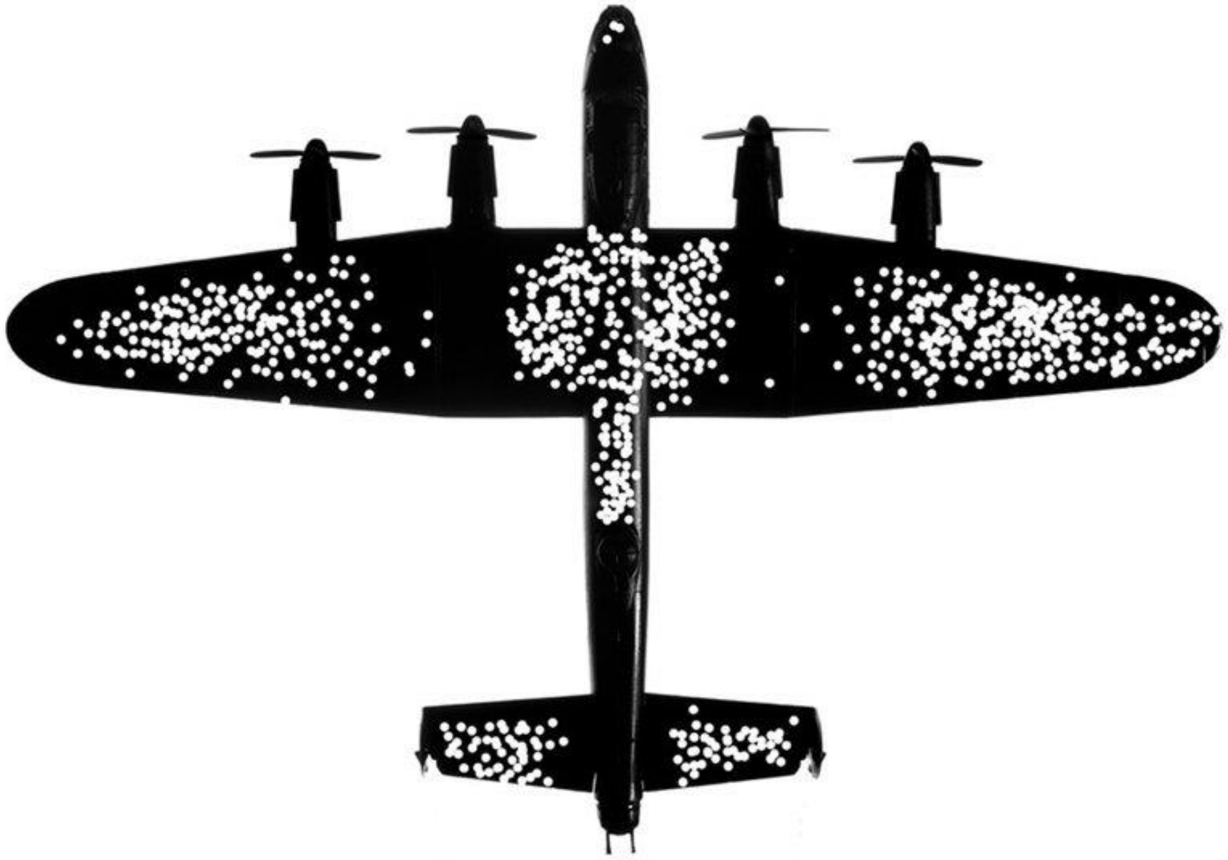
CS 412



JAN 14^{TH} – INTRO TO ML

Machine Learning

What is Machine Learning?

Machine Learning

What is Machine Learning?

- An algorithm which **improves itself** with exposure to data

Machine Learning

What is Machine Learning?

- An algorithm which **improves itself** with exposure to data

Why do we care?

Machine Learning

What is Machine Learning?

- An algorithm which **improves itself** with exposure to data

Why do we care?

- It’s a big buzzword right now, but that shouldn’t be enough

Machine Learning

What is Machine Learning?

- An algorithm which **improves itself** with exposure to data

Why do we care?

- It’s a big buzzword right now, but that shouldn’t be enough
- ML algorithms build a model of data and then make predictions based on that model
- It is as much about the **process** as it is about the model that results

Machine Learning

Problems come in many different varieties

Machine Learning

Problems come in many different varieties

- *Supervised—*
- *Unsupervised –*

Machine Learning

Problems come in many different varieties

- *Supervised*—
  - The algorithm builds a model based on given training data
  - Easier to get an understanding of how *good* a model is
- *Unsupervised* –

