Day 2

Intro to Classification Algorithms

HPSI
Human Performance Systems, Inc.

Day 2 Outline

- Day 1 Recap
- Intro to Classifiers
- K-Nearest Neighbors
  - Algorithm details
  - Choosing ”k”
  - Exercise
  - Pros/Cons
- Probability
  - Probabilistic Classifiers
  - Probability Distributions
  - Exercise
  - Sample Spaces
  - Conditional Probability
  - Bayesian Inference
- Naïve Bayes Classifier
  - Algorithm Details
  - Exercise
  - Pros/Cons
- Logistic Regression
  - Algorithm Details
  - Exercise
- Model Comparison

Day 1 Recap

Classification

- Definition
  - Classification can take two distinct meanings in Machine Learning
    - Unsupervised Learning
      - We may be given a set of observations with the aim of establishing the existence of classes or clusters in the data
    - Supervised Learning
      - We may know for certain that there are so many classes, and the aim is to establish a rule that we can use to classify a new observation into one of the existing classes
  - Today we will be discussing methods and algorithms related to Supervised Classification

Classification

- A few classification examples:
  1. A Person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions. Which of the three conditions does the individual have?
  2. An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.
  3. On the basis of DN sequence data for a number of patients with and without a given disease, a biologist would like to figure out which DNA mutations are disease-causing and which are not.

Classification

- There are many possible techniques that a classifier might use to predict a qualitative response. Today we will discuss three of the most widely-used classifiers:
  - k-Nearest Neighbors
  - Naïve Bayes
  - Logistic Regression

Classification

A few issues to keep in mind when building a classifier

- **Accuracy.** There is the reliability of the rule, usually represented by the proportion of correct classifications, although it may be that some errors are more serious than others, and it may be important to control the error rate for some key class.
- **Speed.** In some circumstances, the speed of the classifier is a major issue. A classifier that is 90% accurate may be preferred over one that is 95% accurate if it is 100 times faster in testing (and such differences in time-scales are not uncommon in neural networks for example). Such considerations would be important for the automatic reading of postal codes, or automatic fault detection of items on a production line for example.
- **Comprehensibility.** If it is a human operator that must apply the classification procedure, the procedure must be easily understood else mistakes will be made in applying the rule. It is important also, that human operators believe the system.
- **Training Time.** Especially in a rapidly changing environment, it may be necessary to learn a classification rule quickly, or make adjustments to an existing rule in real time. "Quickly" might imply also that we need only a small number of observations to establish our rule.[4]

K-Nearest Neighbors

Simple approach for k-NN

Simple goal:

- Predict the label of a data point by:
  - Looking at the ‘k’ closest labeled data points (neighbors)
  - Taking a majority vote
- One of the easiest algorithms to interpret, oftentimes used as a baseline for measuring model performance
- Memory-Based Learning
  - Also known as “case-based” or “example-based” learning
- Intuition behind memory-based learning
  - Similar inputs map to similar outputs
    - If true, we just have to define “similar”
    - Not all similarities created equal…

Memory-Based Learning

- How do we determine “similar”?
- For instance, if we wanted to:
- Predict Brent’s weight
  - Who are the similar people?
  - Similar age, diet, height, waistline, activity level …
- Predict Brent’s IQ
  - Similar occupation, writing style, undergraduate degree, SAT score, …
- How would you quantify a comparison for these two?
  - Need some metric…
    - Distance?

kNN Model Representation

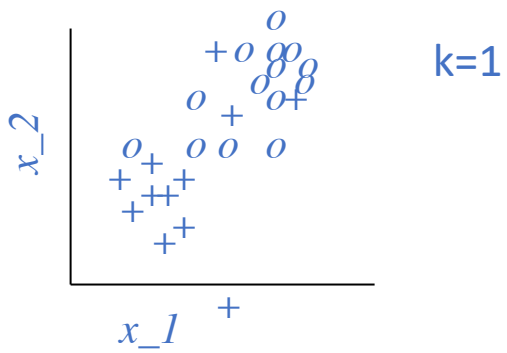
- The model representation for KNN is the entire training dataset.
- It is as simple as that.
- KNN has no “model” other than storing the entire dataset, so there is no learning required.
- Efficient implementations can store the data using complex data structures (k-d trees) to make look-up during prediction efficient.
- Because the entire training dataset is stored, you may want to think carefully about the consistency of your training data. It might be a good idea to curate it, update it often as new data becomes available and remove erroneous and outlier data.

k-NN Approach

- Define a distance $d(x_1, x_2)$ between any 2 examples
  - Examples are just feature vectors
  - So we could just use Euclidean distance …
- Training
  - Index the training examples for fast lookup (build a “database”)
- Test
  - Given a new x, find the closest neighbor (k=1) from training index
  - Classify x the same as its closest neighbor

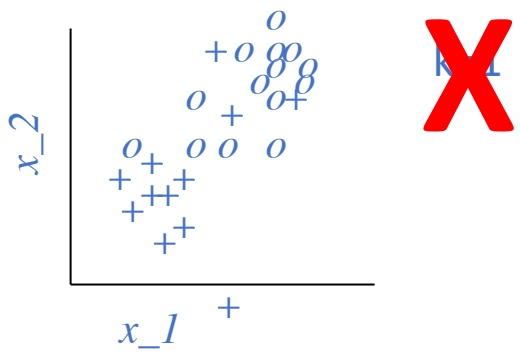
kNN

- Instead of picking the (1) nearest neighbor, what if we picked the k-Nearest Neighbors and have them vote?
- Choosing k points is more reliable in the following cases:
  - Noise in training vectors x
  - Noise in training labels y
  - Overlapping classes



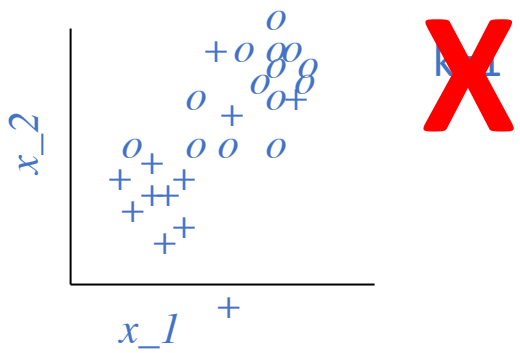
kNN

- Instead of picking the (1) nearest neighbor, what if we picked the k-Nearest Neighbors and have them vote?
- Choosing k points is more reliable in the following cases:
  - Noise in training vectors x
  - Noise in training labels y
  - Overlapping classes



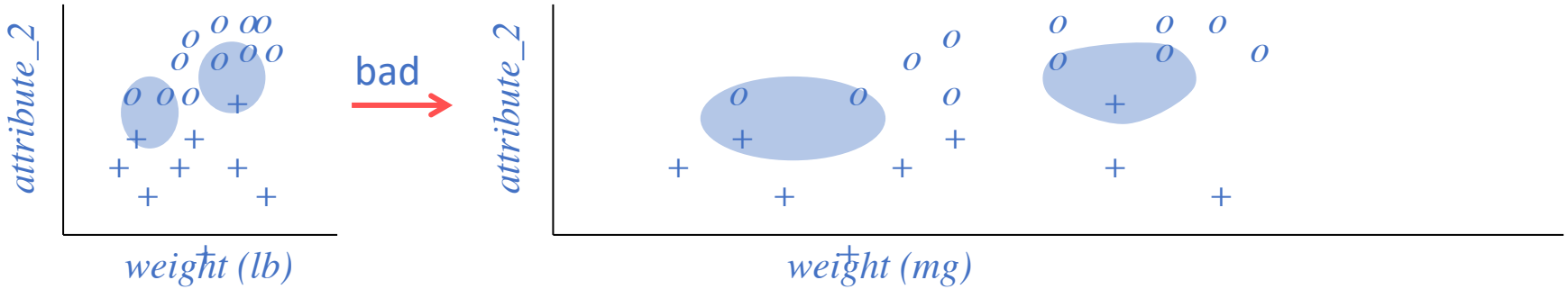
kNN

- Instead of picking the (1) nearest neighbor, what if we picked the k-Nearest Neighbors and have them vote?
- Choosing k points is more reliable in the following cases:
  - Noise in training vectors x
  - Noise in training labels y
  - Overlapping classes
- Why?



kNN distance problem

- Problem:
  - What if the input represents weight in milligrams?
  - Then small differences in physical weight dimension have a huge effect on distances, overwhelming other features
  - Should really correct for these arbitrary "scaling" issues
    - This leads to Standard Scaling (from yesterday)
    - Rescale weights so that standard deviation = 1



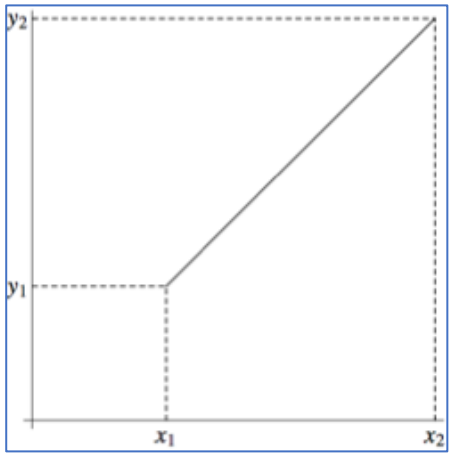
HPSI
Human Performance Systems, Inc.

Making Predictions and Distance

- KNN makes predictions using the training dataset directly.
- Predictions are made for a new instance (x) by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances.
- For regression this might be the mean output variable, in classification this might be the mode (or most common) class value.
- To determine which of the K instances in the training dataset are most similar to a new input a distance measure is used. For real-valued input variables, the most popular distance measure is Euclidean distance.
- Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (xi) across all input attributes j
- In two dimensions:

In the Euclidean plane, if $\mathbf{p} = (p_1, p_2)$ and $\mathbf{q} = (q_1, q_2)$ then the distance is given by

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$



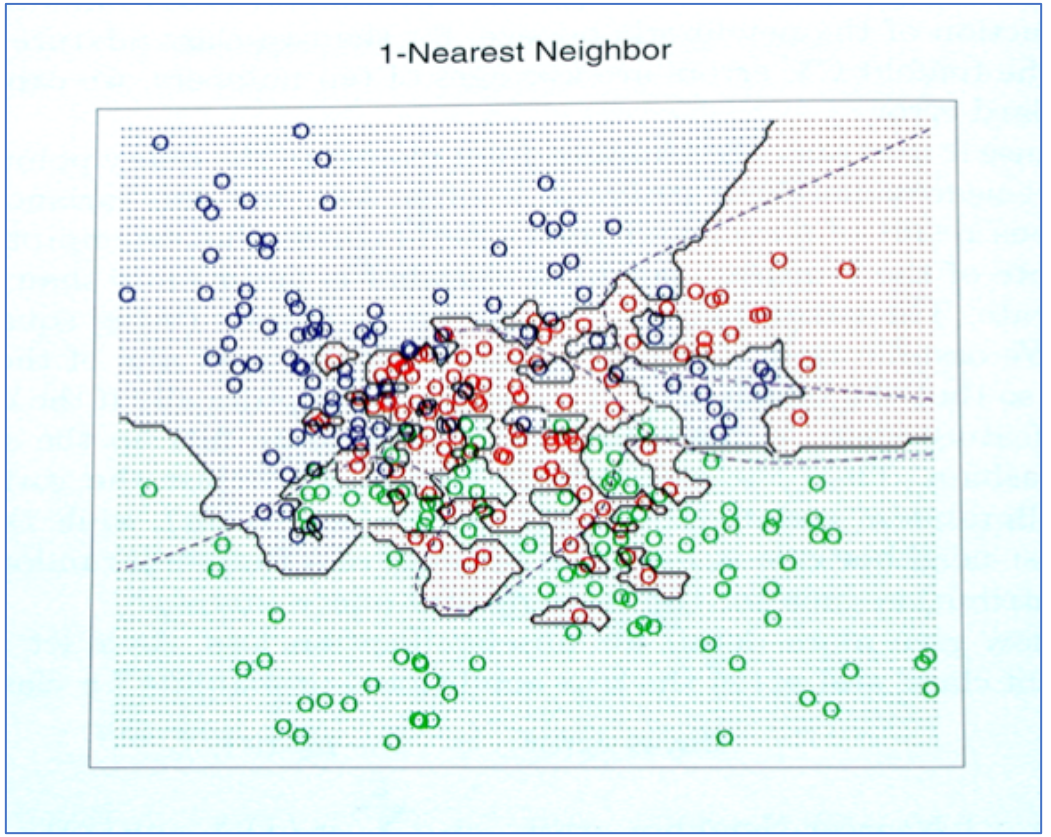
HPSI
Human Performance Systems, Inc.

Distance Metrics

- Other popular distance metrics include:
  - Hamming Distance (Calculate the distance between binary vectors)
  - Manhattan Distance (Calculate the distance between real vectors using the sum of their absolute difference. Also called City Block Distance)
  - Minkowski Distance (Generalization of Euclidean and Manhattan Distance – Lp norm)
- Other less-known distance metrics that can be used include Tanimoto, Jaccard, Mahalanobis, and cosine distance.
- Generally
  - Euclidean is a good distance measure to use if the input variables are similar in type (e.g. all measured widths and heights).
  - Manhattan distance is a good measure to use if the input variables are not similar in type (such as age, gender, height, etc.).
- If you are unsure, you can experiment with different distance metrics and different values of K together and see which mix results in the most accurate models.

