAYES RULE
prior probability
test evidence
posterior probability
So let's makes this specific.
0:35 / 2:19
YouTube

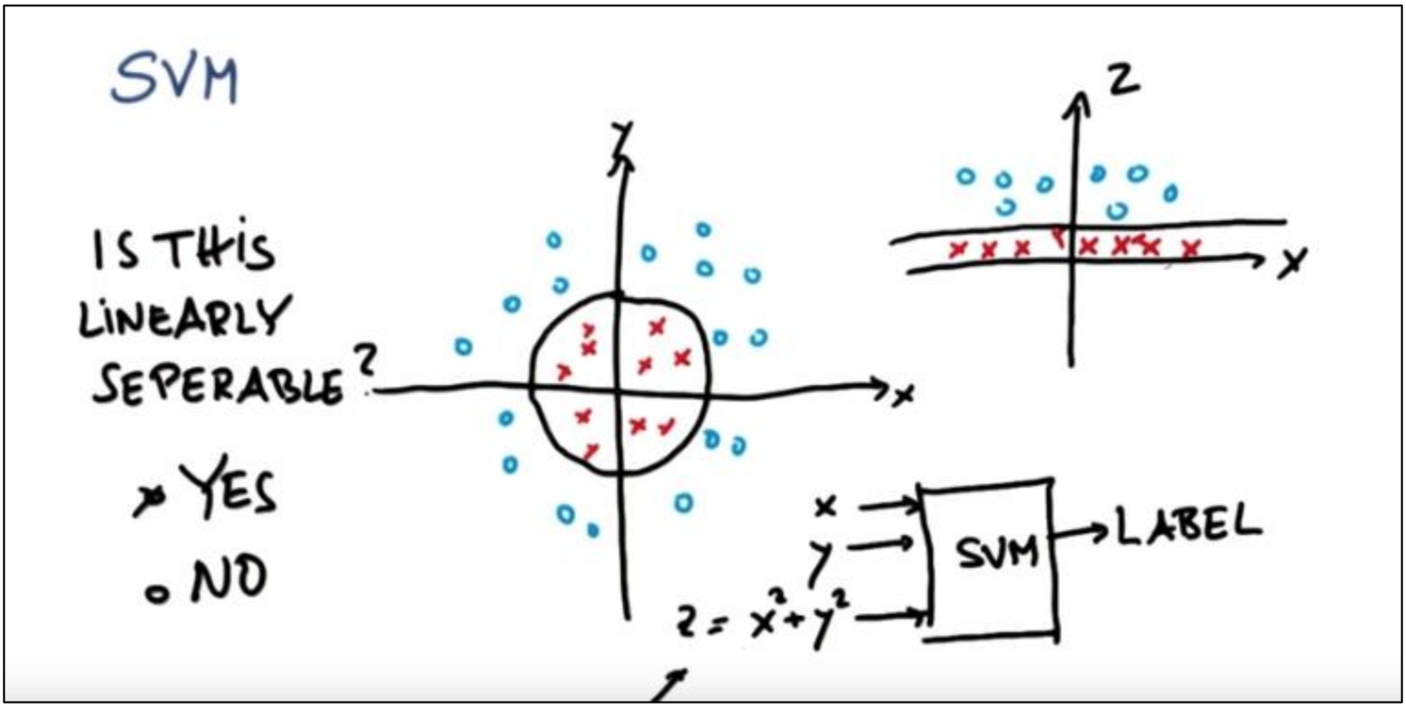

prior: P(C) = 0.01 = 1%
P(¬C) = 0.99
P(Pos|C) = 0.9 = 90%
P(Pos|¬C) = 0.1
P(Neg|¬C) = 0.9
joint: P(C, Pos.) = P(C) · P(Pos|C) = 0.00
P(¬C, Pos.) = P(¬C) · P(Pos|¬C) = 0.09
normalizer P(Pos) = P(C, Pos) + P(¬C, Pos) = 0.108
posterior: P(C|Pos) = 0.0833
P(¬C|Pos) = 0.9167
= 1
0:04 / 0:14
YouTube