Machine Learning

Problems come in many different varieties

- *Supervised—*

  The algorithm builds a model based on given training data

  Easier to get an understanding of how *good* a model is

- *Unsupervised –*

  There is no *known* value for data points, the model just tries

  to build trends and see patterns

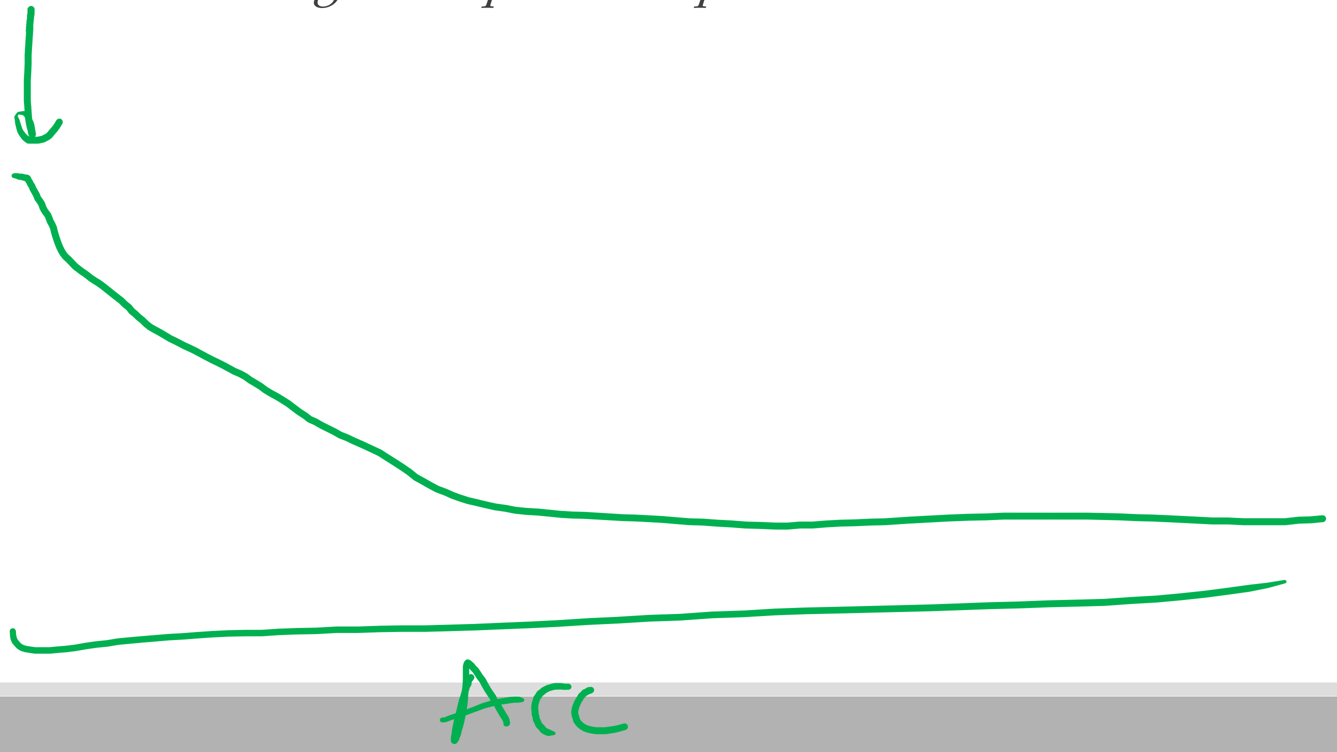
Machine Learning

What does good machine learning look like?

Machine Learning

What does good machine learning look like?

- *Data driven*
- *Statistically validated*
- *Understanding of the problem space*



Machine Learning

What does good machine learning look like?

- *Data driven*
- *Statistically validated*
- *Understanding of the problem space*

Why do we care?

Machine Learning

What does good machine learning look like?

- *Data driven*
- *Statistically validated*
- *Understanding of the problem space*

Why do we care?

- Very easy to do incorrectly
- “Off the shelf” ML
- Involved in increasingly important parts of our lives

About the course

USPS Mail Data
- “1”s and “5”s

5 assignments + one Neural network implementation
- (30% of your grade = 5% each + 5% for Neural network implementation)
- LaTeX/Jupyter
- Will contain both written and code portions
- Due Wednesdays – at least one week per assignment

Final project:
- Teams of up to 3, data of your choice

Language ”officially” supported: Python + Sci-kit learn + matplotlib
- Common/easy to use for data science/machine learning

About the course

Midterm
- TBA Next Week, Likely the week before spring break
- 20% of your grade

Final exam (20% of your grade)

Textbooks (both available online as pdf and are optional):
- *The Elements of Statistical Learning,* Hastie, Tibshirani.Friedman (HTF)
- *Hands on Machine learning with Scikit-learn and TensorFlow,* Aurélien Géron
- *(STAT381 review) Mathematical Statistics with Applications,* Wackerly, Mendenhall, Scheaffer

Most lectures will have a corresponding reading—most of them from the HTF book

About the course

Topics list:

- Supervised learning
  - Regression
  - Nearest-neighbors
  - Support Vector Machines
  - Neural networks
  - Ensemble methods
- Data handling
  - Pre-processing
  - Feature selection
  - Concentration bounds
  - Biased data
  - Noise/outlier detection
- Graphical Models
  - Markov graphs
  - Decision trees
  - Bayes networks
- Unsupervised learning
  - Clustering
  - k-Cover
- If-we-have-time topics
  - Reinforcement learning
  - Runtime improvements
  - Data visualization

About the course

Course administration:

- Piazza – you should have received an invitation yesterday
  - Syllabus with the official write up will be there
  - Unmarked slides will be posted there before lecture
  - Homework write-ups will be posted there
  - Great place for questions
- Gradescope – invitation will come out later this week
  - All homework submissions (PDF + code)
- Blackboard
  - The final gradebook will be posted there toward the end of the semester, but it will not be used until then
  - Lecture capture (will start next week)

About the course

Course administration:

- More next week

Expectations from me:

- Useful feedback back quickly
- Responsive to piazza posts
- Get materials to you in a timely fashion

About the course

Course administration:

- More next week

Expectations from me:

- Useful feedback back quickly
- Responsive to piazza posts
- Get materials to you in a timely fashion

Expectations from you:

- Ask questions (anonymous google form)
- Be attentive in class

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 1:

    Choose rock

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 1:
    Choose rock

What’s wrong with this algorithm?