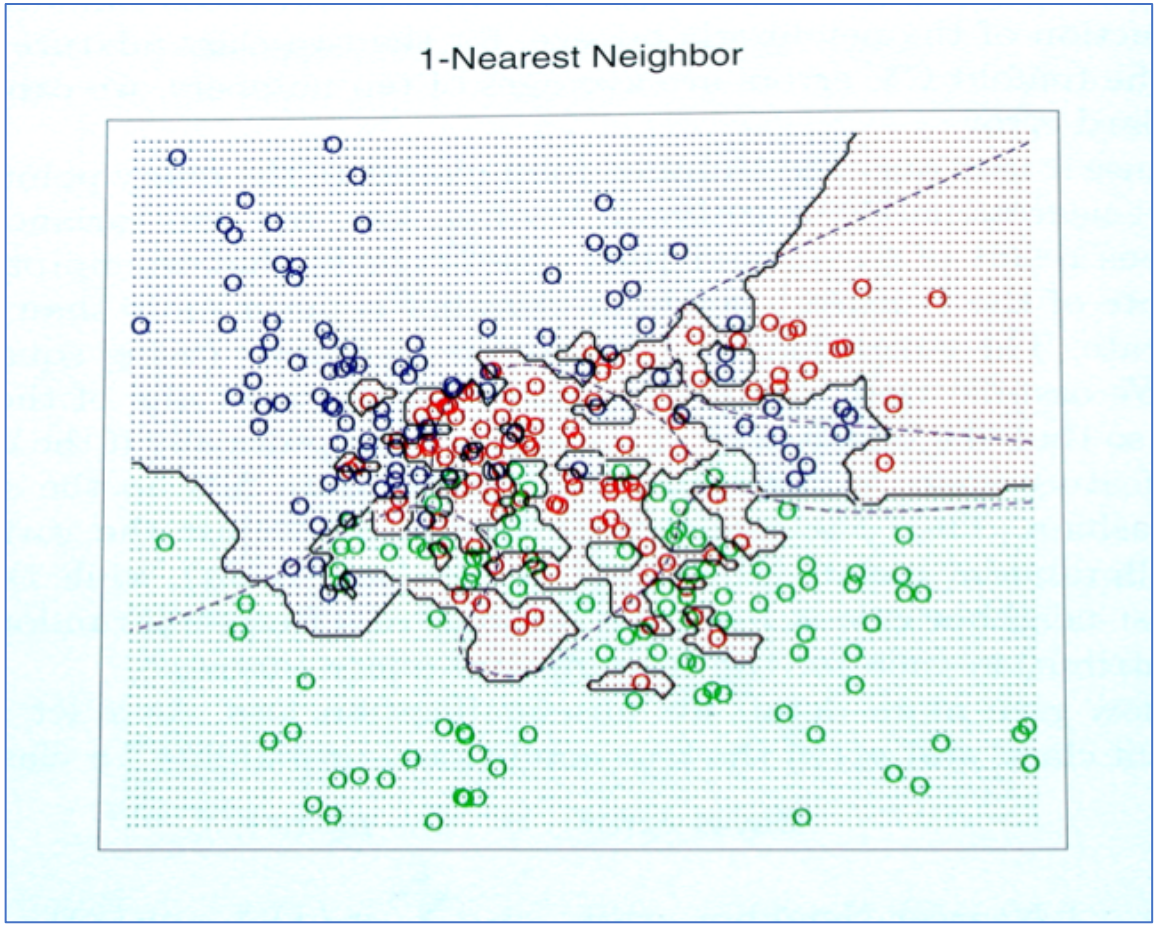
kNN Decision Boundaries

- kNN can learn complex decision boundaries



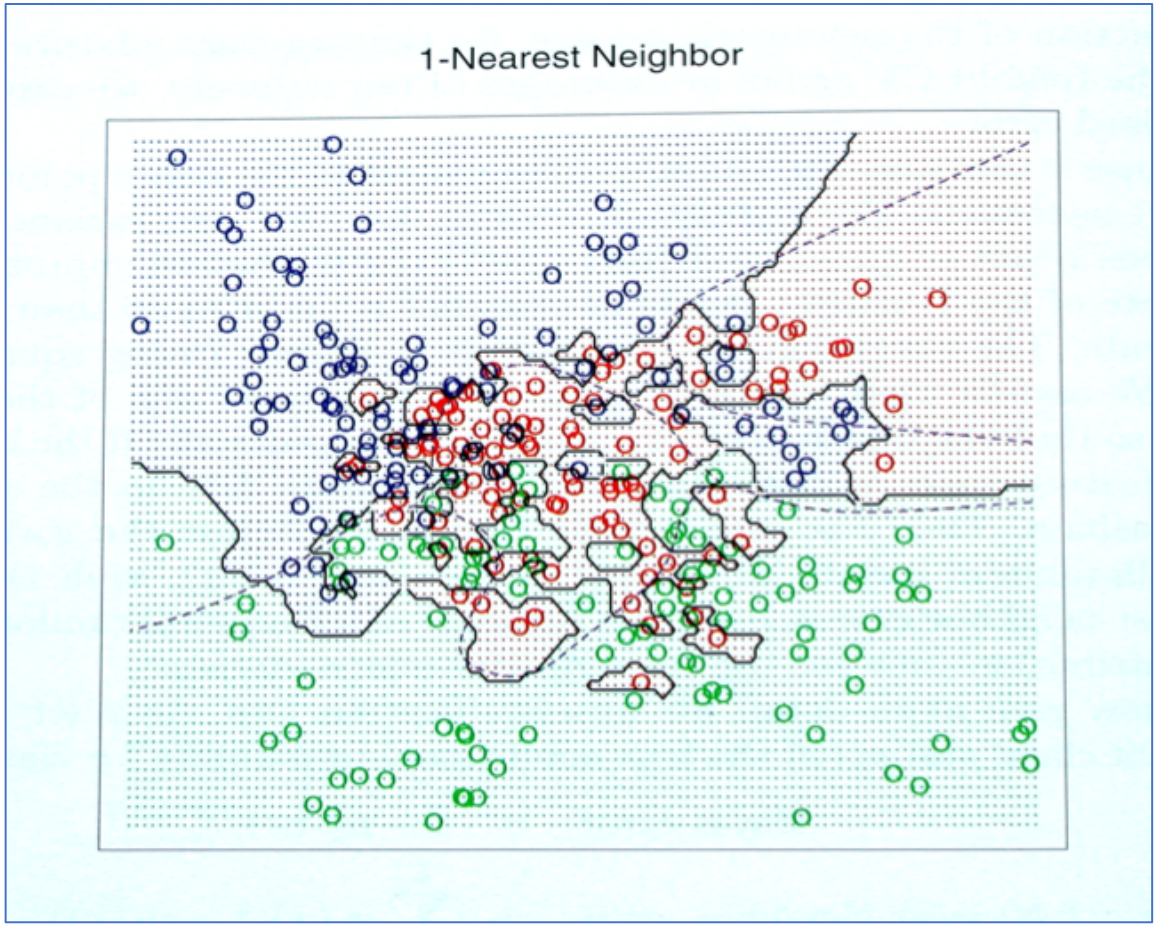
kNN Decision Boundaries

- Consider this example with R,G,B classes with significant overlap



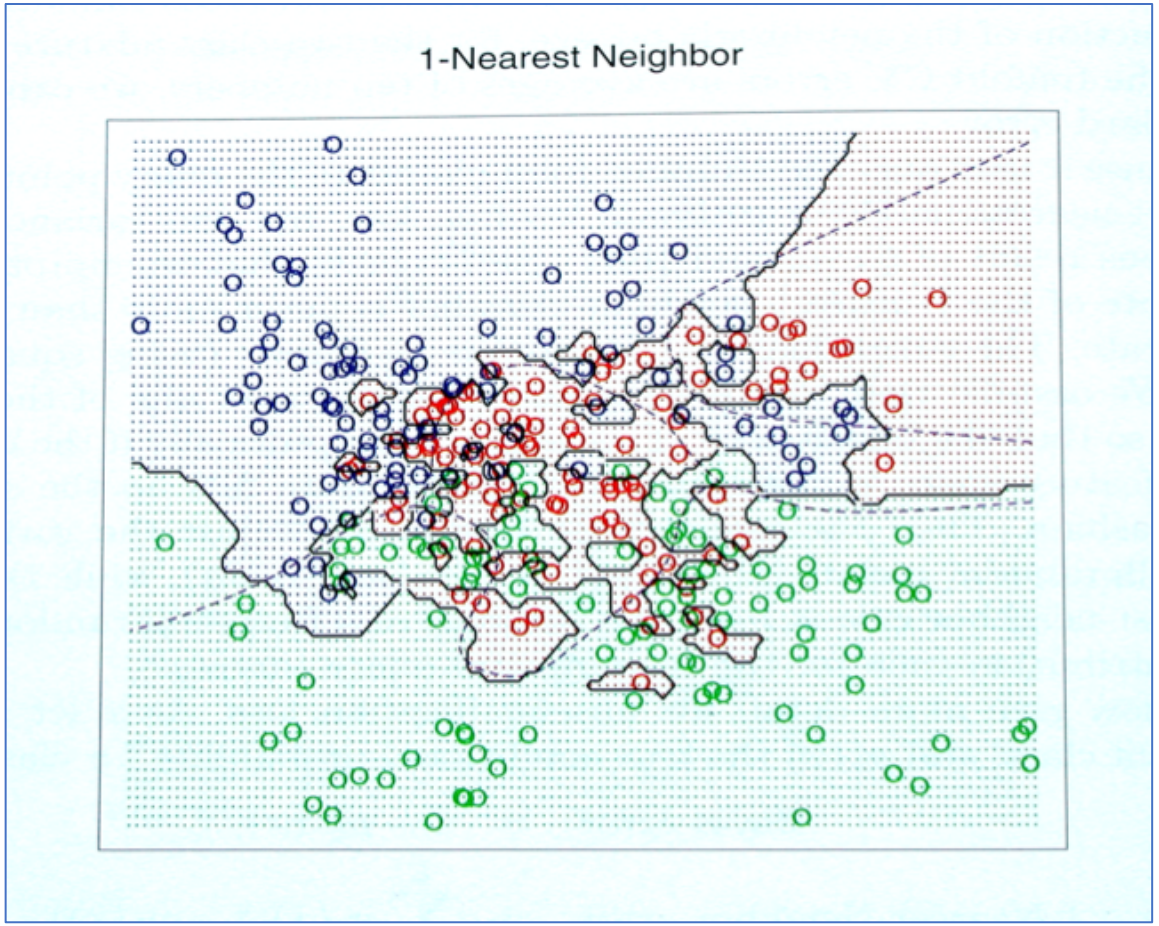
kNN Decision Boundaries

- Consider this example with R,G,B classes with significant overlap
- k=1 Decision Boundary
  - Looks complex
  - Overfitting?



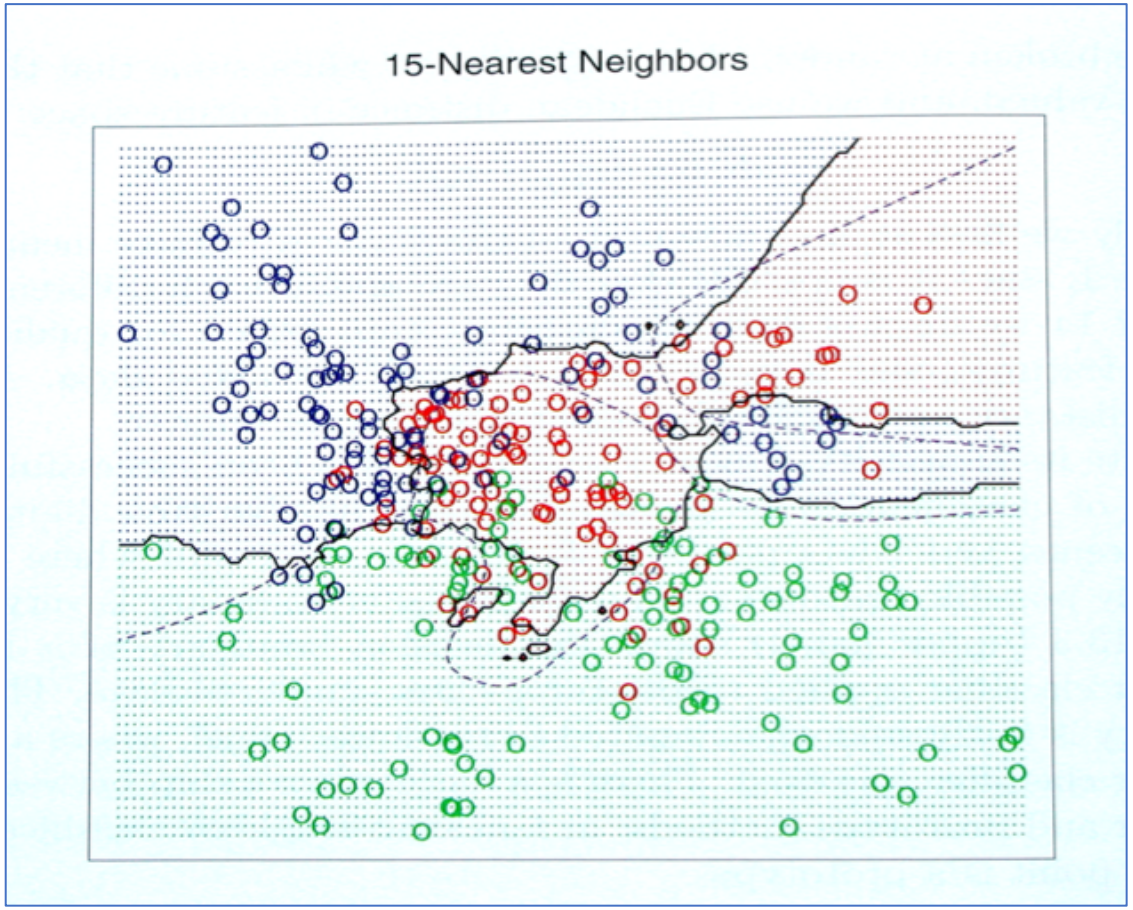
kNN Decision Boundaries

- k=1 Decision Boundary
  - Looks complex
  - Overfitting?
- What if we were to increase k?



kNN Decision Boundaries

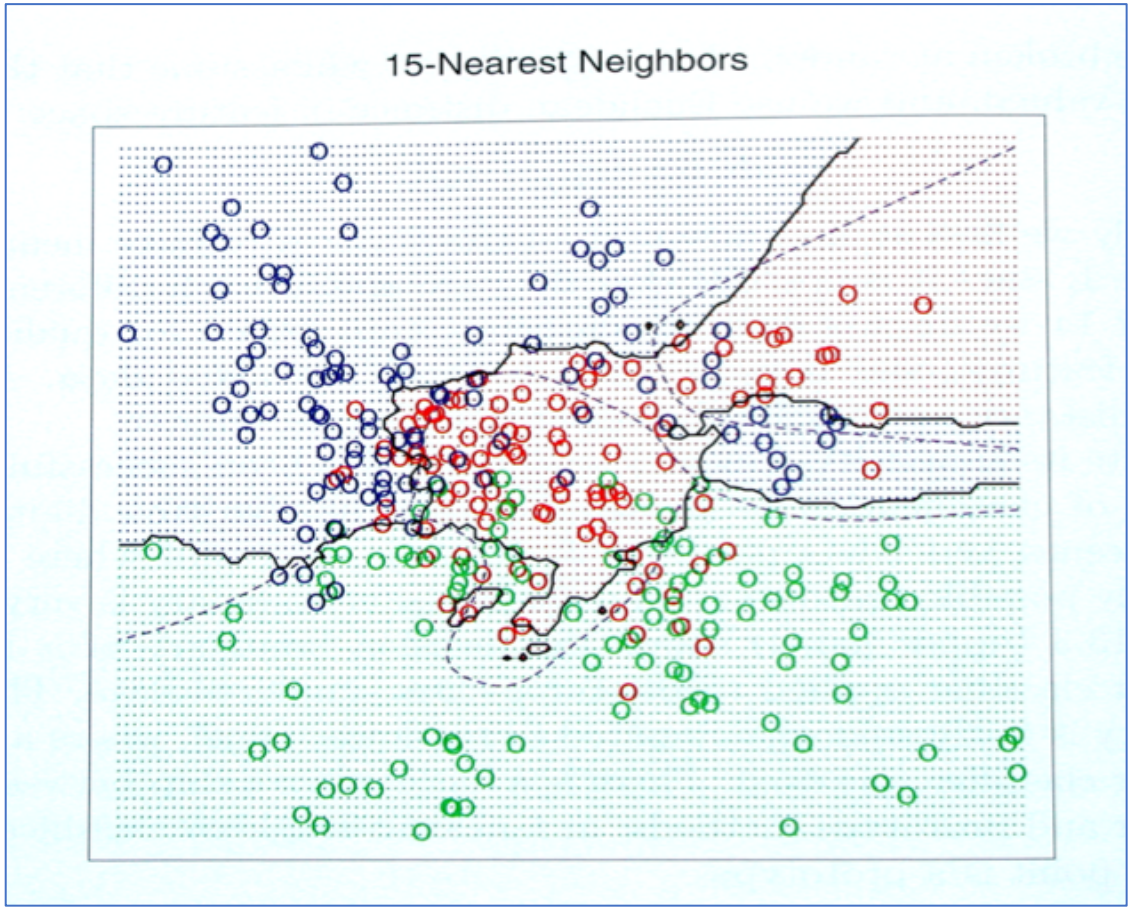
- k=1 Decision Boundary
  - Looks complex
  - Overfitting?
- What if we were to increase k?
  - K=15 Decision boundary



HPSI
Human Performance Systems, Inc.

kNN Decision Boundaries

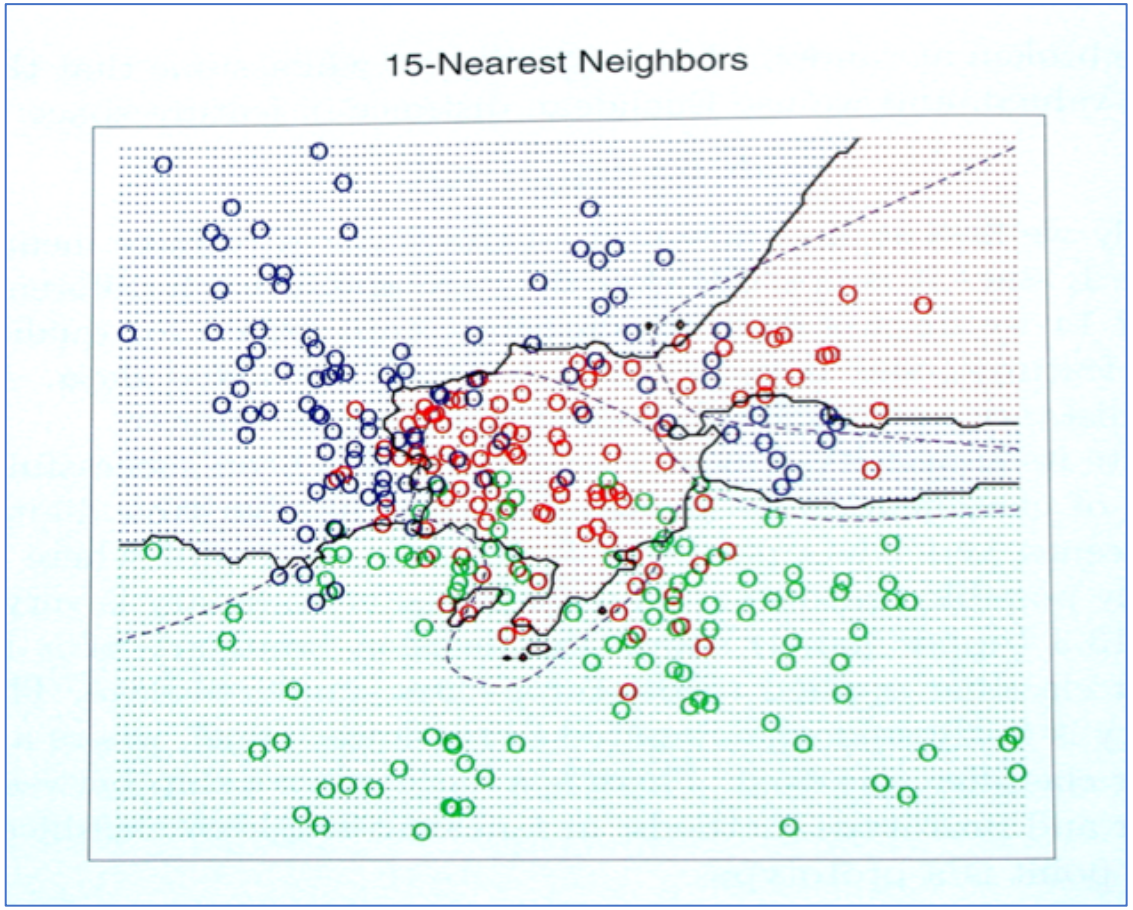
- k=1 Decision Boundary
  - Looks complex
  - Overfitting?
- What if we were to increase k?
  - K=15 Decision boundary
  - Smoother boundaries



HPSI
Human Performance Systems, Inc.