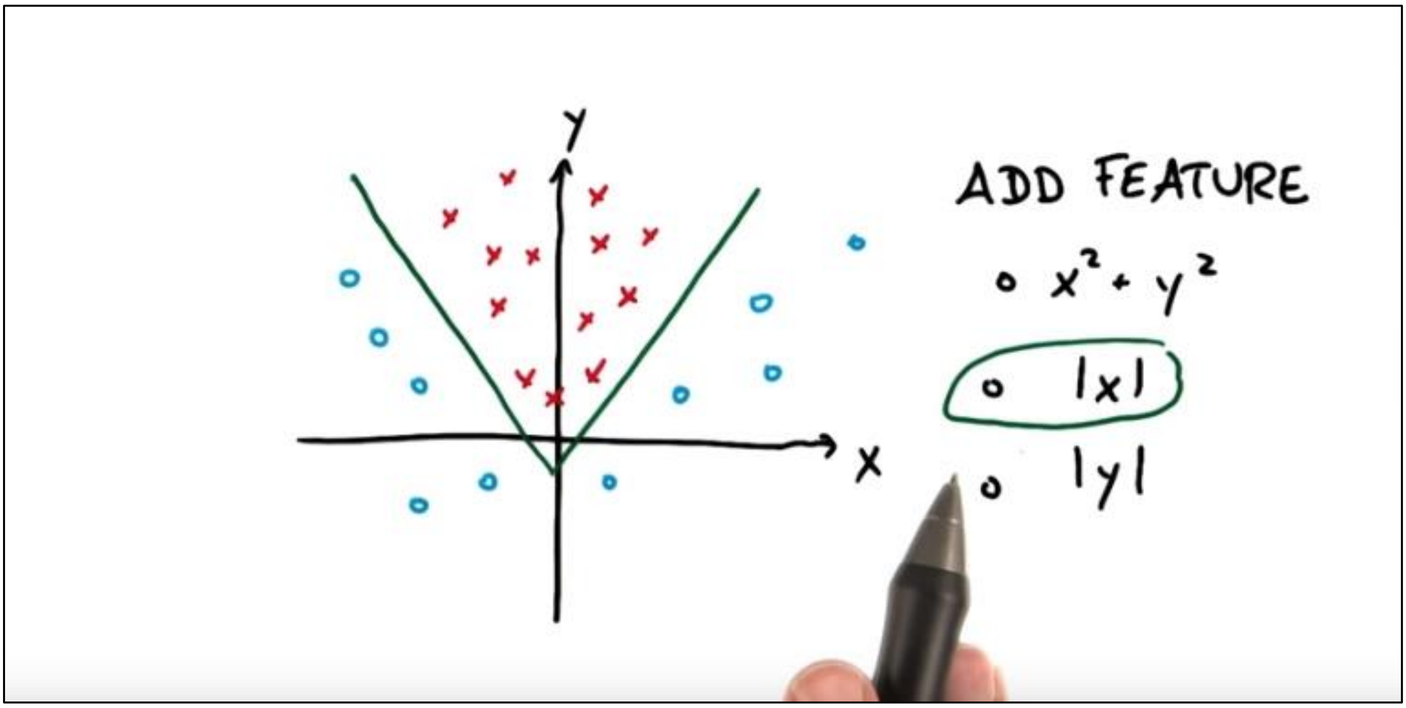

TEXT LEARNING — NAIVE BAYES
CHRIS
.1
.8
.1
LOVE
DEAL
LIFE
SARA
.5
.2
.3
LOVE
DEAL
LIFE
P(CHRIS) = 0.5
P(SARA) = 0.5
LIFE DEAL
• CHRIS .1 .8 .5 = 0.04
• SARA .3 .2 .5 = 0.03