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 1:
    Choose rock

What’s wrong with this algorithm?

- **It’s predictable**

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 1:
    Choose rock

What’s wrong with this algorithm?

- **It’s predictable**
- Is that a bad thing?

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 1:
Choose rock

What’s wrong with this algorithm?

- **It’s predictable**
- Is that a bad thing?

For this problem, yes! This is a competitive game and if the opponent has information about how you’re going to perform, they have an advantage

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 2:

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 2:

- Choose at random

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 2:

- Choose at random

What can we say about this approach?

Example

Let's play rock-paper-scissors (or more accurately, let's make an algorithm that plays)

Algorithm 2:

- Choose at random

What can we say about this approach?

- Hard-to-predict
- Should win 1/3 of the time (and draw 1/3 of the time)

Example

Let's play rock-paper-scissors (or more accurately, let's make an algorithm that plays)

Algorithm 2:

- Choose at random

What can we say about this approach?

- Hard-to-predict
- Should win 1/3 of the time (and draw 1/3 of the time)

Is there some improvement we could make? Hint: this is an ML course

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 2:

- Choose at random

What can we say about this approach?

- Hard-to-predict
- Should win 1/3 of the time (and draw 1/3 of the time)

Is there some improvement we could make? Hint: this is an ML course

- Doesn’t do any better against the “always rock” algorithm 1
- Should model the opponent behavior

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 3:

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 3:

- Keep a table of all the moves the opponent makes
  and choose the one that beats its most common choice

What does this algorithm try to exploit?

R	P	S
$\boxed{10}$	$\boxed{0}$	$\boxed{0}$
	a	a

Example

Let's play rock-paper-scissors (or more accurately, let's make an algorithm that plays)

Algorithm 3:

- Keep a table of all the moves the opponent makes
  and choose the one that beats its most common choice

What does this algorithm try to exploit?

- Maybe the opposing player has a favorite!
- What if the opposing player finds out that this is my strategy? (Always an advantage)

Example

Let's play rock-paper-scissors (or more accurately, let's make an algorithm that plays)

Algorithm 3:

- Keep a table of all the moves the opponent makes
  and choose the one that beats its most common choice

What does this algorithm try to exploit?

- Maybe the opposing player has a favorite!
- What if the opposing player finds out that this is my strategy? (Always an advantage)

*Let's reflect on the RPS question. What does this **problem** suppose?*

- The opposing player must have some level of predictability

Example

Let's play rock-paper-scissors (or more accurately, let's make an algorithm that plays)

Algorithm 4:

- Record all of the previous choices by the opposing player, shove it into a ML algorithm and then let the prediction variable be the next throw.

- Example:
- R -> S
- RS -> P
- RSP -> P
- RSPP -> S
- RSPPS -> R
- RSPPSR -> R

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 4:

- Record all of the previous choices by the opposing player, shove it into a ML algorithm and then let the prediction variable be the next throw.
- **No. Bad. Not by the end of this course, you won’t**
- Example:
- R -> S
- RS -> P
- RSP -> P
- RSPP -> S
- RSPPS -> R
- RSPPSR -> R

What’s wrong with this approach?

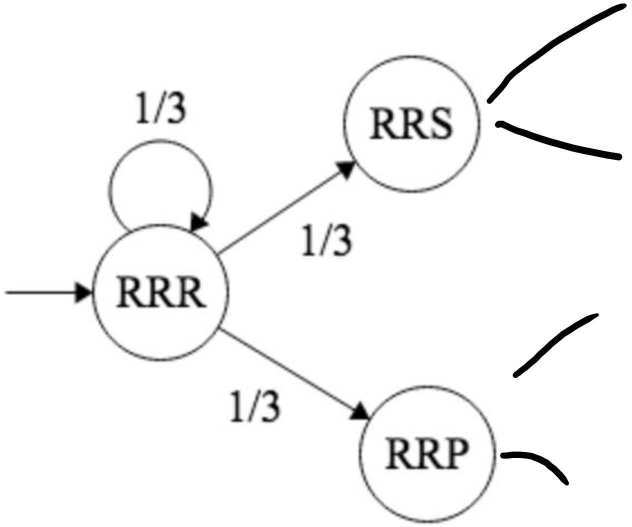
From a CS perspective, each piece of data has a differing width

Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 4:

- Keep a record of the last 3 throws, and see what the opposing player usually throws next