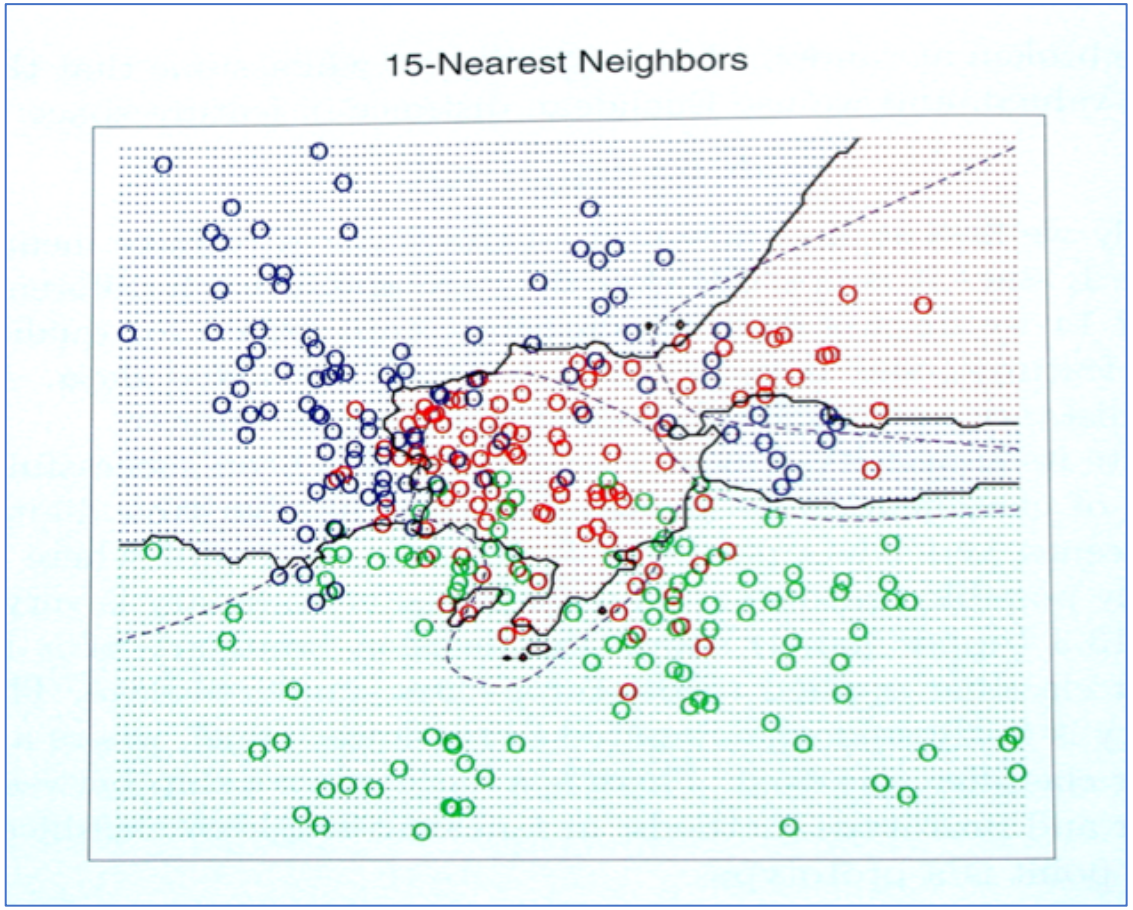
kNN Decision Boundaries

- k=1 Decision Boundary
  - Looks complex
  - Overfitting?
- What if we were to increase k?
  - K=15 Decision boundary
  - Smoother boundaries
  - Generalizes better on unseen data



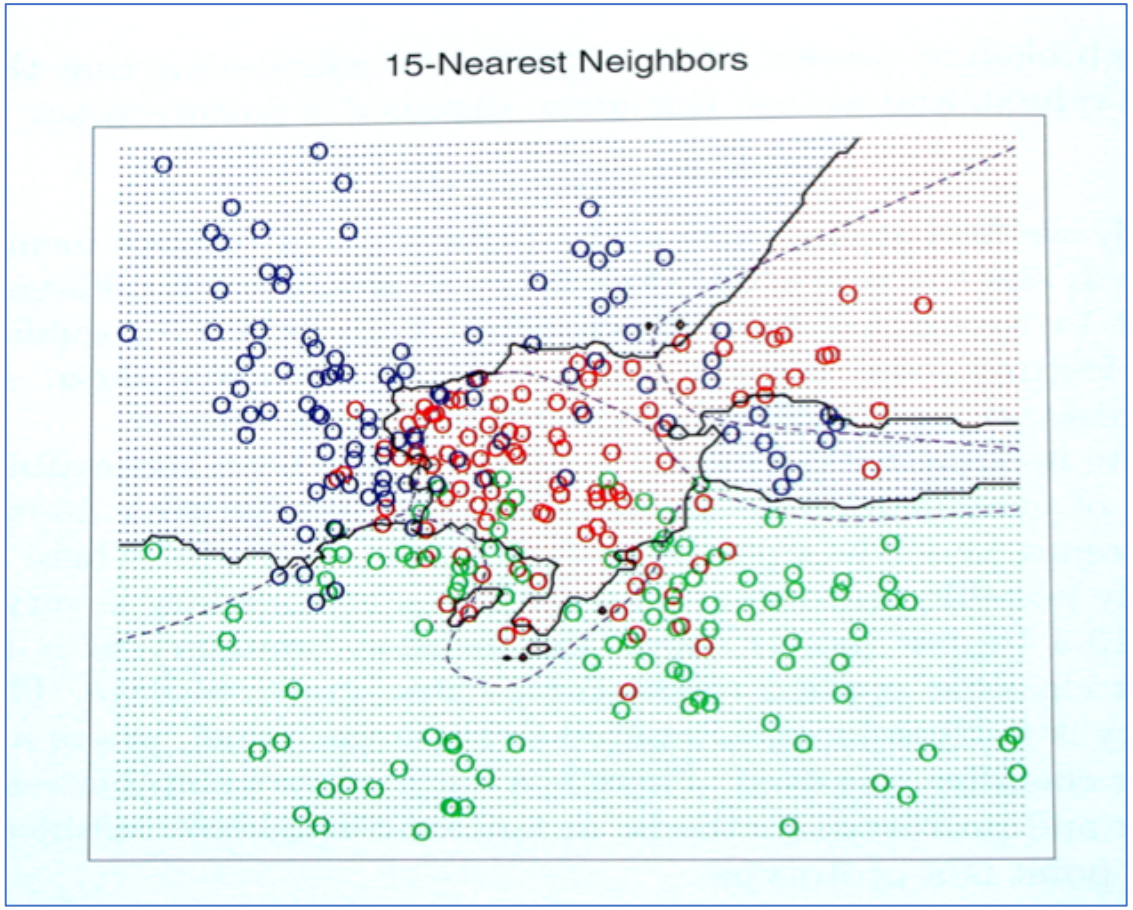
kNN Decision Boundaries

- k=1 Decision Boundary
  - Looks complex
  - Overfitting?
- What if we were to increase k?
  - K=15 Decision boundary
  - Smoother boundaries
  - Generalizes better on unseen data
- What makes the boundaries smoother?



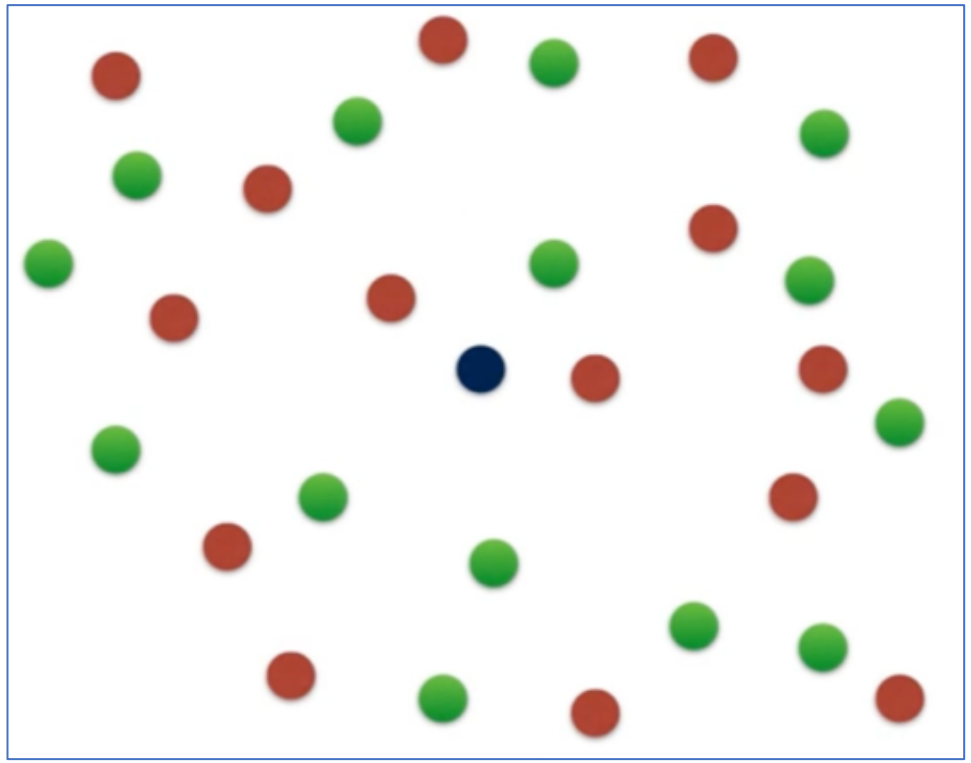
kNN Decision Boundaries

- k=1 Decision Boundary
  - Looks complex
  - Overfitting?
- What if we were to increase k?
  - K=15 Decision boundary
  - Smoother boundaries
  - Generalizes better on unseen data
- What makes the boundaries smoother?
- Let's look at a two-class (binary) example



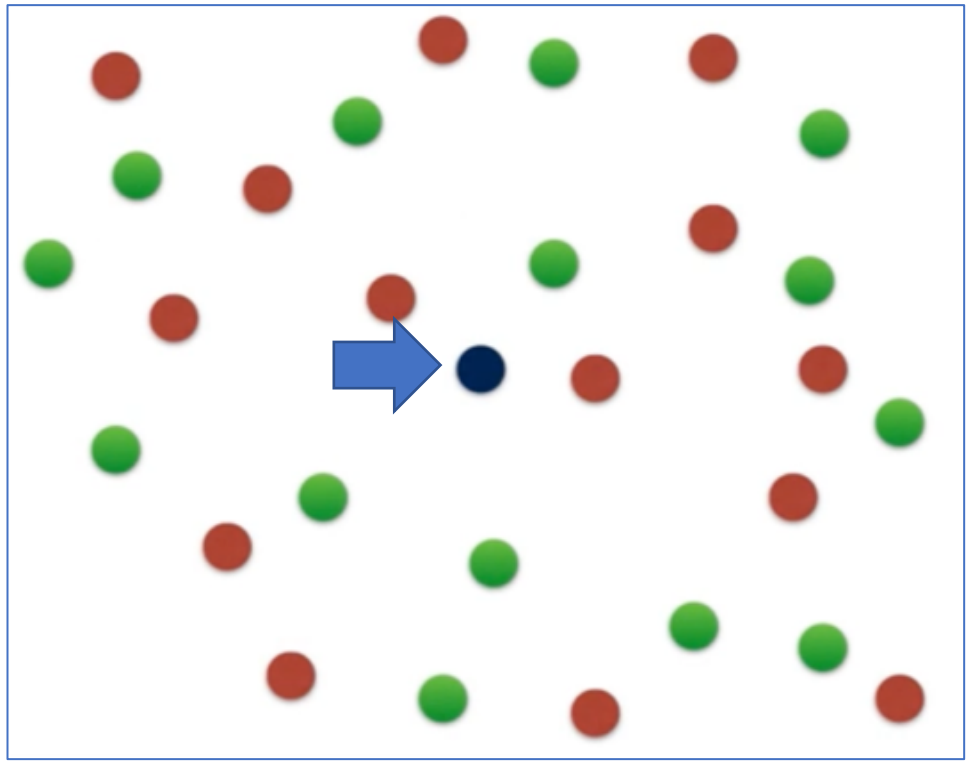
k-NN Graphical Example

- Consider this two-dimensional dataset with points classified as Red or Green



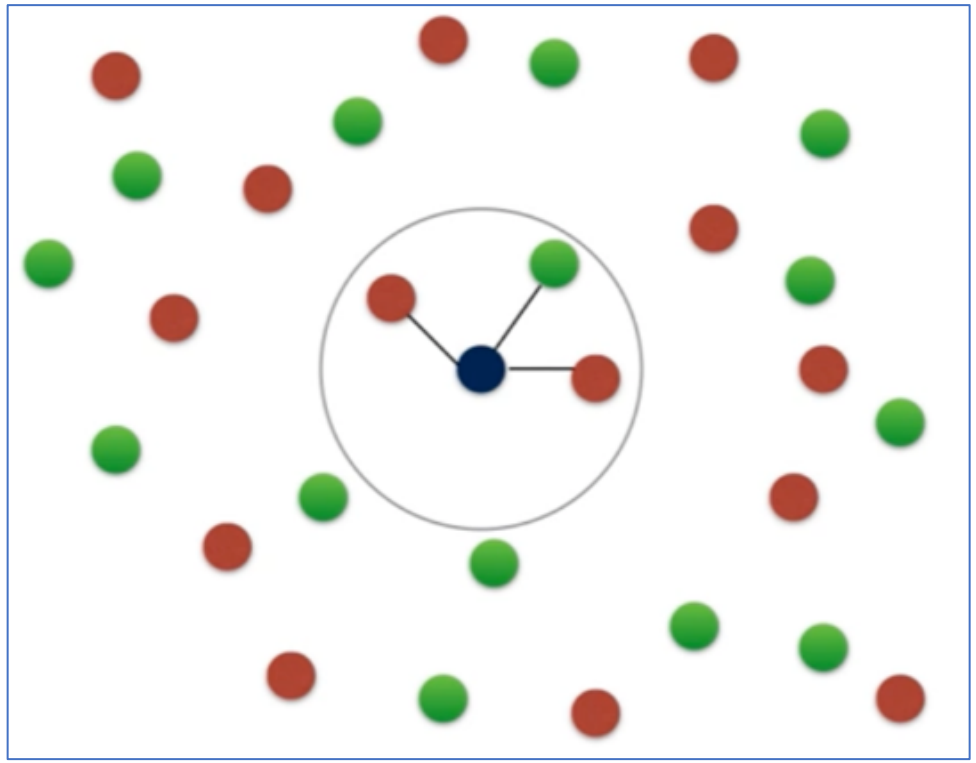
k-NN Graphical Example

- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point



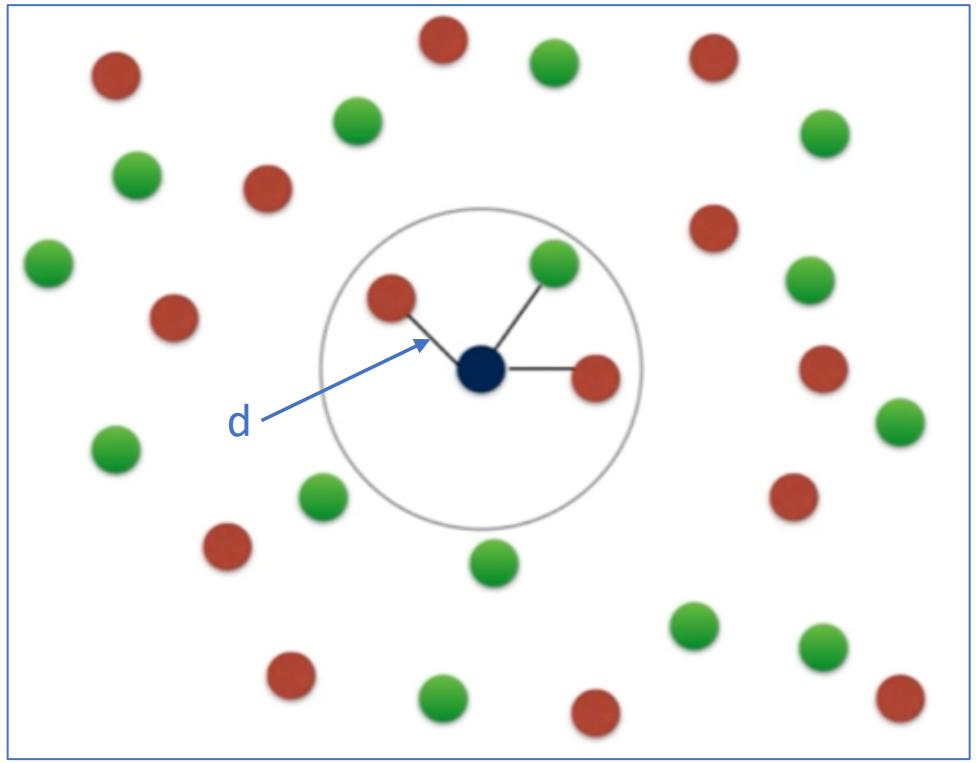
k-NN Graphical Example

- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors



k-NN Graphical Example

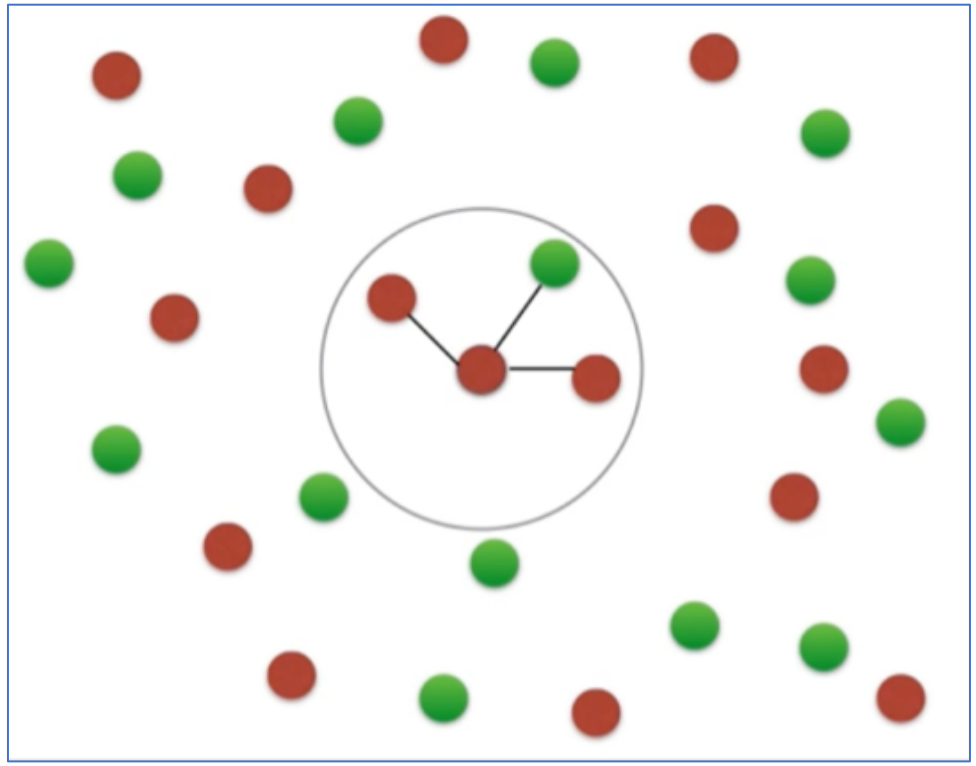
- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors
  - Measured by some distance



HPSI
Human Performance Systems, Inc.

k-NN Graphical Example

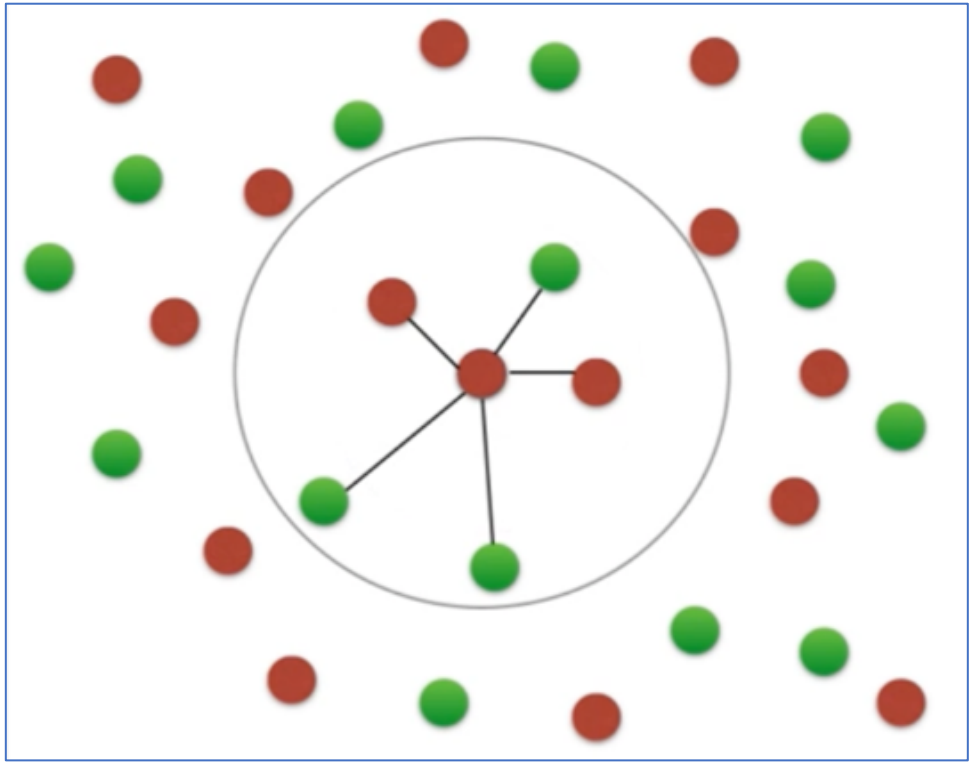
- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors
  - Measured by some distance
  - The point is classified as Red



HPSI
Human Performance Systems, Inc.