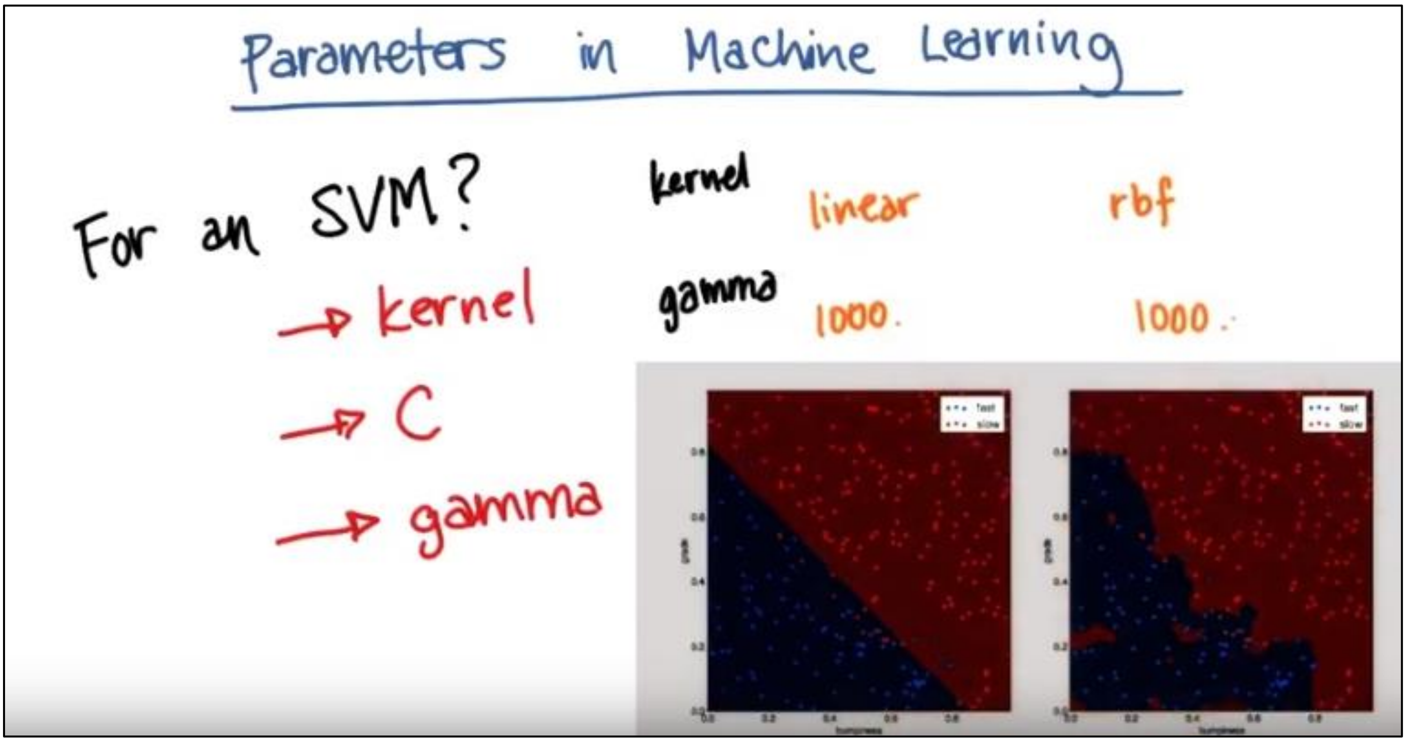

TEXT LEARNING — NAIVE BAYES
CHRIS
.1
.8
.1
LOVE
DEAL
LIFE
SARA
.5
.2
.3
LOVE
DEAL
LIFE
P(CHRIS) = 0.5
P(SARA) = 0.5
P(CHRIS | "LIFE DEAL") = .57
P(SARA | "LIFE DEAL") = .43
0.04
0.03
0.04
0.04+0.03
4
7