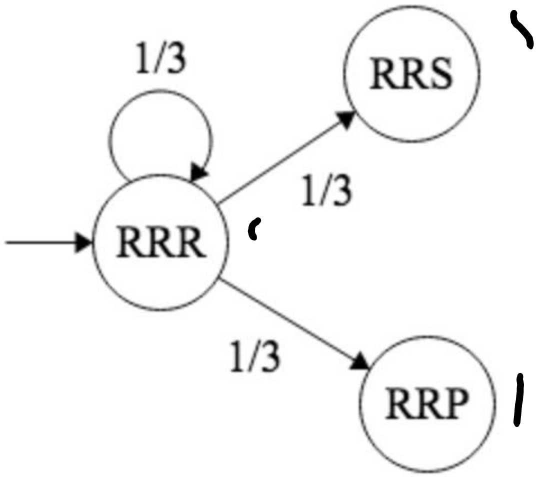
$3^3 = 27$

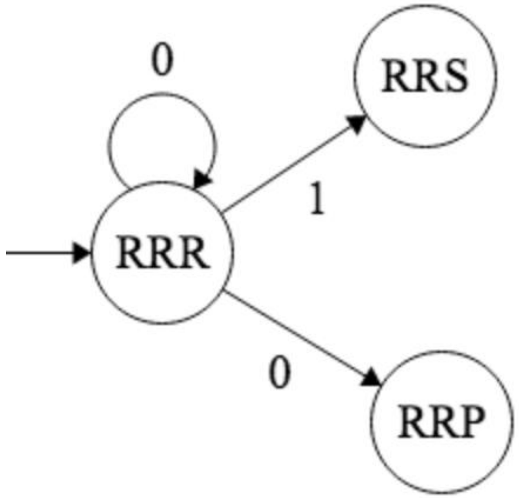
Example

Let’s play rock-paper-scissors (or more accurately, let’s make an algorithm that plays)

Algorithm 4:

- Keep a record of the last 3 throws, and see what the opposing player usually throws next





Example

This approach is called a Markov model

- Probability based graph, good at solving "turn-based" sort of prediction problems with discrete states
- Update the probabilities for each value as you play
  - Reinforcement + machine learning

How accurate is it?

- We can't quite quantify that yet

What other problems could we solve with this approach?

Example

ppsproject

This approach is called a Markov model

- Probability based graph, good at solving "turn-based" sort of prediction problems with discrete states
- Update the probabilities for each value as you play
  - Reinforcement + machine learning

How accurate is it?

- We can't quite quantify that yet

$30{,}000^n$

What other problems could we solve with this approach?

- Text prediction! (This was the go-to method until recently)

By the end of this course

You should be able to:

- Recognize a problem that could have an ML approach applied to it
- Try multiple ML models and select the one that best fits the data
- Accurately report on the quality of the given model

Big takeaway today?

- Human beings are predictable!

Probability and Statistics Review

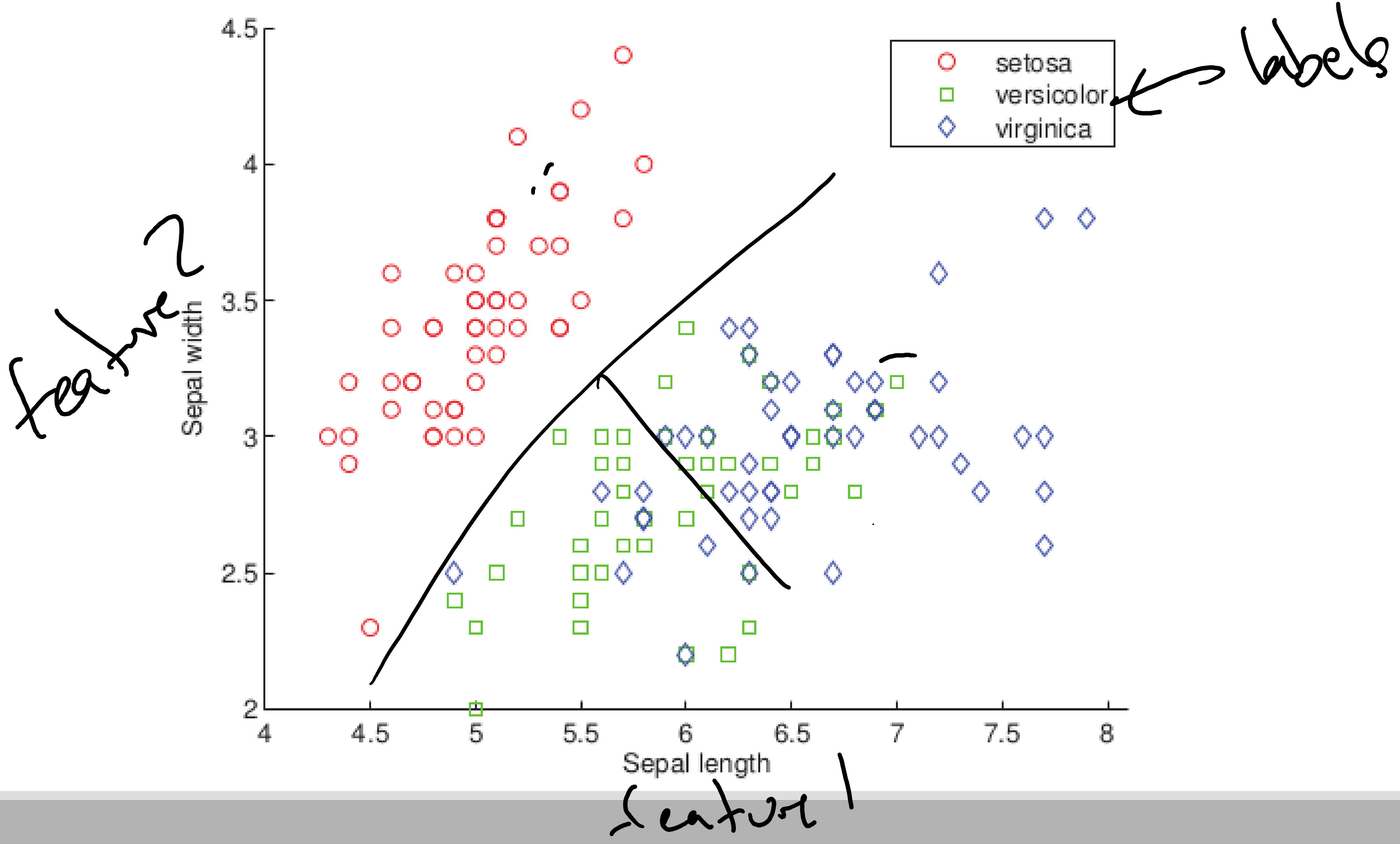
Why is probability important to ML?