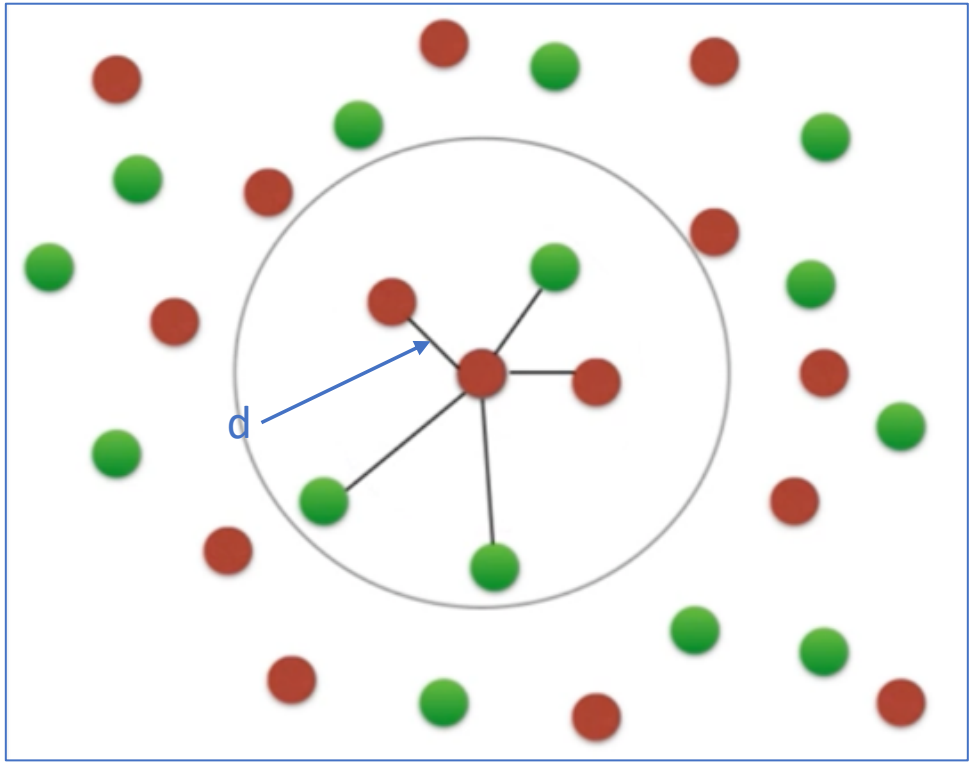
k-NN Graphical Example

- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors
  - Measured by some distance
  - The point is classified as Red
- If we consider k=5 neighbors



k-NN Graphical Example

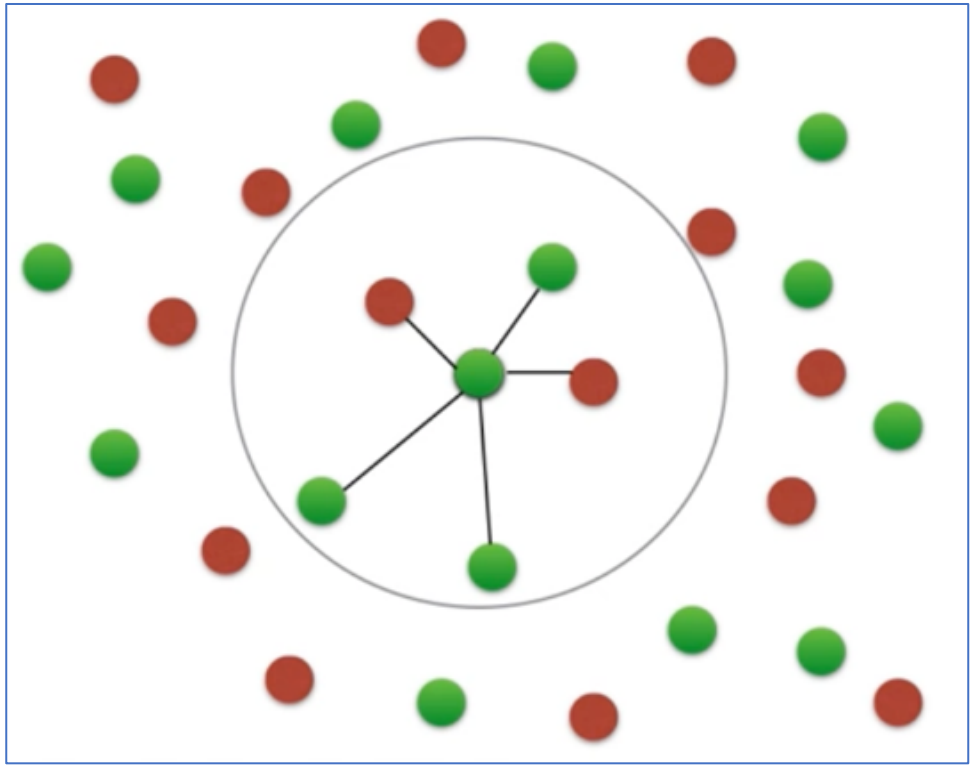
- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors
  - Measured by some distance
  - The point is classified as Red
- If we consider k=5 neighbors
  - Measured by some distance



HPSI
Human Performance Systems, Inc.

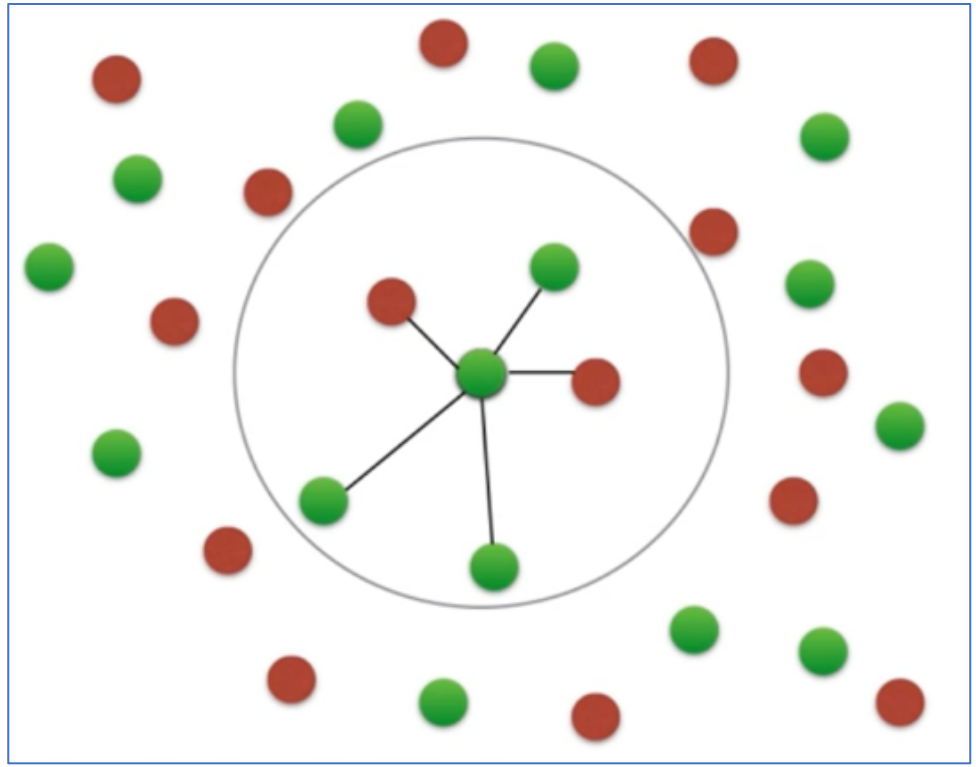
k-NN Graphical Example

- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors
  - Measured by some distance
  - The point is classified as Red
- If we consider k=5 neighbors
  - Measured by some distance
  - The point is classified as Green



k-NN Graphical Example

- Consider this two-dimensional dataset with points classified as Red or Green
- We want to Classify this point
- If we consider k=3 neighbors
  - Measured by some distance
  - The point is classified as Red
- If we consider k=5 neighbors
  - Measured by some distance
  - The point is classified as Green
- So, how do we know what k to choose?



How to choose “k”

- Odd k (often 1, 3, or 5):
  - Avoids problem of breaking ties (in a binary classifier)
- Large k:
  - Less sensitive to noise (particularly class noise)
  - Better probability estimates for discrete classes
  - Larger training sets allow larger values of k
- Small k:
  - Captures fine structure of problem space better
  - May be necessary with small training sets
- It is dependent on your training data
- A good rule of thumb is $k = \sqrt{n}$ where n is the number of training examples
- Can plot an elbow curve (see example)

Best Preparation of Data for kNN

- **Rescale Data**: KNN performs much better if all of the data has the same scale.
  - Scaling (Min/Max) your data to the range [0, 1] is a good idea.
  - It may also be a good idea to standardize your data if it has a Gaussian distribution.
- **Address Missing Data**: Missing data will mean that the distance between samples can not be calculated. These samples should either be excluded (dropped) or the missing values could be imputed.
- **Lower Dimensionality**: KNN is best suited for lower dimensional data. You can try it on high dimensional data (hundreds or thousands of input variables) but be aware that it may not perform as well as other techniques. KNN can benefit from feature selection that reduces the dimensionality of the input feature space.

Recap/Summary of kNN

- Nonparametric - makes no explicit assumptions about the underlying distribution of the input
- Instance/memory-based learning means that this algorithm doesn't explicitly learn a model. Instead, it chooses to memorize the training instances which are subsequently used as "knowledge" for the prediction phase
- Learns arbitrarily complicated decision boundaries
- Lazy learner - a learning method that generalizes data in the testing phase, rather than during the training phase
  - Don't do any work until you know what you want to predict
  - A benefit of lazy learning is that it can quickly adapt to changes, since it is not expecting a certain generalized dataset.
  - Very fast training time, but very slow prediction (has to search for the nearest neighbors)

Advantages of k-NN

- Simple and fast to deploy
  - Little to no training time
- Easy to interpret/explain
- Naturally handles multiclass datasets
- Non-parametric
  - Does not assume any probability distributions on the input data

Disadvantages of k-NN

- Storage of model takes a lot of disk space (contains entire training dataset)
- Curse of Dimensionality - often works best with 25 or fewer dimensions
  - There is little difference between the nearest and farthest neighbor in high dimensional data
- Computationally expensive predictions (large search problem to find nearest neighbors)
  - Might be impractical in industry settings
- Need to normalize - suffers from skewed class distributions
  - If one type of category occurs much more than another, classifying an input will be more biased towards that one category (dominates the majority vote since it is more likely to be neighbors with the input)

kNN Exercise

Naïve Bayes

Naïve Bayes

- Naive Bayes is a simple classification technique that relies on conditional probability, and predicts the most probable class given a set of inputs
- It is often used as a baseline for more complex models

A new method: Naïve Bayes

- Naive Bayes is a simple technique for predicting the most probable class/label given a set of features/inputs
  - It is often used as a baseline for more complex models

$$\text{Naive Bayes Classifier: } \underset{Y}{\text{argmax}}\ P(Y|\vec{X}) = \underset{Y}{\text{argmax}}\ P(x_1|Y)P(x_2|Y)...P(x_3|Y)P(Y)$$

$$= \underset{Y}{\text{argmax}}\ P(Y)\prod_{i=1}^{\kappa} P(x_i|Y)$$

How can we unpack this?

Understanding Naïve Bayes Classifiers

- To be able to unpack the Naïve Bayes classifier definition, we need a good grasp on the following topics
  - Random variables
  - Distributions
    - Continuous
    - Discrete
  - Statistical Independence
  - Probability
    - Conditional Probability
    - Joint Probability
    - Marginal Probability

Random Variables

- A random variable is a random number determined by chance, or more formally, drawn according to a probability distribution which specifies the probability that its value falls in any given interval.
- Discrete Random Variable
  - Taking any of a specified finite or countable list of values, endowed with a probability mass function characteristic of the random variable's probability distribution
- Continuous
  - Taking any numerical value in an interval or collection of intervals, via a probability density function that is characteristic of the random variable's probability distribution; or a mixture of both types.

Random Variables

- Why do we care about Random Variables?
- Our goal is to predict the target/class
- We are not given the true (presumably deterministic) function
- We are only given observations
- Uncertainty arises through:
  - Noisy measurements
  - Finite size of data sets
  - Ambiguity: The word “bank” can mean (1) a financial institution, (2) the side of a river,or (3) tilting an airplane. Which meaning was intended, based on the words that appear nearby?
- Probability theory provides a consistent framework for the quantification and manipulation of uncertainty
- Allows us to make optimal predictions given all the information available to us, even though that information may be incomplete or ambiguous

Probability

- Probabilities assign numbers to possibilities
- A probability needs to satisfy three properties (Kolmogorov, 1956):
  - A probability must be nonnegative
  - The sum of the probabilities across all events in the entire sample space must be 1
  - For any two mutually exclusive events, the probability that one or the other occurs is the sum of their individual probabilities
    - For example, the probability that a fair six-sided die comes up 3 OR 4 is 1/6 + 1/6 = 2/6.

Probability Distributions

- A probability distribution is simply a list of all possible events and their corresponding probabilities
- There are two kinds of probability distributions
  - Discrete Distribution:
    - Probability of heads or tails
  - Continuous Distribution:
    - Probabilities of people's heights

Discrete Probability Distribution

- When the sample space consists of discrete outcomes (e.g., heads or tails), the probability distribution is a list of probabilities of the outcomes
- The probability of a discrete outcome is called a **probability mass**
- The sum of the probability masses across the sample space must be 1

Discrete Probability Example

- Example
  - Consider the simple experiment of tossing a coin three times. Let X = number of times the coin comes up heads. The 8 possible elementary events and the corresponding values for X are:

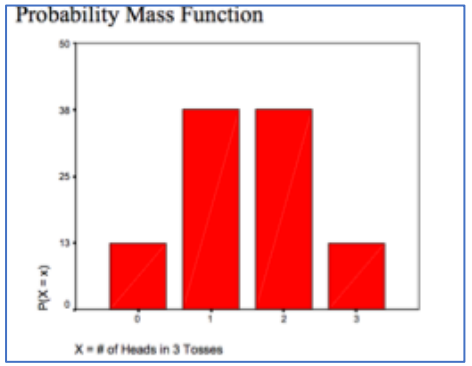
Elementary Event	Count of Heads (X)
TTT	0
TTH	1
THT	1
HTT	1
THH	2
HTH	2
HHT	2
HHH	3

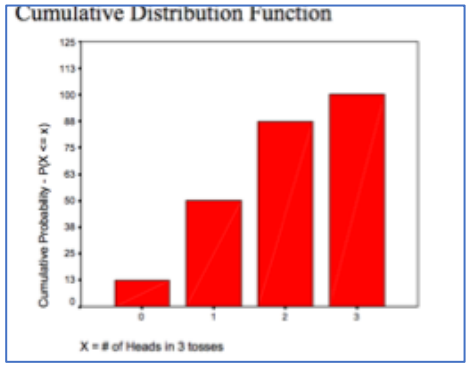
Discrete Probability Example

- Example
  - Therefore, the probability distribution for the number of heads occurring in three coin tosses is

Count of Heads (X)	p(x)	F(x)
0	1/8	1/8
1	3/8	4/8
2	3/8	7/8
3	1/8	1

$$P(x) = \begin{cases} 1/8 & \text{if } x = 0 \\ 3/8 & \text{if } x = 1, 2 \\ 1/8 & \text{if } x = 3 \\ 0 & \text{Otherwise} \end{cases}$$





HPSI
Human Performance Systems, Inc.

Continuous Probability Distribution

- When the sample space consists of continuous outcomes (ex: people's heights) we cannot use probability mass for a specific outcome.
- Why not?