TEXT LEARNING — NAIVE BAYES
CHRIS
SARA
.1
.8
.1
.5
.2
.3
LOVE DEAL LIFE
LOVE DEAL LIFE
P(CHRIS) = 0.5
P(SARA) = 0.5
P(CHRIS | "LOVE DEAL") = .444
P(SARA | "LOVE DEAL") = .555
.1·.8·.5 = 0.04
.5·.2·.5 = 0.05
0.09

SVM

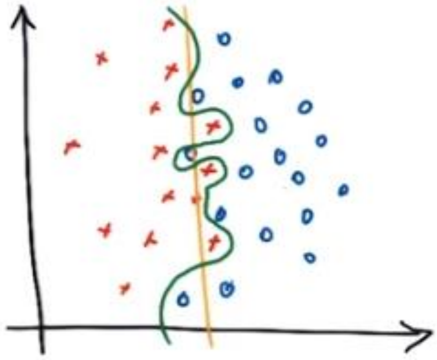

SVM
IS THIS LINEARLY SEPERABLE?
× YES
○ NO
y
x
z
x
x
y
z = x² + y²
SVM
LABEL


y
x
ADD FEATURE
$x^2 + y^2$
$|x|$
$|y|$


Parameters in Machine Learning
For an SVM?
→ kernel
→ C
→ gamma
kernel
linear
rbf
gamma
1000.
1000.

SVM C Parameter

C – controls tradeoff between smooth decision boundary and classifying training points correctly



Quiz

does a large C mean you expect a smooth boundary, or that you will get more training points correct?

- smooth boundary
- more training points correct

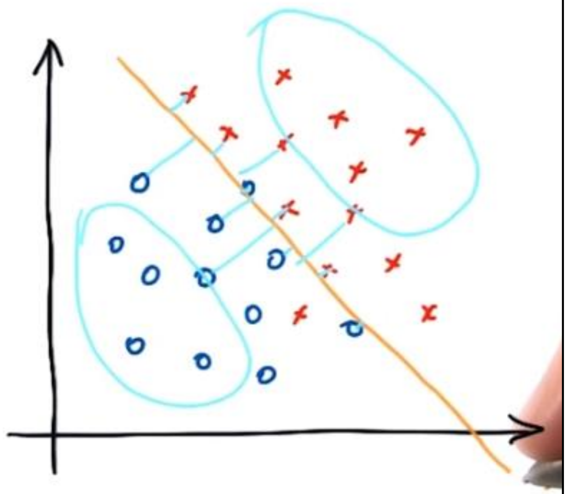
use rbf kernel!