- Assessing "luck" vs. significant improvements
- Expressing uncertainty in predictions is useful

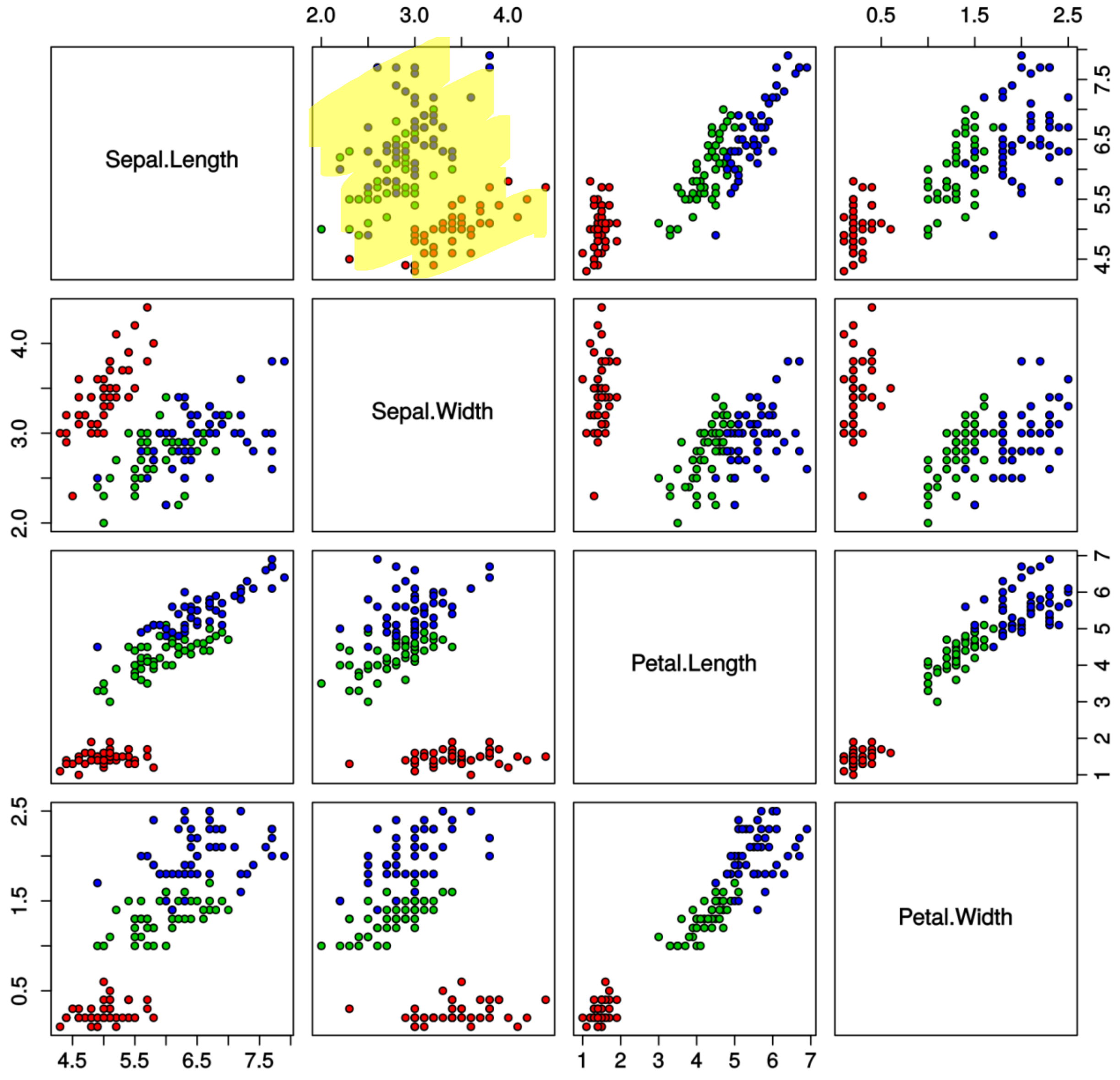
We want to make the decision that is **most likely** to be correct
--given the data that we have

- This is non-trivial
- There may not be an objective answer
- Most likely *according to which statistical model*

For more RPS code/examples (www.rpscontest.com)


4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8
Sepal width
Sepal length
setosa
versicolor
virginica
labels
feature 2
feature 1

Iris Data (red=setosa,green=versicolor,blue=virginica)