Continuous Probability Distribution – Probability Density

- When the sample space consists of continuous outcomes (ex: people's heights) we cannot use probability mass for a specific outcome.
- Why not?
  - Because the probability mass for a specific outcome will be zero
  - In other words, the probability of someone's height being exactly 67.2141390842076153…
- Instead, we can:
  - Discretize the space into a finite set of mutually exclusive and exhaustive intervals
  - Calculate the probability mass in each interval
  - Use the ratio of probability mass to interval width
  - This ratio is called the **Probability Density**

Probability Density

- The top panel of this figure shows the discretized intervals and probability mass in each interval
- The second panel shows the probability density
- The third panel shows the narrower intervals and probability mass in each interval
- The bottom panel shows the probability density corresponding to the more narrow intervals
- Generally, the skinnier the intervals are, the more accurate the probability density is

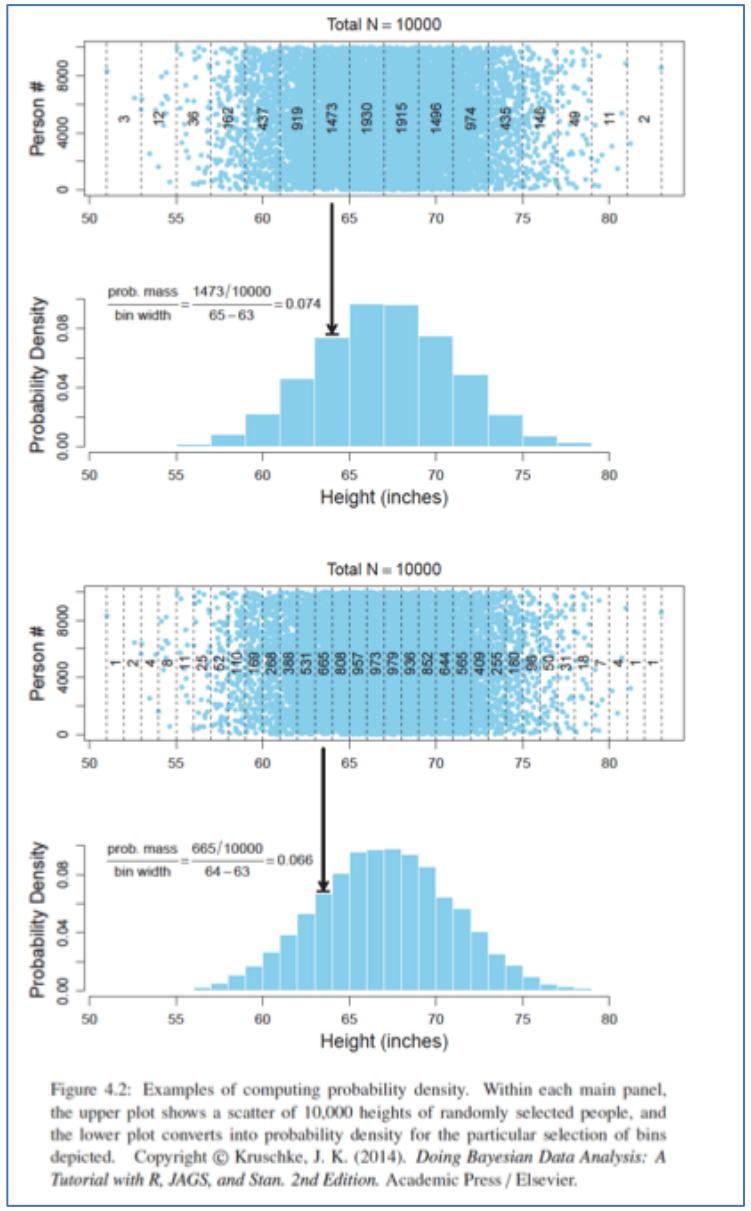


Figure 4.2: Examples of computing probability density. Within each main panel, the upper plot shows a scatter of 10,000 heights of randomly selected people, and the lower plot converts into probability density for the particular selection of bins depicted. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

HPSI
Human Performance Systems, Inc.

Probability Density

- While probability mass cannot exceed 1, probability densities can
- The upper panel of this figure shows that most of the probability mass is concentrated around 84
- Consequently, the probability density near 84 exceeds 1.0, as shown in the lower panel
- This simply means that there is a high concentration of probability mass relative to the width of the interval

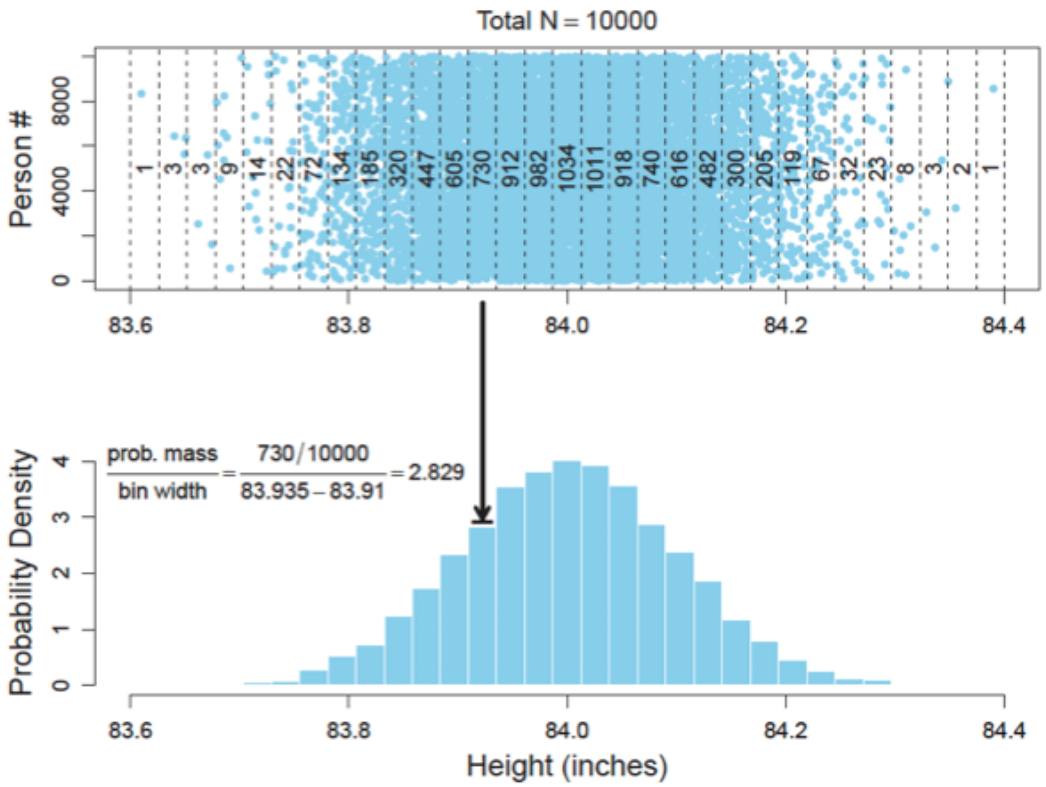


Figure 4.3: Example of probability density greater than 1.0. Here, all the probability mass is concentrated into a small region of the scale, and therefore the density can be high at some values of the scale. The annotated calculation of density uses rounded interval limits for display. (For this example, we can imagine that the points refer to manufactured doors instead of people, and therefore the *y*-axis of the top panel should be labelled "Door" instead of "Person.") Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

HPSI
Human Performance Systems, Inc.

Properties of Probability Density Functions

- We need to define some notations first
- Let:
  - x be the continuous variable
  - Δx be the width of an interval on x
  - i be an index for the intervals
  - $[x_i, x_i + \Delta x]$ be the interval between x_i and $x_i + \Delta x$
  - $P([x_i, x_i + \Delta x])$be the probability mass of the ith interval
- Then the sum of those probability masses must be 1:

$$\sum_i P([x_i, x_i + \Delta x]) = 1$$

- We can rewrite the equation above in terms of the density of each interval, by dividing and multiplying by x:

$$\sum_i \frac{\Delta x * P([x_i, x_i + \Delta x])}{\Delta x} = 1$$

Properties of Probability Density Functions

- In the limit, as the interval width becomes infinitesimal, we denote:
  - Summation as $\int$ instead of $\sum$
- Then, the previous equation (in terms of density) can be rewritten as:

$$\sum_i \frac{\Delta x * P([x_i, x_i + \Delta x])}{\Delta x} = 1 \Rightarrow \int dx\, p(x) = 1$$

- We use $p(x)$ to represent the probability mass when x is discrete
- Thus, what $p(x)$ represents depends on the context
  - Is x discrete or continuous?

The Normal Probability Density Functions

- Perhaps the most famous probability density function is the normal distribution, also known as the Gaussian distribution
- The probability density function of normal distribution is

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

- Recall, what are σ and μ? what do they control?
- An example of the probability density is shown in the figure where the x axis is divided into a dense comb of small intervals
- The figure also shows that the area under the curve is, in fact, 1

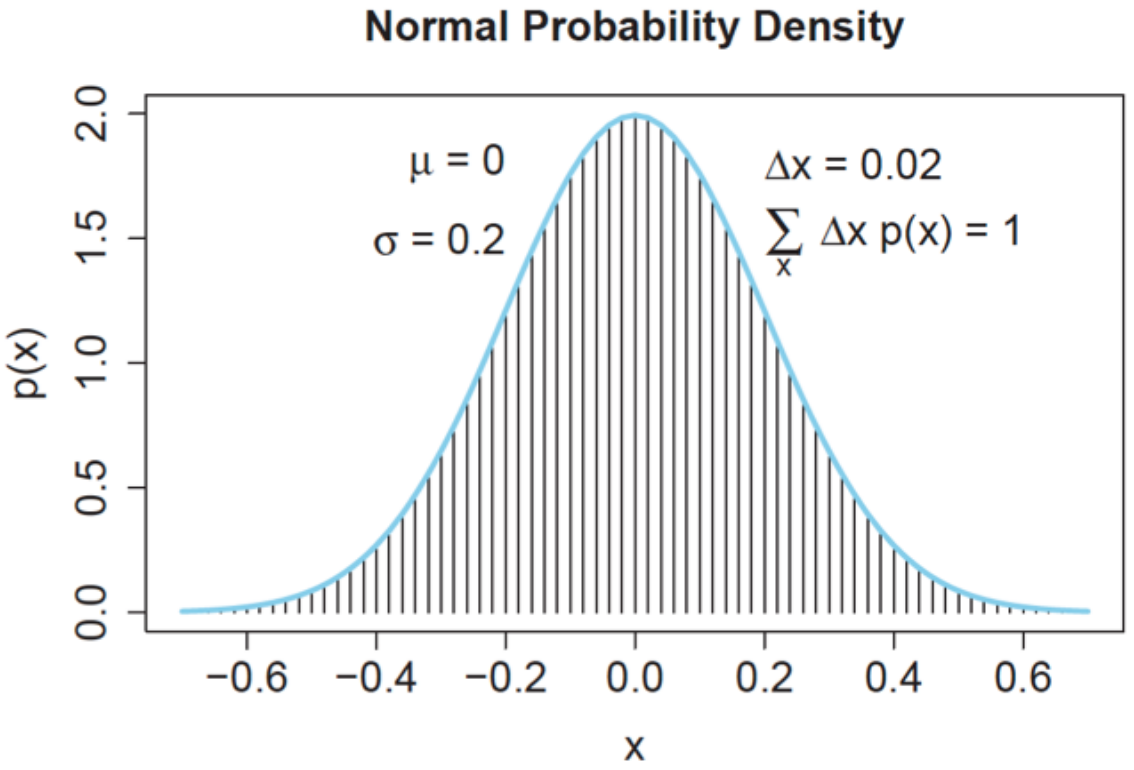
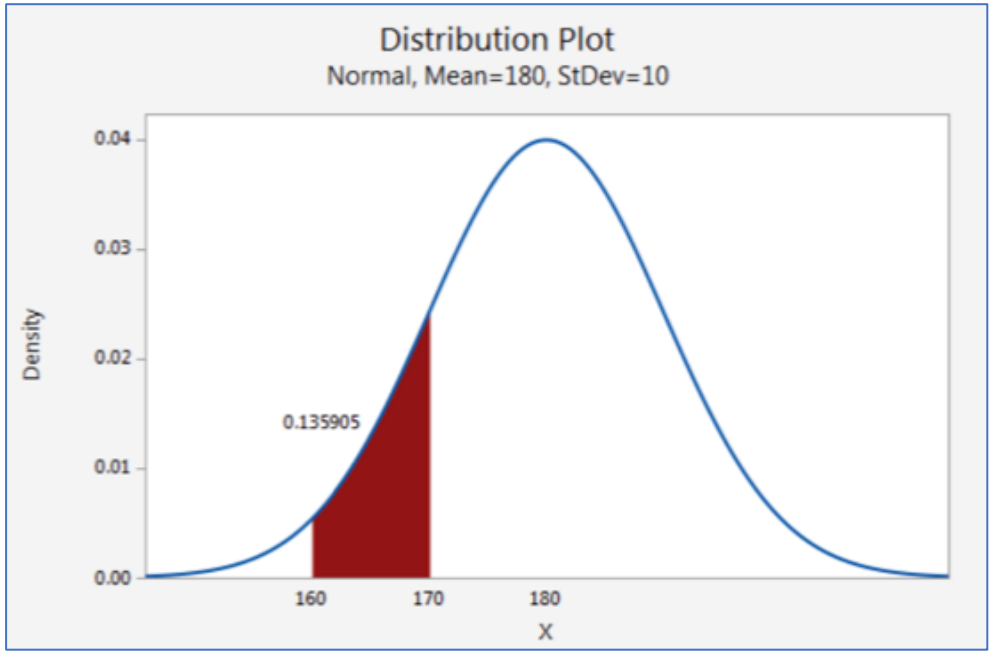


Figure 4.4: A normal probability density function, shown with a comb of narrow intervals. The integral is approximated by summing the width times height of each interval. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Example - Continuous Normal Distribution

- Example of the continuous distribution of weights
  - The continuous normal distribution can describe the distribution of weight of adult males.
  - For example, you can calculate the probability that a man weighs between 160 and 170 pounds.
  - The area of this range is 0.136; therefore, the probability that a randomly selected man weighs between 160 and 170 pounds is 13.6%.
  - The entire area under the curve equals 1.0

$$\int_{-\infty}^{+\infty} p(x)dx = 1$$



Random Variables/Distributions Exercise

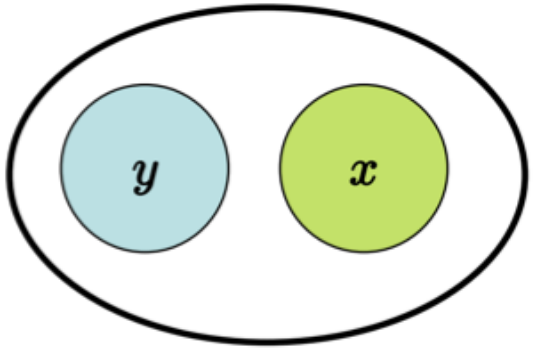
Bayes Rule (Just a preview)

- Bayes rule is merely the mathematical relation between the prior allocations of credibility and the posterior reallocation of credibility (conditional on data)

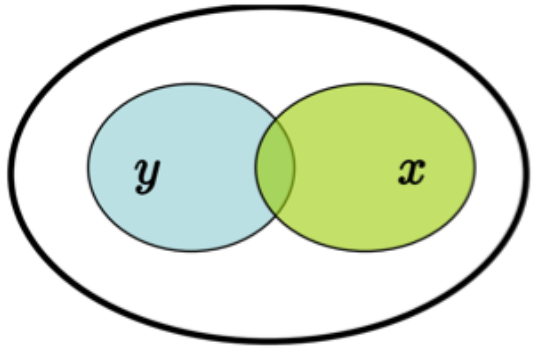
$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

Joint Probability

- Joint Probability
  - Knowing that y occurred reduces the sample space to y
    - The part of y where x also occurred, or the probability of x and y occurring, is:
    - $P(x, y) = P(x \cap y)$
  - Order does not matter:
    - $P(x, y) = P(y, x)$



- Disjoint Sets
- Mutually Exclusive Events
- $x \cap y = \emptyset$



- Intersecting sets

Joint Probability and Marginal Probability

- This table shows the probabilities of various combinations of people's eye/hair color
- Each entry indicates the **joint probability** of particular combinations of eye color (e) and hair color (h), denoted by $p(e, h)$
- The right margin of the table shows the probabilities of the eye colors overall, collapsed across hair colors
- Such probabilities are called **marginal probability**, denoted by $p(e)$:

$$p(e) = \sum_{h} p(e, h)$$

- The marginal probabilities of the hair colors, $p(h)$, are indicated on the lower margin of the table:

$$p(h) = \sum_{e} p(e, h)$$

Table 4.1: Proportions of combinations of hair color and eye color. Some rows or columns may not sum exactly to their displayed marginals because of rounding error from the original data. Data adapted from Snee (1974). Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Eye Color	Hair Color				Marginal
	Black	Brunette	Red	Blond	(Eye Color)
Brown	.11	.20	.04	.01	.37
Blue	.03	.14	.03	.16	.36
Hazel	.03	.09	.02	.02	.16
Green	.01	.05	.02	.03	.11
Marginal (Hair Color)	.18	.48	.12	.21	1.0

Conditional Probability

- Conditional Probability
  - $\mathrm{P}(x|y)$ is the probability of the occurrence of event x, given that y occurred is given as:
    - $\mathrm{P}(x|y) = \frac{P(x \cap y)}{P(y)} = \frac{P(x,y)}{P(y)}$
  - Answers the question:
    - How does the probability of an event change if we have extra information?