SVM γ (gamma) parameter

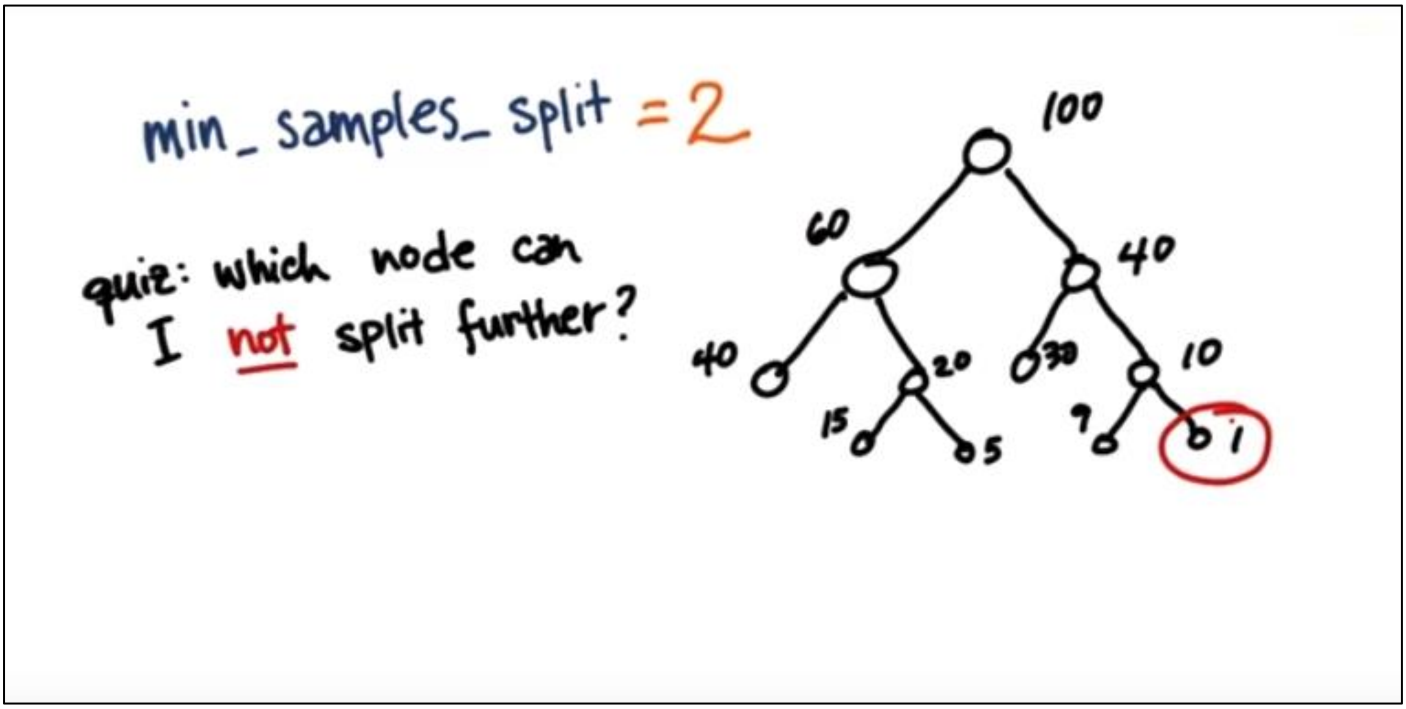
γ – defines how far the influence of a single training example reaches