Random Variables

A variable with a value chosen by "chance"

For example:

X could be true or false

Y could be any real number

Z could be a vector of integers

$P(T) = 0.4 \quad P(F) = 0.6$

Random Variables

Discrete:

$P(x) \geq 0$

$\sum_x P(x) = 1$

Implies $P(x) \leq 1$

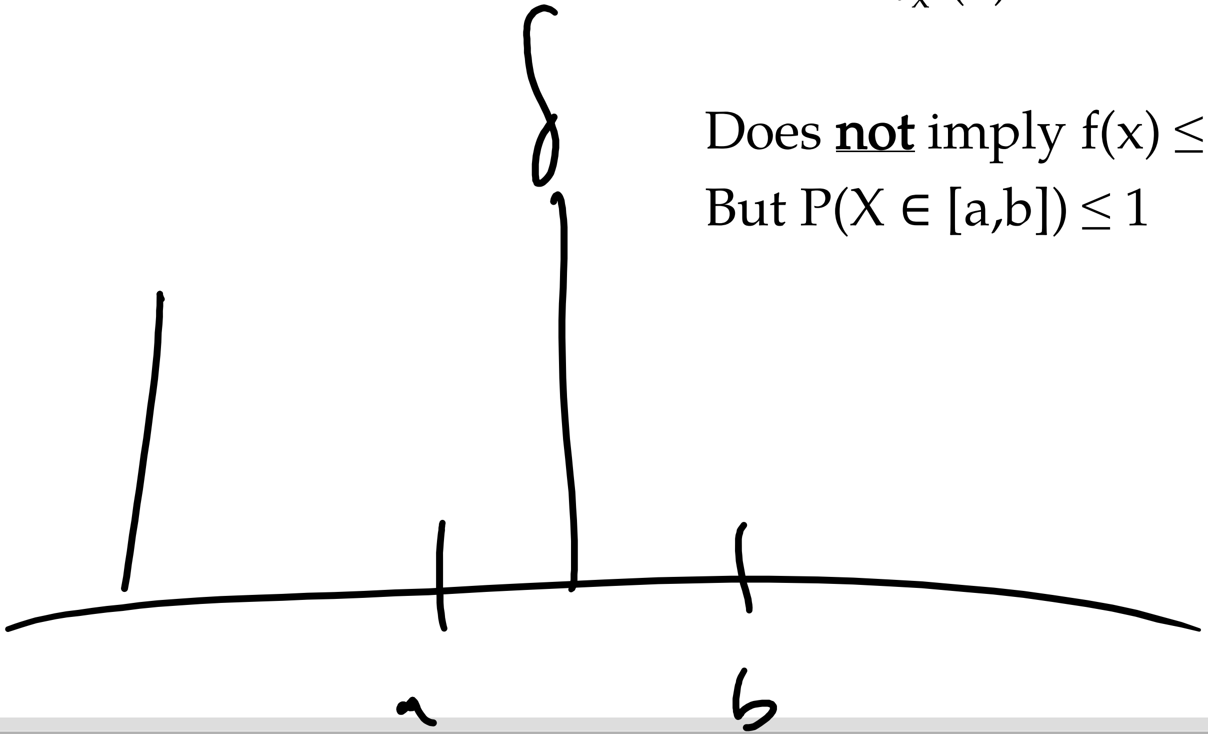
Continuous:

$f(x) \geq 0$

$\int_x f(x)\, dx = 1$

Does **not** implies $f(x) \leq 1$

But $P(X \in [a,b]) \leq 1$



Continuous random variables

Example: $X \sim \text{Uniform}[0, 1]$

What is $P(X = 0.5)$?

What is $P(X = 1/\pi)$?

What is $P(1/\pi \le X \le 0.5)$ $\qquad 0.5 - 1/\pi$

Cumulative distribution function: $F(q) = P(X \le q)$

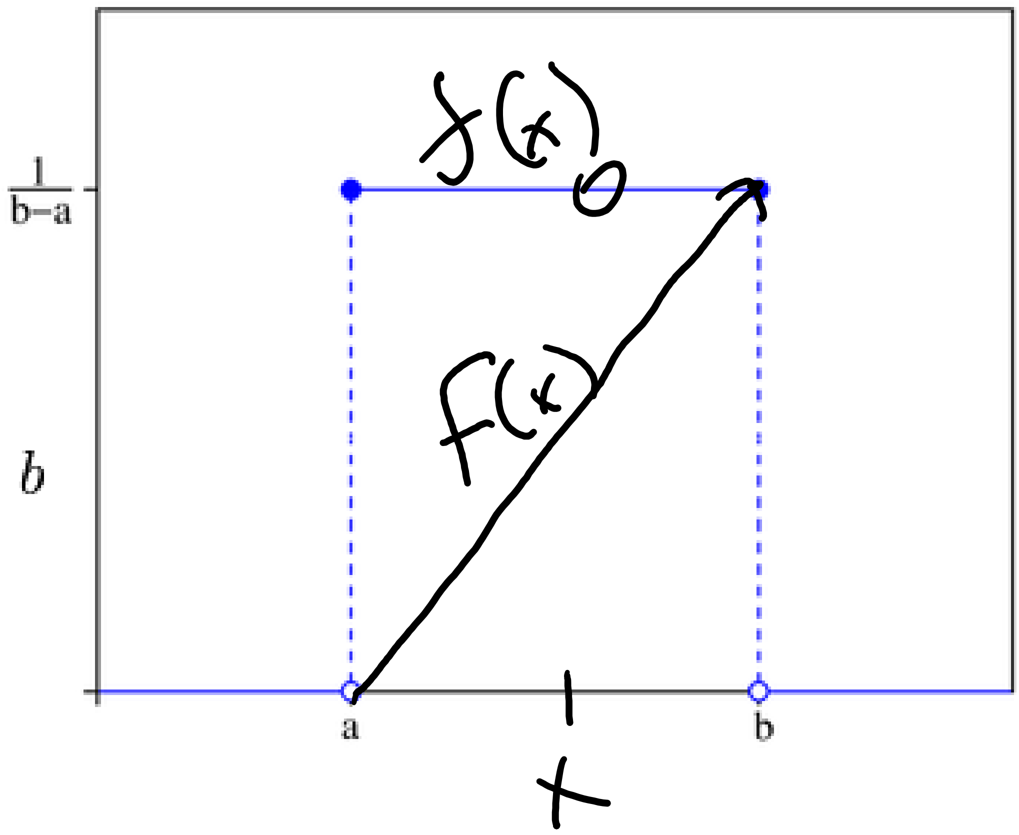
Probability density function: $f(x) = \frac{d}{dx} F(x)$