Table 4.2: Example of conditional probability. Of the blue-eyed people in Table 4.1, what proportion have hair color h? Each cell shows $p(h|\text{blue}) = p(\text{blue}, h)/p(\text{blue})$ rounded to two decimal points. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Eye Color	Hair Color				Marginal
	Black	Brunette	Red	Blond	(Eye Color)
Blue	.03/.36 = .08	.14/.36 = .39	.03/.36 = .08	.16/.36 = .45	.36/.36 = 1.0

Conditional Probability – Order Matters

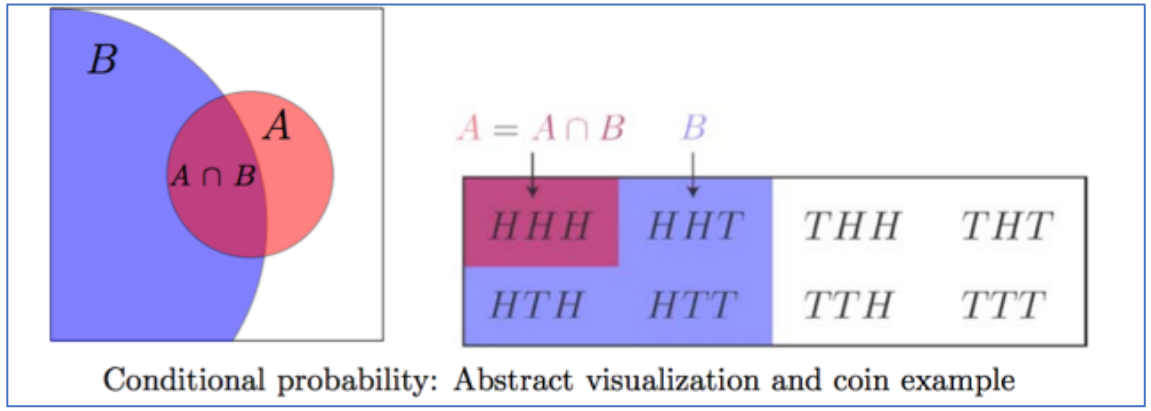
- $\mathrm{P}(x|y) \neq \mathrm{P}(y|x)$
  - Why?
- $\mathrm{P}(cute|puppy) \neq \mathrm{P}(puppy|cute)$

Conditional Probability Example

- Coin Toss Example:
  - Toss a fair coin 3 times
  - What is the probability of 3 heads?
    - Answer:
      - $Sample\ Space\ = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$
      - All outcomes are equally likely (if the coin is fair)
      - $P(HHH) = \frac{1}{8}$
  - Suppose we are told that the first toss was heads
  - Given this information, how should we compute the probability of {HHH}?
    - Answer:
      - We have a new (reduced) $Sample\ Space\ = \{HHH, HHT, HTH, HTT\}$
      - All outcomes are still equally likely (the coin is still fair)
      - $P(HHH) = \frac{1}{4}$

Conditional Probability Example

- We can visualize the conditional probability as follows
  - Think of $P(A)$ as the proportion of the area of the whole sample space taken up by A
  - For $P(A|B)$ we restrict our attention to B
  - $P(A|B)$ is the proportion of B taken up by A
    - $P(A|B) = \frac{P(A \cap B)}{P(B)}$



Conditional probability: Abstract visualization and coin example

Statistical Independence

- Independent Events
  - If x and y are independent then they are unconnected and not related to each other
  - We have:
    $\mathrm{P}(x|y) = P(x)$
  - From there it follows that
    $P(x, y) = P(x) * P(y)$
  - In other words, knowing that y occurred does not change the probability that x occurs (and vice versa)
  - Examples of absolute independence include:
    - Eye color and height
    - Hair color and weight

Statistical Independence Example

- Independent Events
  - If we want to calculate the joint probability of two independent events, we can simply multiply each probability together to get the joint probability
  - "Joint Distribution" = "Product Distribution"
  - $\mathrm{P}(x, y) = P(x) * P(y)$
- For Example:
  - Probability of tossing a coin and getting "Heads":

$$P(Heads) = P(x) = \frac{1}{2}$$

  - Probability of rolling a dice and getting "3":

$$P(Roll\ \text{“3”}) = P(y) = \frac{1}{6}$$

$$P(Heads) * P(Roll\ \text{“3”}) = \mathrm{P}(x, y) = \frac{1}{2} * \frac{1}{6} = \frac{1}{12}$$

Bayesian Inference

➢The Naïve Bayes Classifier relies on three things:

1. Independence assumption
2. The notion of conditional probability
3. **Bayesian Inference -** a method of statistical inference in which Bayes Theorem is used to update the probability for a hypothesis as more evidence becomes available.

Bayesian Inference Example

- We observe that the sidewalk is wet
- What are the possible causes?

Bayesian Inference Example

- We observe that the sidewalk is wet
- What are the possible causes?
  - It rained recently
  - Sprinkler
  - Broken water main (pipe)
  - Person spilled a drink
  - Dog marked his territory

Bayesian Inference Example

- Based on information that we have, we have some notion that certain probabilities are greater than others
- For example:
  - P(recent rain) > P(sprinkler)
  - P(recent rain) > P(spilled drink)
- Bayesian inference incorporates previous knowledge (prior probabilities)

Bayesian Inference Example

- Observation A:
  - Suppose we *observe* that the sidewalk is wet, in addition to the grass, the trees, the street, and the parked cars
  - How do the probabilities change given this new information?

Bayesian Inference Example

- Observation A:
  - Suppose we *observe* that the sidewalk is wet, in addition to the grass, the trees, the street, and the parked cars
  - How do the probabilities change given this new information?
  - P(recent rain | Observation A)

Bayesian Inference Example

- Observation A:
  - Suppose we *observe* that the sidewalk is wet, in addition to the grass, the trees, the street, and the parked cars
  - How do the probabilities change given this new information?
  - P(recent rain | Observation A) ↑

Bayesian Inference Example

- Observation A:
  - Suppose we *observe* that the sidewalk is wet, in addition to the grass, the trees, the street, and the parked cars
  - How do the probabilities change given this new information?
  - P(recent rain | Observation A) ↑
  - P(spilled drink | Observation A)

Bayesian Inference Example

- Observation A:
  - Suppose we *observe* that the sidewalk is wet, in addition to the grass, the trees, the street, and the parked cars
  - How do the probabilities change given this new information?
  - P(recent rain | Observation A) ↑
  - P(spilled drink | Observation A) ↓

Bayesian Inference Example

- Observation B:
  - Now suppose that instead we *observe* that the sidewalk is wet, but it is localized to a small area next to an empty water bottle
  - How do the probabilities change given this new information?

Bayesian Inference Example

- Observation B:
  - Now suppose that instead we *observe* that the sidewalk is wet, but it is localized to a small area next to an empty water bottle
  - How do the probabilities change given this new information?
  - P(recent rain | Observation B)

Bayesian Inference Example

- Observation B:
  - Now suppose that instead we *observe* that the sidewalk is wet, but it is localized to a small area next to an empty water bottle
  - How do the probabilities change given this new information?
  - P(recent rain | Observation B) ↓

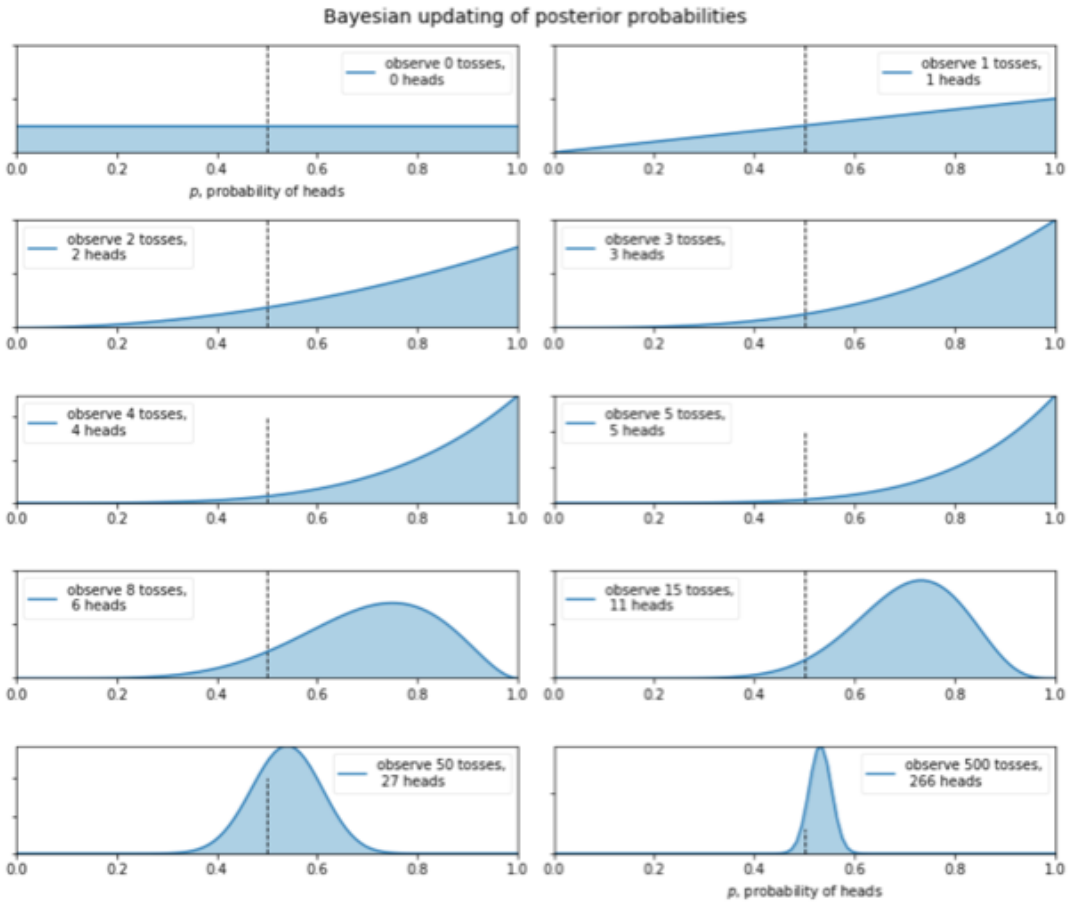
Bayesian Inference Example

- Observation B:
  - Now suppose that instead we *observe* that the sidewalk is wet, but it is localized to a small area next to an empty water bottle
  - How do the probabilities change given this new information?
  - P(recent rain | Observation B) ↓
  - P(spilled drink | Observation B)

Bayesian Inference Example

- Observation B:
  - Now suppose that instead we *observe* that the sidewalk is wet, but it is localized to a small area next to an empty water bottle
  - How do the probabilities change given this new information?
  - P(recent rain | Observation B) ↓
  - P(spilled drink | Observation B) ↑

Bayesian Updating



Applying Bayesian Inference to parameters/data

- Naïve Bayes is a machine learning method that can be used to predict the likelihood that an event will occur given evidence that is present in your data
- The Naïve Bayes Classifier relies on Bayesian inference at its core
- Makes two “Naïve” assumptions over attributes

Naïve Bayes Classifier Assumptions

- Fundamental assumption:
  - Each feature makes an **independent** and **equal** contribution to the outcome
    - We assume that no pair of features are dependent on one another in any way (complete independence)
      - Temperature has nothing to do with humidity, and has no effect on whether or not it is windy
    - Each feature is given the same weight/importance
      - We assume that none of the attributes is irrelevant, and that they are all contributing equally to the outcome.

Naïve Bayes Classifier

- To illustrate the inner workings of the Naïve Bayes Classifier, we will consider an example:
  - We have recorded weather features about the last 14 times that we played golf.
  - We also recorded the result of whether the conditions were favorable or not.
  - We will demonstrate how a Naïve Bayes Classifier can be used to determine whether or not the a specific set of weather conditions supports playing golf.

Naïve Bayes Classifier Example

- Weather Independent Variables:
  - Outlook
    - (Rainy, Overcast, Sunny)
  - Temperature
    - (Hot, Mild, Cool)
  - Humidity
    - (High, Normal)
  - Windy
    - (False, True)
- Weather Dependent Variable:
  - Play Golf
    - (YES, NO)

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

How does the Naïve Bayes Classifier work?

- Step 1:
  - Convert the data set into frequency tables
  - Use the frequency tables to calculate likelihood tables
- Step 2:
  - Use the product rule to obtain a joint conditional probability for the attributes
- Step 3:
  - Use Bayes Rule to calculate the posterior probability for each class variable
  - Once this has been done for all classes, output the class with the highest probability

Naïve Bayes Classifier Example

- Step 1:
  - Create a frequency table for the Class:
    - P(C) or P(CLASS)
    - P(YES)

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

- Step 1:
  - Create a frequency table for the Class:
    - P(C) or P(CLASS)
    - P(YES) = 9/14

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

- Step 1:
  - Create a frequency table for the Class:
    - P(C) or P(CLASS)
    - P(YES) = 9/14
    - P(NO) = 5/14

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

- Step 1:
  - Create a frequency table for the Class:
    - P(C) or P(CLASS)
    - P(YES) = 9/14
    - P(NO) = 5/14
- Class Frequency Table:

Play	
YES	9
NO	5
Total	**14**

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

How does the Naïve Bayes Classifier work?

- Step 1:
  - Convert the data set into frequency tables
  - Use the frequency tables to calculate likelihood tables
- Step 2:
  - Use the product rule to obtain a joint conditional probability for the attributes
- Step 3:
  - Use Bayes Rule to calculate the posterior probability for each class variable
  - Once this has been done for all classes, output the class with the highest probability

Naïve Bayes Classifier Example

- Step 1:
  - P(C) or P(CLASS)
  - P(YES) = 9/14 = 0.643
  - P(NO) = 5/14 = 0.357
- The Class Frequency Table can be used to create the Class Likelihood Table by dividing each class frequency by the total (relative probability)
- Likelihood Table:

Play		P(YES)
YES	9	9/14 = 0.643
NO	5	5/14 = 0.357
Total	**14**	**14/14 = 100%**

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

- The next step is to repeat this process for each weather feature and compute their corresponding Feature Frequency Tables
- These Frequency tables can then be used to create Likelihood tables as previously demonstrated for each feature (weather condition) in our dataset

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

➢**Outlook** Frequency and Likelihood

Outlook				
	YES	NO	P(YES)	P(NO)
Sunny	2	3	2/9 = 0.222	3/5 = 0.6
Overcast	4	0	4/9 = 0.444	0/5 = 0
Rainy	3	2	3/9 = 0.33	2/5 = 0.4
Total	9	5	9/9 = 1	5/5 = 1

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

➢**Temperature** Frequency and Likelihood

Temperature				
	YES	NO	P(YES)	P(NO)
Hot	2	2	2/9 = 0.22	2/5 = 0.4
Mild	4	2	4/9 = 0.44	2/5 = 0.4
Cool	3	1	3/9 = 0.33	1/5 = 0.2
Total	9	5	9/9 = 1	5/5 = 1

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

➢ **Humidity** Frequency and Likelihood

Humidity				
	YES	NO	P(YES)	P(NO)
High	3	4	3/9 = 0.33	4/5 = 0.8
Normal	6	1	6/9 = 0.66	1/5 = 0.2
Total	9	5	9/9 = 1	5/5 = 1

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Naïve Bayes Classifier Example

➢ **Windy** Frequency and Likelihood

Windy				
	YES	NO	P(YES)	P(NO)
FALSE	6	2	6/9 = 0.66	2/5 = 0.4
TRUE	3	3	3/9 = 0.33	3/5 = 0.6
Total	9	5	9/9 = 1	5/5 = 1

	outlook	temperature	humidity	windy	play_golf
0	Rainy	Hot	High	False	NO
1	Rainy	Hot	High	True	NO
2	Overcast	Hot	High	False	YES
3	Sunny	Mild	High	False	YES
4	Sunny	Cool	Normal	False	YES
5	Sunny	Cool	Normal	True	NO
6	Overcast	Cool	Normal	True	YES
7	Rainy	Mild	High	False	NO
8	Rainy	Cool	Normal	False	YES
9	Sunny	Mild	Normal	False	YES
10	Rainy	Mild	Normal	True	YES
11	Overcast	Mild	High	True	YES
12	Overcast	Hot	Normal	False	YES
13	Sunny	Mild	High	True	NO

Bayes Rule applied to data

- Bayes rule is merely the mathematical relation between the prior allocations of credibility and the posterior reallocation of credibility (conditional on data)

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

Bayes Rule Explained

Posterior Probability of class (C) given predictor (X)

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

HPSI
Human Performance Systems, Inc.

Bayes Rule Explained

Posterior Probability of class (C) given predictor (X)

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

This represents the probability of C being true, provided X is true

Bayes Rule Explained

Likelihood - the conditional probability of the predictor-given the class

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

HPSI
Human Performance Systems, Inc.

Bayes Rule Explained

Likelihood - the conditional probability of the predictor-given the class

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

This represents the probability of X being true provided C is true

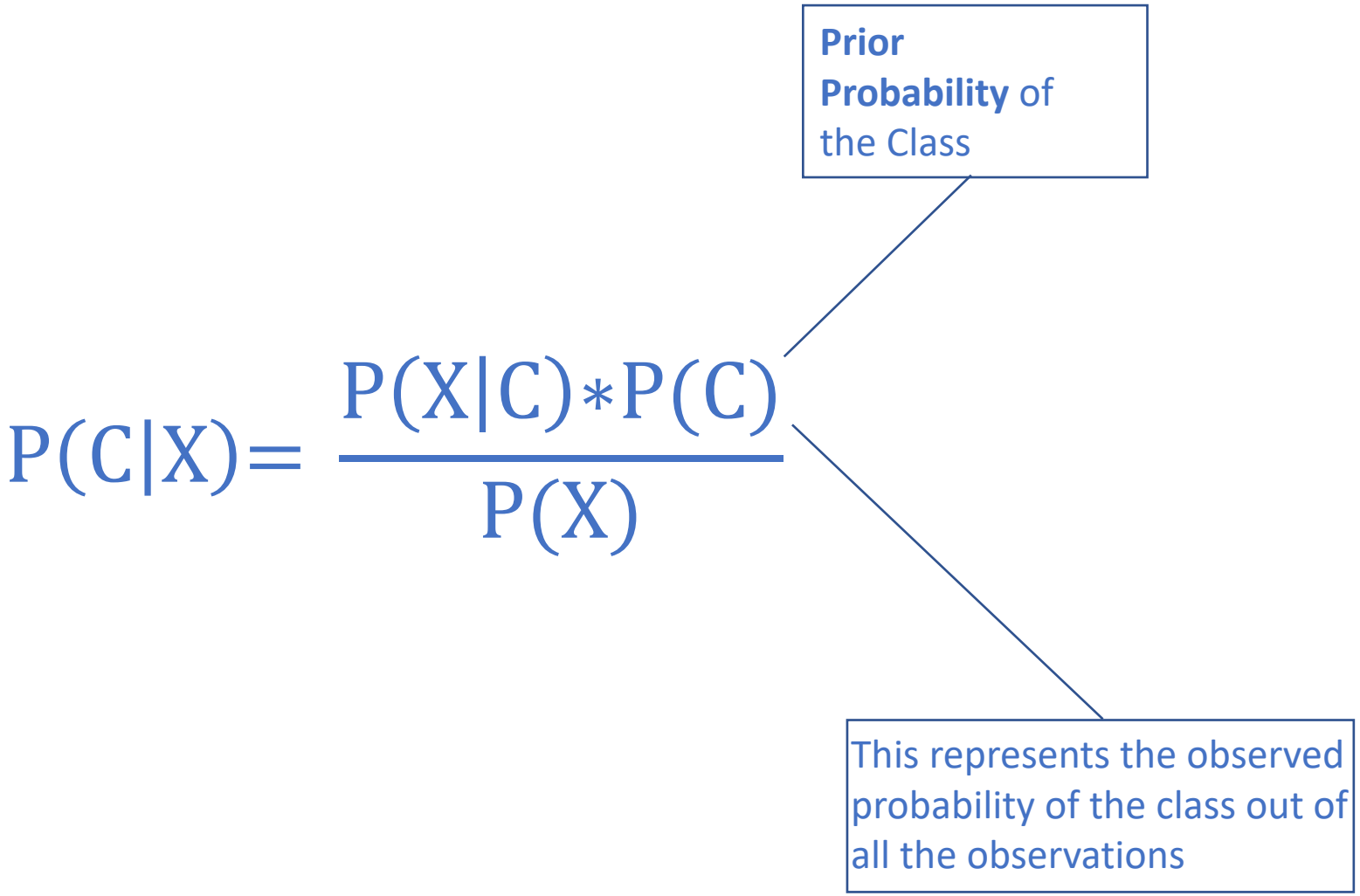
HPSI
Human Performance Systems, Inc.