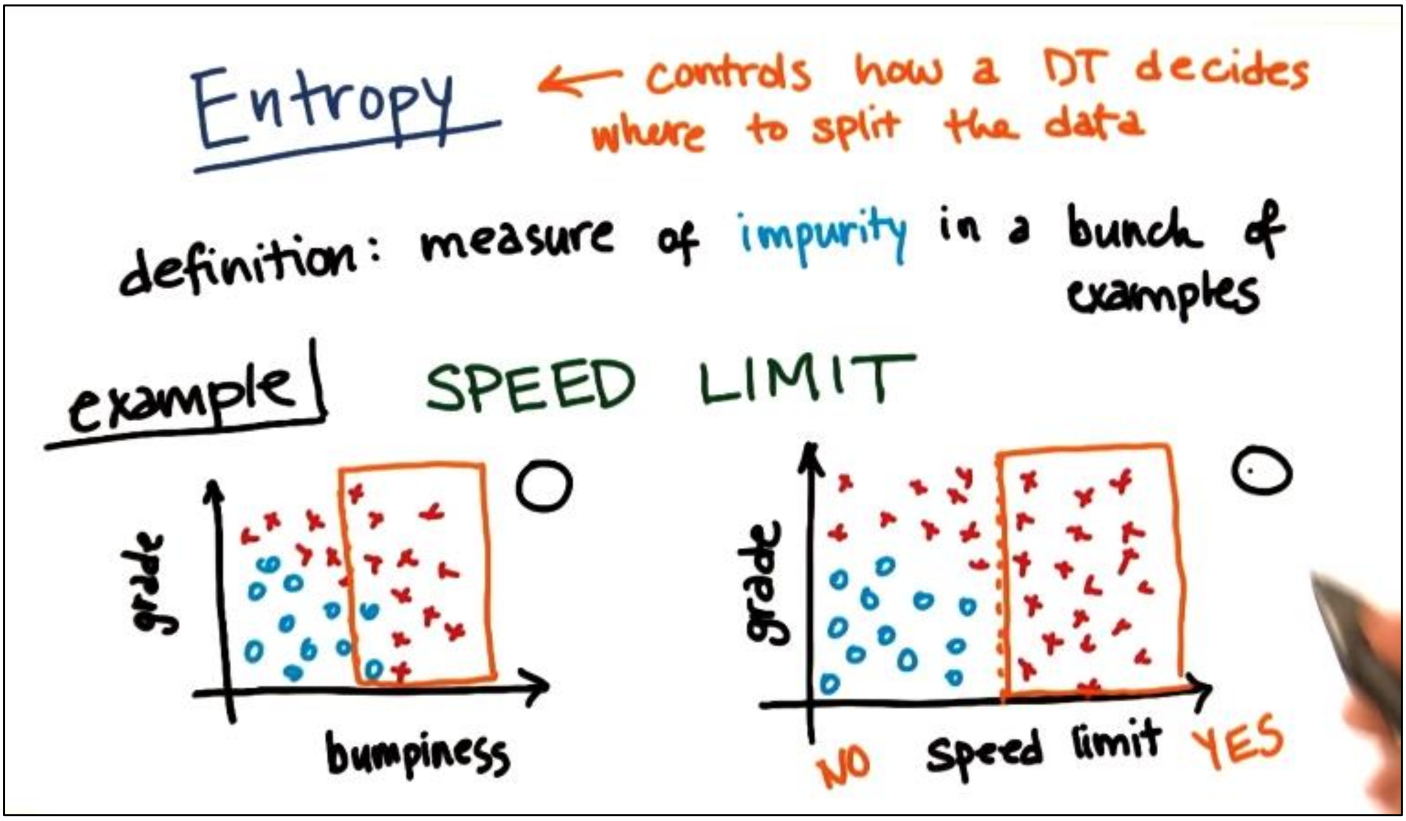
low values – far

high values – close



Decision Tree





Entropy

← controls how a DT decides where to split the data

definition: measure of impurity in a bunch of examples

$$\text{entropy} = \sum_i -p_i \log_2 (p_i)$$

sum over all classes

fraction of examples in class i

Entropy

← controls how a DT decides where to split the data

definition: measure of impurity in a bunch of examples

intution

all examples are same class

→ entropy = 0

examples are evenly split between classes

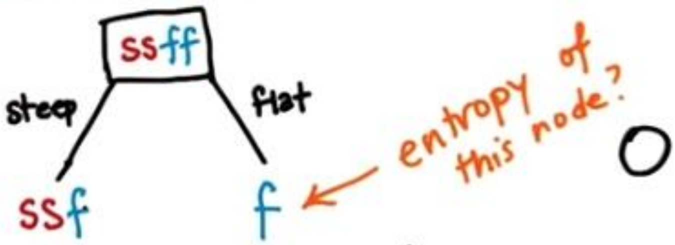
→ entropy = 1.0

Information Gain

information gain = entropy (parent) − [weighted average] entropy (children)

grade	bumpiness	speed limit	speed
steep	bumpy	yes	slow
steep	smooth	yes	slow
flat	bumpy	no	fast
steep	smooth	no	fast

entropy of parent = 1.0



$P_{slow} = 2/3$ $P_{fast} = 1/3$

Information Gain

entropy (flat) = 0

$P_{slow} = 2/3$ $P_{fast} = 1/3$

information gain = entropy (parent) − [weighted average] entropy (children)

grade	bumpiness	speed limit	speed
steep	bumpy	yes	slow
steep	smooth	yes	slow
flat	bumpy	no	fast
steep	smooth	no	fast

entropy of parent = 1.0

$$\text{entropy} = \sum_i -P_i \log_2 P_i$$

$$\text{entropy} = \left[-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right]$$

$$= 0.9184$$

Algorithms (Pick One)

k nearest neighbors — classic, simple, easy to understand

adaboost

random forest

"ensemble methods"
meta classifiers built from (usually) decision trees