$P(a \le X \le b) = \int_{x=a}^{b} f(x)\, dx$

Uniform Distribution

$$\int^{x}_{x}$$

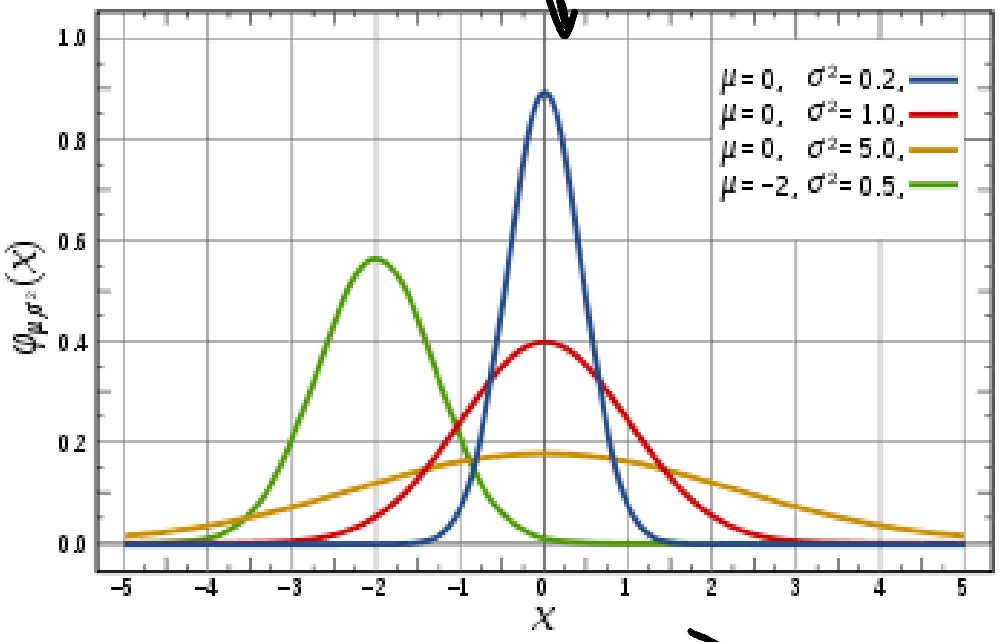
$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases}$$