Bayes Rule Explained

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

Prior Probability of the Class

Bayes Rule Explained



HPSI
Human Performance Systems, Inc.

Bayes Rule Explained

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

Prior Probability of the Predictor

Bayes Rule Explained

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

Prior Probability of the Predictor

This represents the observed probability of the predictor out of all the observations

How does the Naïve Bayes Classifier work?

- Step 1:
  - Convert the data set into frequency tables
  - Use the frequency tables to calculate likelihood tables
- Step 2:
  - Use the product rule to obtain a joint conditional probability for the attributes
- Step 3:
  - Use Bayes Rule to calculate the posterior probability for each class variable
  - Once this has been done for all classes, output the class with the highest probability

Bayes Rule Explained

Prior Probability of the Predictor can be estimated directly by multiplying the individual relative frequencies of each predictor due to the naive independence assumption

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

Bayes Rule Explained

Prior Probability of the Predictor can be estimated directly by multiplying the individual relative frequencies of each predictor (due to the naive independence assumption)

Key Idea: the "naive assumption" allows us to multiply the probabilities

$$P(C|X) = \frac{[P(X_1|C) * P(X_2|C) * P(X_3|C) \ldots * P(X_n|C)] * P(C)}{P(X)}$$

HPSI
Human Performance Systems, Inc.

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we should golf or not

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

$$P(C|X) = \frac{[P(X_1|C) * P(X_2|C) * P(X_3|C) \ldots * P(X_n|C)] * P(C)}{P(X)}$$

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we should golf or not

Outlook	Temperature	Humidity	Windy?	Play?
X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE	YES

$$P(YES|X) = \frac{[P(Sunny|YES) * P(Cool|YES) * P(High|YES) * P(TRUE|YES)] * P(YES)}{P(X)}$$

Naïve Bayes Classifier Example

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

- Outlook = Sunny
  - P(Sunny|YES)=0.22
  - P(Sunny|NO)=0.6

Outlook				
	YES	NO	P(YES)	P(NO)
Sunny	2	3	**2/9 = 0.222**	**3/5 = 0.6**
Overcast	4	0	4/9 = 0.444	0/5 = 0
Rainy	3	2	3/9 = 0.33	2/5 = 0.4
Total	9	5	9/9 = 1	5/5 = 1

Naïve Bayes Classifier Example

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

- Temperature = Cool
  - P(Cool|YES) = 0.33
  - P(Cool|NO) = 0.2

	Temperature			
	YES	NO	P(YES)	P(NO)
Hot	2	2	2/9 = 0.22	2/5 = 0.4
Mild	4	2	4/9 = 0.44	2/5 = 0.4
Cool	3	1	**3/9 = 0.33**	**1/5 = 0.2**
Total	9	5	9/9 = 1	5/5 = 1

Naïve Bayes Classifier Example

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

- Humidity = High
  - P(High|YES) = 0.33
  - P(High|NO) = 0.8

Humidity				
	YES	NO	P(YES)	P(NO)
High	3	4	**3/9 = 0.33**	**4/5 = 0.8**
Normal	6	1	6/9 = 0.66	1/5 = 0.2
Total	9	5	9/9 = 1	5/5 = 1

Naïve Bayes Classifier Example

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

- Windy = True
  - P(TRUE|YES) = 0.33
  - P(TRUE|NO) = 0.6

Windy				
	YES	NO	P(YES)	P(NO)
FALSE	6	2	6/9 = 0.66	2/5 = 0.4
TRUE	3	3	**3/9 = 0.33**	**3/5 = 0.6**
Total	9	5	9/9 = 1	5/5 = 1

Naïve Bayes Classifier Example

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

- Play = YES
  - P(C) = P(YES) = 0.64
  - P(C) = P(NO) = 0.36

Play		P(YES)
YES	9	**9/14 = 0.64**
NO	5	**5/14 = 0.36**
Total	14	14/14 = 100%

How does the Naïve Bayes Classifier work?

- Step 1:
  - Convert the data set into frequency tables
  - Use the frequency tables to calculate likelihood tables
- Step 2:
  - Use the product rule to obtain a joint conditional probability for the attributes
- Step 3:
  - Use Bayes Rule to calculate the posterior probability for each class variable
  - Once this has been done for all classes, output the class with the highest probability

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

$$P(YES|X) = \frac{[P(Sunny|YES) * P(Cool|YES) * P(High|YES) * P(TRUE|YES)] * P(YES)}{P(X)}$$

HPSI
Human Performance Systems, Inc.

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

$$\mathrm{P(YES|X)} = \frac{[0.22 * 0.33 * 0.33 * 0.33] * 0.64}{\mathrm{P(X)}}$$

HPSI
Human Performance Systems, Inc.

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

$$P(YES|X) = \frac{0.0053}{P(X)}$$

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

$$P(NO|X) = \frac{[P(Sunny|NO) * P(Cool|NO) * P(High|NO) * P(TRUE|NO)] * P(NO)}{P(X)}$$

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

$$P(NO|X) = \frac{[0.6 * 0.2 * 0.8 * 0.6] * 0.36}{P(X)}$$

HPSI
Human Performance Systems, Inc.

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

$$P(NO|X) = \frac{0.0206}{P(X)}$$

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

Likelihood of playing golf under these conditions:

$$P(YES|X) = \frac{0.0053}{P(X)}$$

Likelihood of *NOT* playing golf under these conditions:

$$P(NO|X) = \frac{0.0206}{P(X)}$$

HPSI
Human Performance Systems, Inc.

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

Likelihood of playing golf under these conditions:

$$P(YES|X) = \frac{0.0053}{P(X)}$$

Probability of Play = YES:

$$P(YES|X) = \frac{0.0053}{(0.0053 + 0.0206)} = 20.5\%$$

Likelihood of *NOT* playing golf under these conditions:

$$P(NO|X) = \frac{0.0206}{P(X)}$$

Probability of Play = NO:

$$P(NO|X) = \frac{0.0206}{(0.0053 + 0.0206)} = 79.5\%$$

Naïve Bayes Classifier Example

Lets consider the following case, and try to predict whether we can golf or not

	Outlook	Temperature	Humidity	Windy?
X	X_1= Sunny	X_2 = Cool	X_3 = High	X_4 = TRUE
P(X|YES)	0.22	0.33	0.33	0.33
P(X|NO)	0.6	0.2	0.8	0.6

	Play?
C	YES
P(YES)	0.64
P(NO)	0.36

Likelihood of playing golf under these conditions:

$$P(YES|X) = \frac{0.0053}{P(X)}$$

Probability of Play = YES:

$$P(YES|X) = \frac{0.0053}{(0.0053 + 0.0206)} = 20.5\%$$

Likelihood of *NOT* playing golf under these conditions:

$$P(NO|X) = \frac{0.0206}{P(X)}$$

Probability of Play = NO:

$$P(NO|X) = \frac{0.0206}{(0.0053 + 0.0206)} = 79.5\%$$

HPSI
Human Performance Systems, Inc.

Naïve Bayes Classifier Results

- Probability of Play = YES
  - 0.0053/(0.0053 + 0.0206) = 20.5%

- Probability of Play = NO
  - 0.0206 / (0.0053 + 0.0206) = 79.5%

Naïve Bayes Classifier Results

The Naïve Bayes Classifier predicts that we should **not** play golf under the conditions in question

Outlook	Temperature	Humidity	Windy?	Play?
Sunny	Cool	High	TRUE	YES

HPSI
Human Performance Systems, Inc.

Naïve Bayes Exercise

Properties of Bayes Classifiers

- Incrementality:
  - With each training example, the prior and the likelihood can be updated dynamically (Flexible)
  - Combines prior knowledge and observed data:
  - Prior probability of a hypothesis multiplied with the probability of the hypothesis given the training data
- Probabilistic hypotheses:
  - Outputs not only a classification, but a probability distribution over all classes
- Performs well with multi-class prediction
- Meta-classification:
  - The outputs of several classifiers can easily be combined (ensembled)
    - E.g. by multiplying the probabilities that all classifiers predict for a given class

Improving Naïve Bayes

- If continuous features do not have normal distribution, we should use transformation or different methods to convert it in normal distribution.
- If test data set has zero frequency issue, apply smoothing techniques “Laplace Correction” to predict the class of test data set
- Remove correlated features, as the highly correlated features are voted twice in the model and it can lead to over inflating importance
- Not about tuning parameters
  - Naive Bayes classifiers have few parameters to tune
  - Preprocessing is more effective

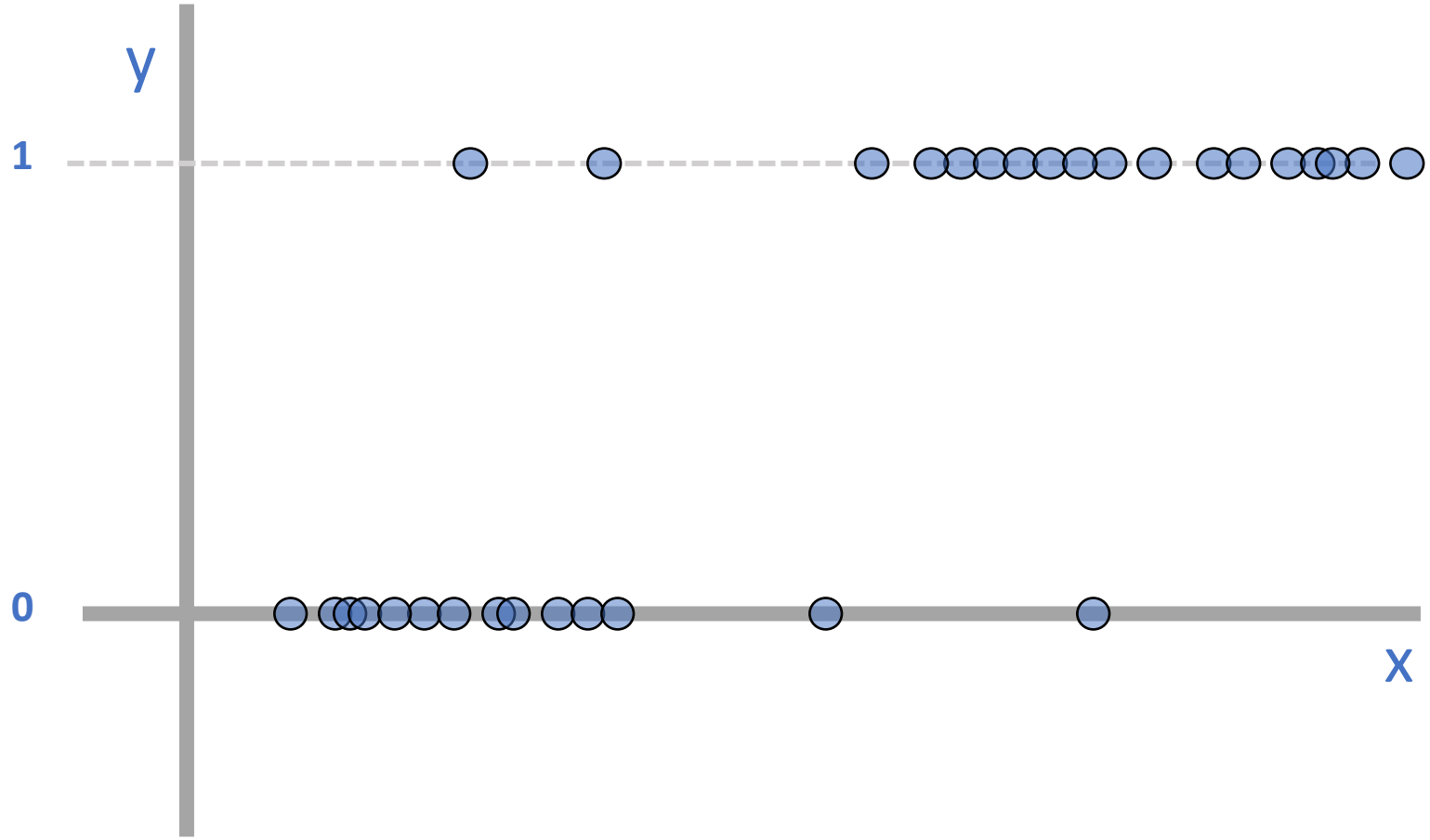
Logistic Regression

Logistic Regression

- When we think of linear regression, we usually think of predicting a numeric quantity
- Logistic Regression is somewhat misleading in that it serves primarily as a binary classification model
- Categorical response (y) with 2 levels (binary: 0 and 1)
  - Passing or failing a test
  - Surviving a plane crash or not
  - Hospitalisation required or not
  - Diagnosis of diabetes (yes / no)
  - Labelling (over/under some threshold)
- Predictor variables (x_i) can take on any form: binary, categorical, and/or continuous

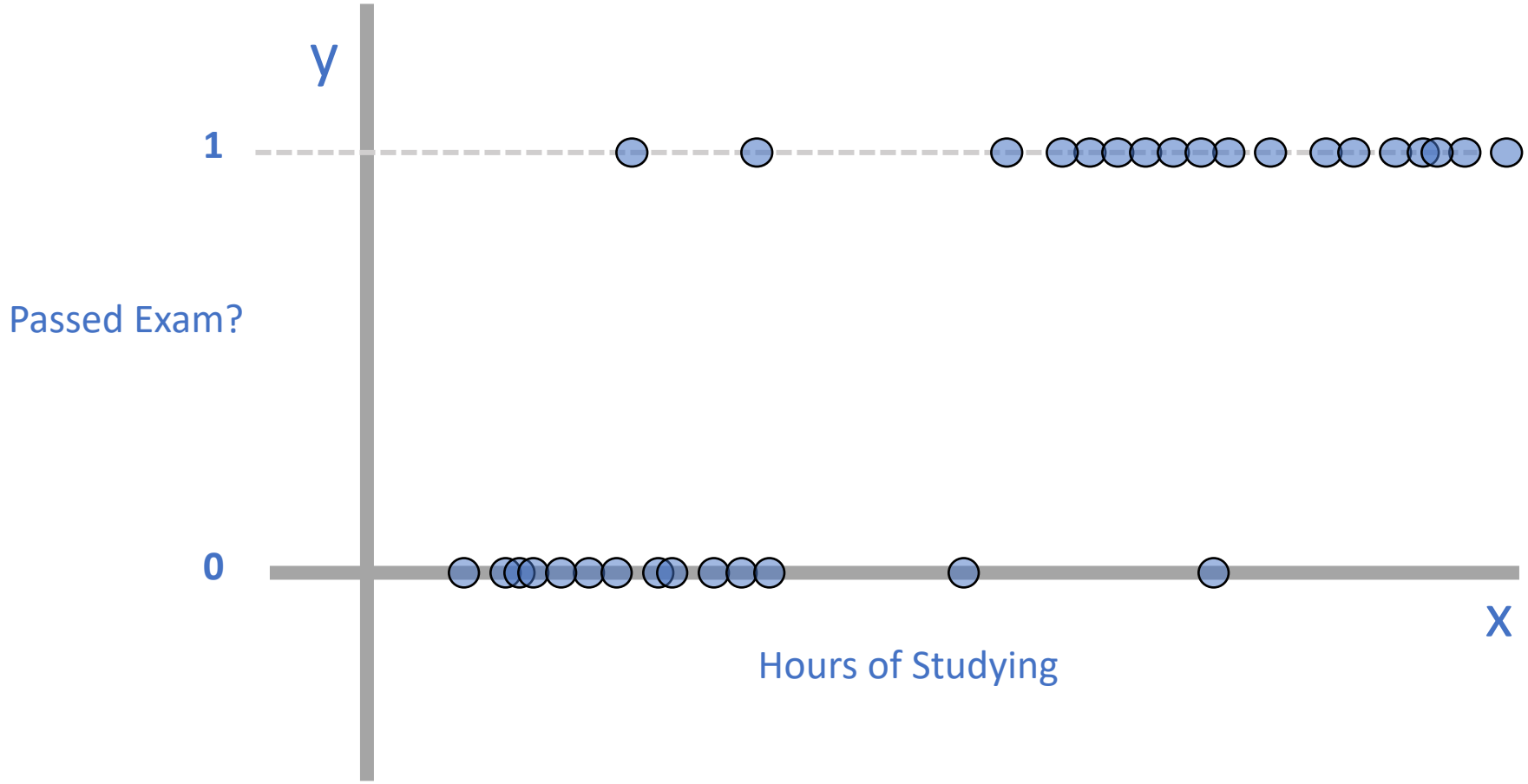
Logistic Regression Classification

Consider this set of binary data

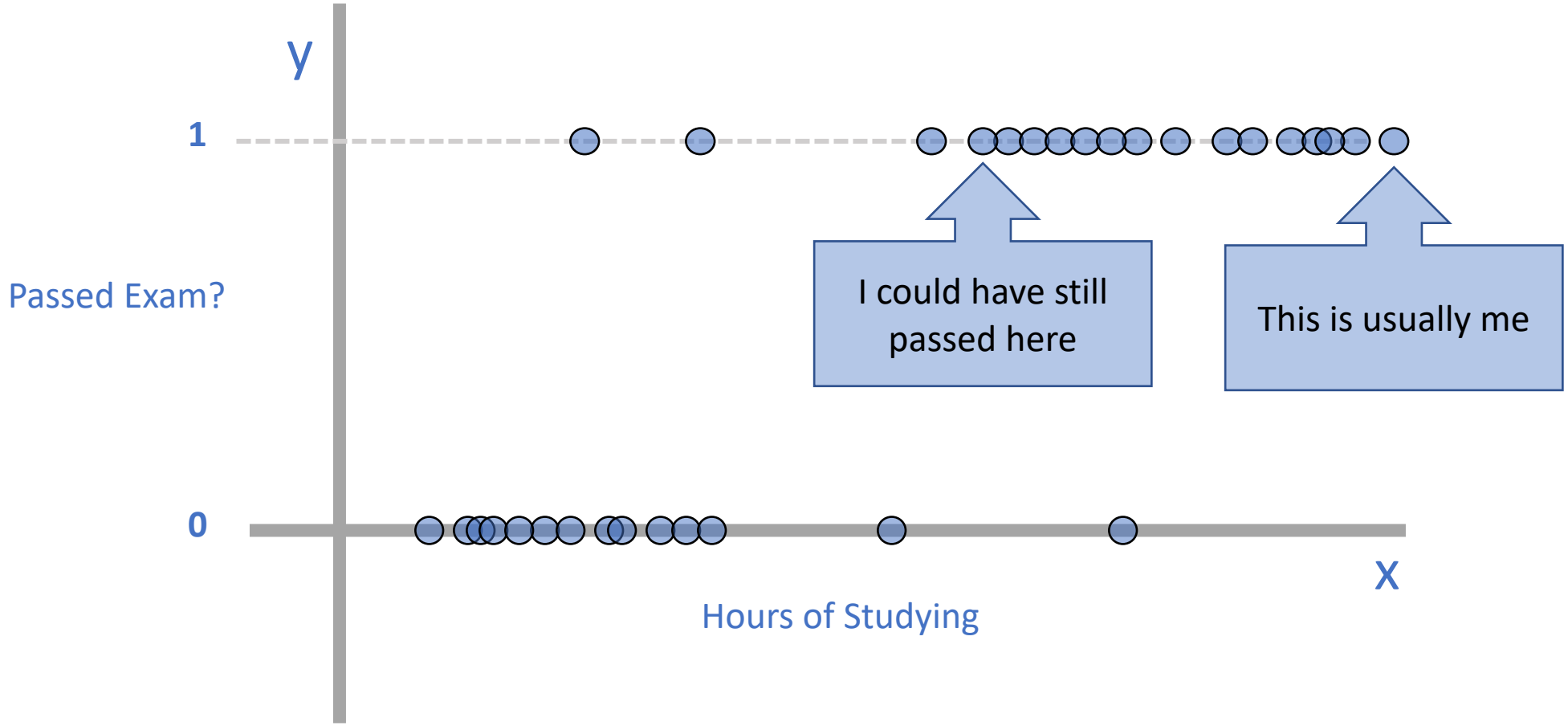


HPSI
Human Performance Systems, Inc.