Gaussian Distribution

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

μ

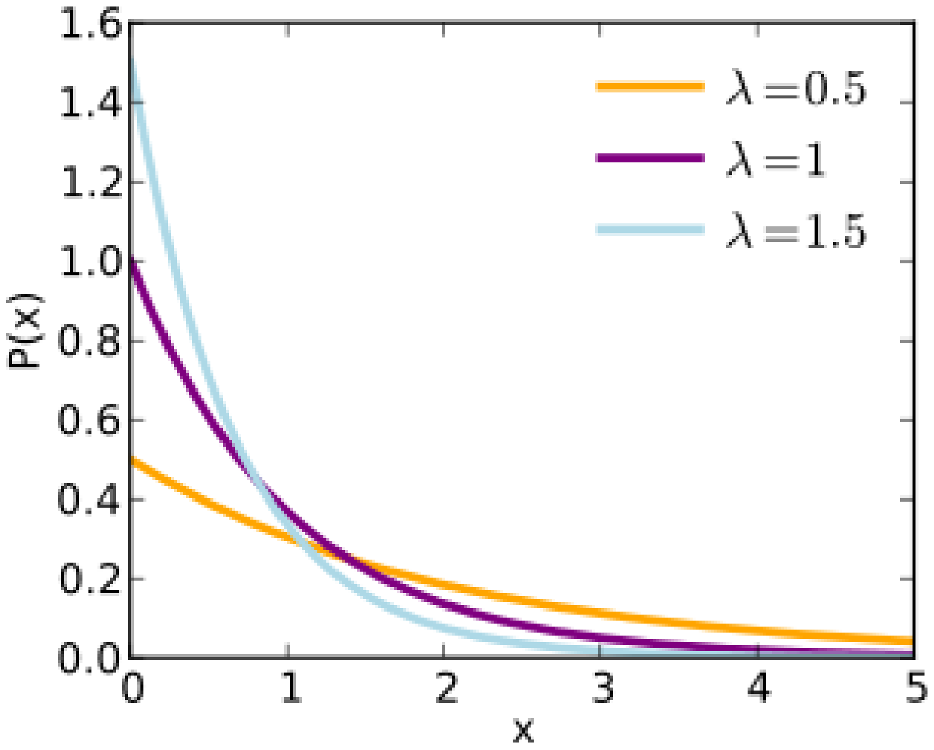


σ = stdev

σ^2 = variance

Exponential Distribution

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \ge 0, \\ 0, & x < 0. \end{cases}$$



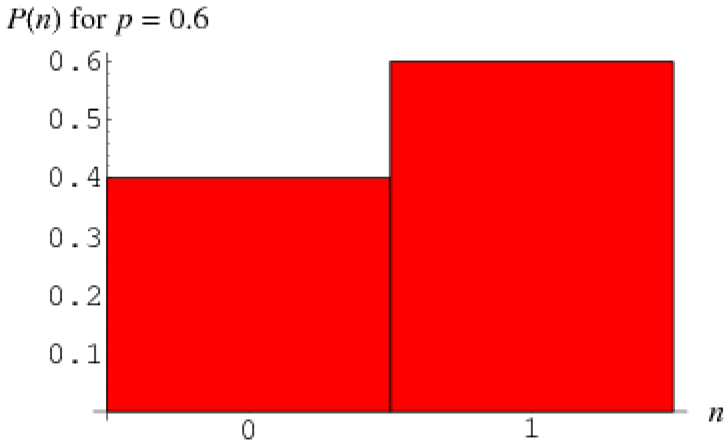
Bernoulli Distribution

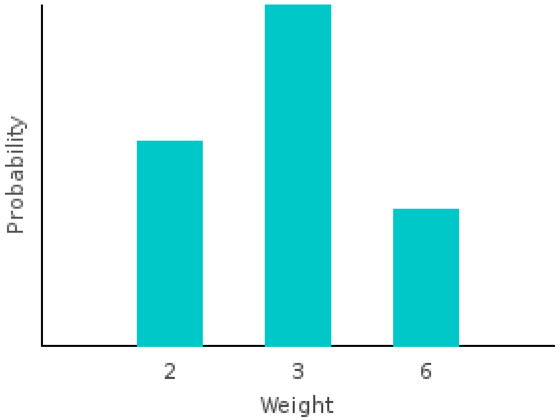
P(heads) = θ

P(tails) = 1-θ

θ ∈ [0, 1]

Extension: Categorical – given probabilities for multiple events





Expectation

What happens <u>on average</u>?

$$E[g(X)] = \sum P(x)\, g(x)$$

$$(\text{or } \textstyle\int_x f(x)\, g(x)\, dx)$$

If $X_i \sim \text{Bernoulli}(\theta)$, $E[\sum_{i=1:N} X_i]$?

Linearity properties:

$$E[g_1(X) + g_2(X)] = E[g_1(X)] + E[g_2(X)]$$

$$E[\alpha\, g(X)] = \alpha\, E[g(X)]$$

Mean / Variance / Std. Dev.

Mean: Distribution average, $\mu = E[X]$

Variance: How far distribution is "spread out"

$$\begin{aligned}\sigma^2 &= E[(X - E[X])^2] \\ &= E[(X - \mu)^2] \\ &= E[X^2] - 2\,E[X\,\mu] + \mu^2 \\ &= E[X^2] - \mu^2\end{aligned}$$

Standard Deviation: σ

Example: Chicago Weather

Random variables: Temp, Sky, Precipitation

Temperature	Sky	Precipitation	Probability
Cold	Clear	No Snow	28.0%
Cold	Clear	Snow	0.0%
Cold	Cloudy	No Snow	8.4%
Cold	Cloudy	Snow	33.6%
Warm	Clear	No Snow	12.0%
Warm	Clear	Snow	0.0%
Warm	Cloudy	No Snow	18.0%
Warm	Cloudy	Snow	0.0%

What is the probability of being cold?
What is the probability of snow?

Marginal Probabilities

Given a joint probability table, what are the probabilities of a subset of events?

$P(x) = \Sigma_{y,\,z}\, P(x, y, z)$

$P(x, y) = \Sigma_{z}\, P(x, y, z)$

Conditional Probabilities

Given that one of the random variables has a certain value, what is the distribution of other variables?

$$P(x \mid y, z) = P(x, y, z) \,/\, P(y, z)$$

$$P(x, y \mid z) = P(x, y, z) \,/\, P(z)$$

Example: Chicago Weather

Temperature	Sky	Precipitation	Probability
Cold	Clear	No Snow	28.0%
Cold	Clear	Snow	0.0%
Cold	Cloudy	No Snow	8.4%
Cold	Cloudy	Snow	33.6%
Warm	Clear	No Snow	12.0%
Warm	Clear	Snow	0.0%
Warm	Cloudy	No Snow	18.0%
Warm	Cloudy	Snow	0.0%

If it is Cold and Cloudy, what is the probability of Snow?
If it is Cold, what is the probability of Snow?