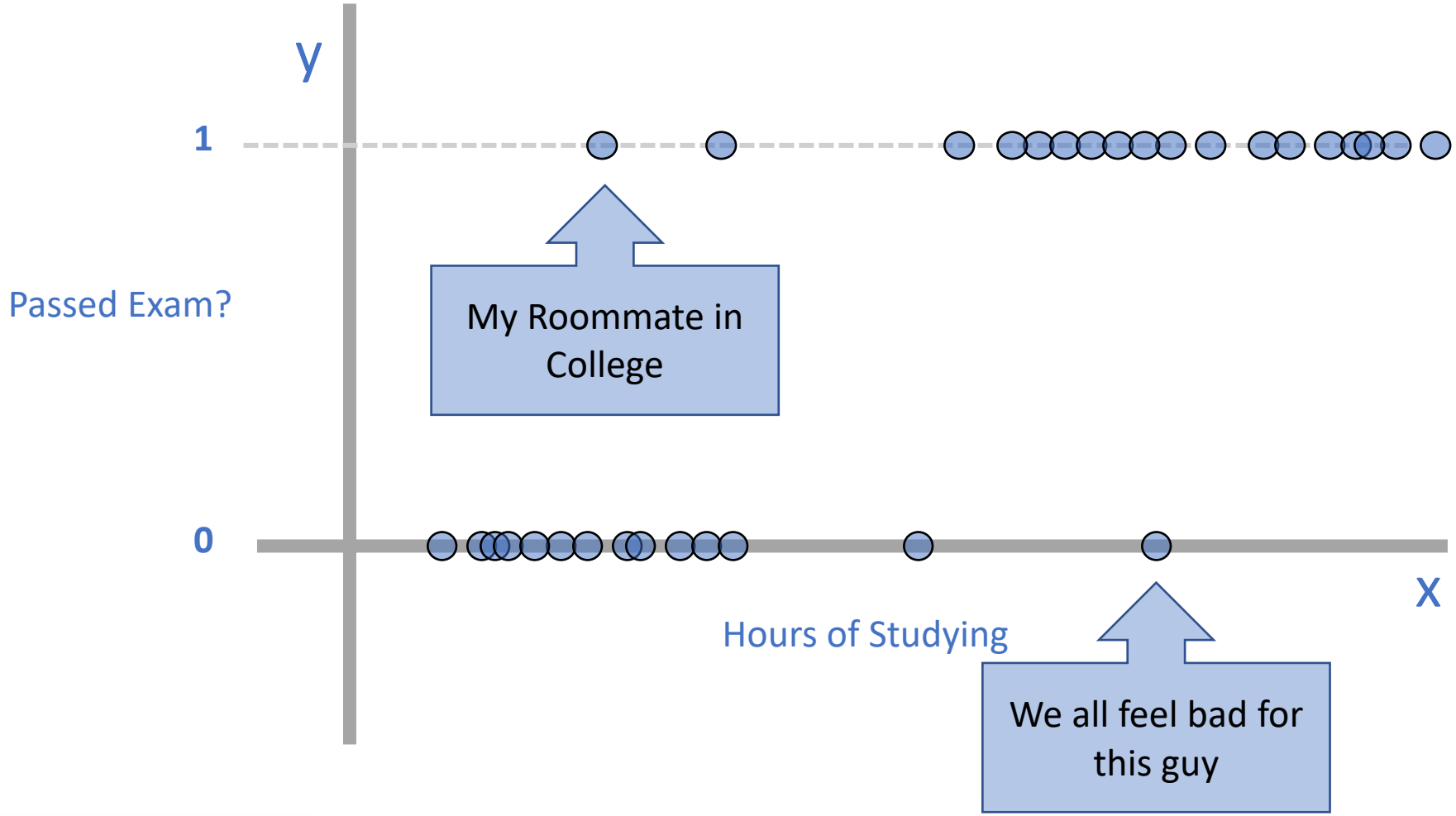
Consider this set of binary data



Consider this set of binary data

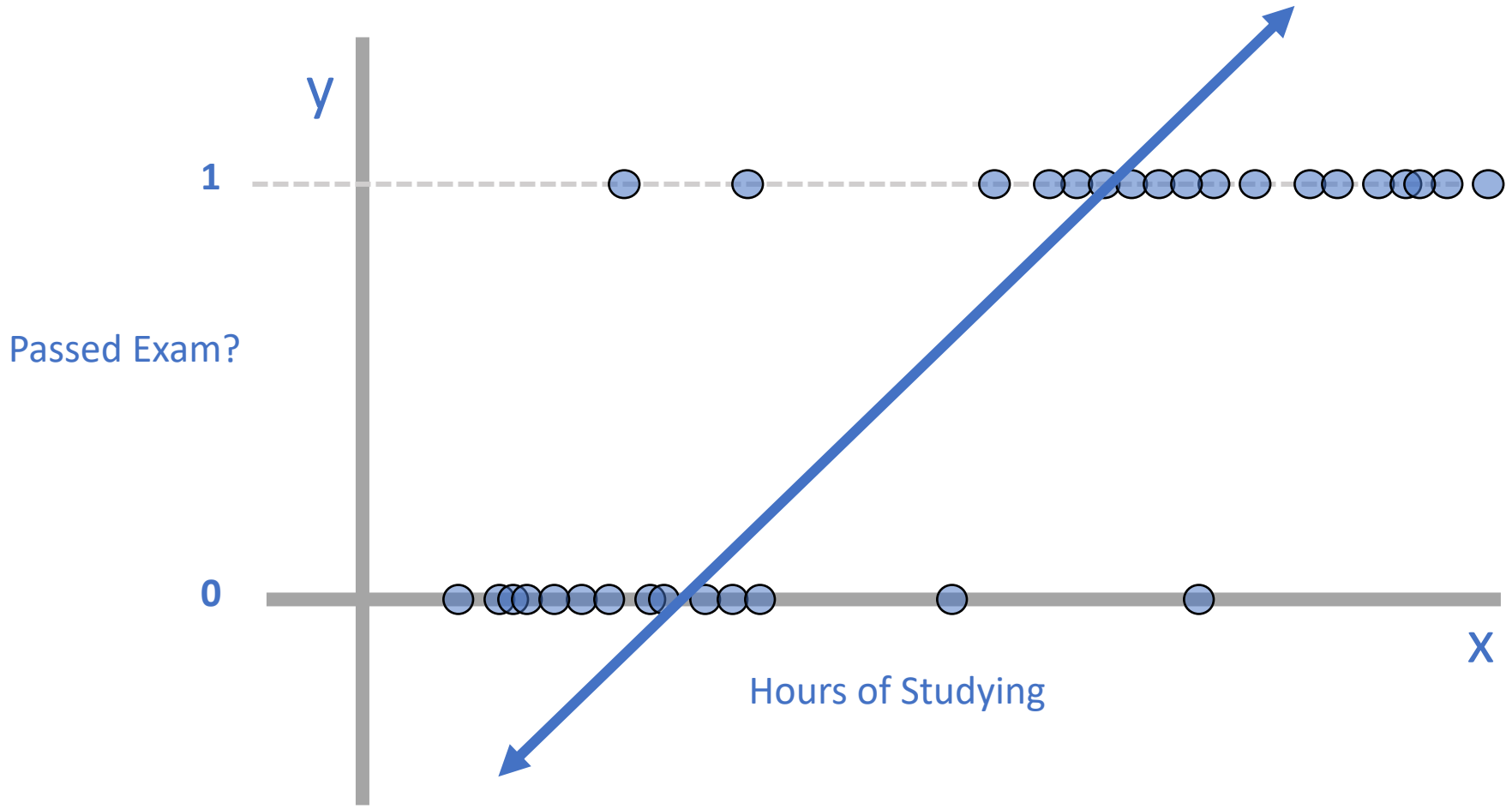


Consider this set of binary data



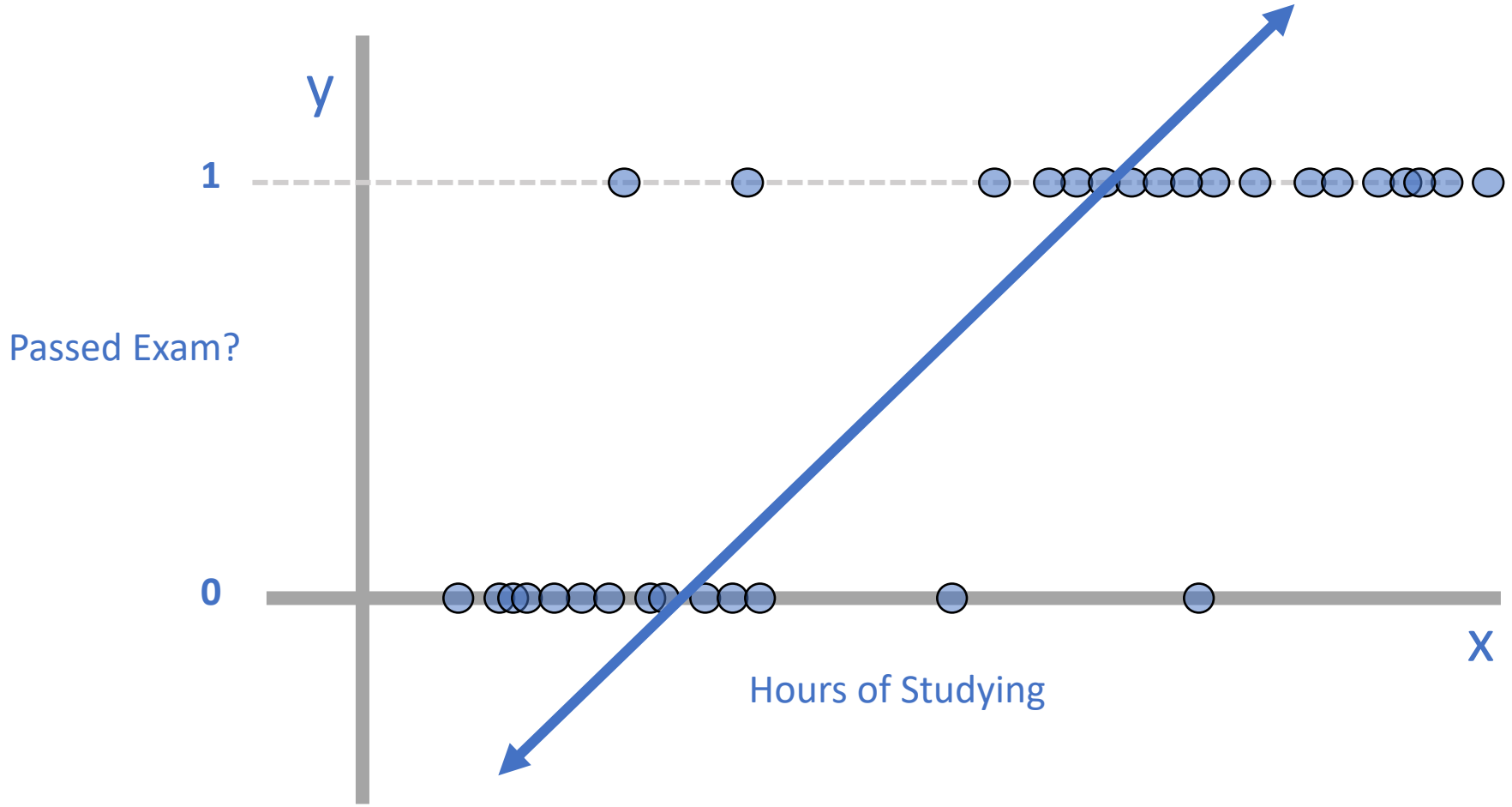
Logistic Regression Classification

- Linear Model? – Aside from being binary, there's nothing special about (y)



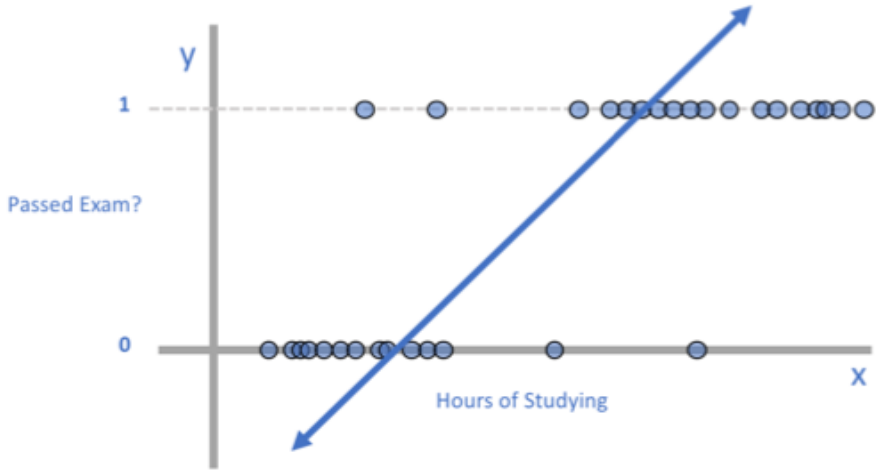
Logistic Regression Classification

- The value of "Passed Exam" is higher if a student studies more



Logistic Regression Classification

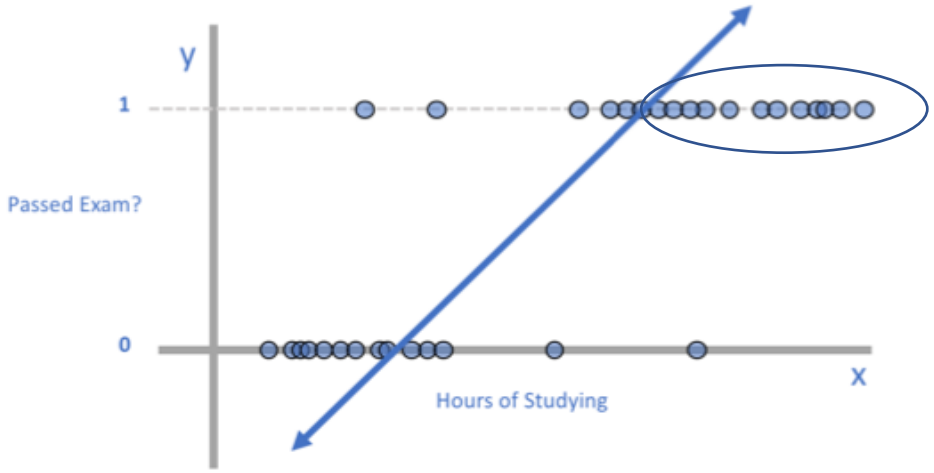
- Linear Model
  - $Pass = \beta_0 + \beta_1 hours\ of\ studying + \varepsilon$
- Problem
  - We want to see what makes the dependent variable change from a 0 to a 1
  - This can also be interpreted as what increases the likelihood of passing, or $P(pass = 1)$ which we can simply denote as p.
  - We should then be able to re-write the linear model as
  - $P(pass = 1) = p = \beta_0 + \beta_1 hours\ of\ studying + \varepsilon$
  - Every additional hour of studying increases the probability of passing by $x\%$



Logistic Regression Classification

$$P(pass = 1) = \beta_0 + \beta_1 hours\ of\ studying + \varepsilon$$

- Recall
  - Probabilities are bounded by $0 \leq p \leq 1$
  - If we were to look at students who studied this many hours, our model would predict a probability greater than 1

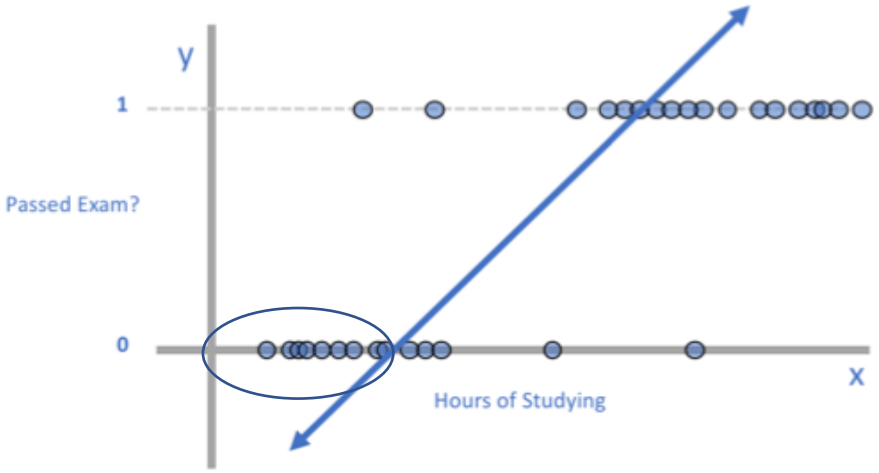


HPSI
Human Performance Systems, Inc.

Logistic Regression Classification

$$P(pass = 1) = \beta_0 + \beta_1 hours\ of\ studying + \varepsilon$$

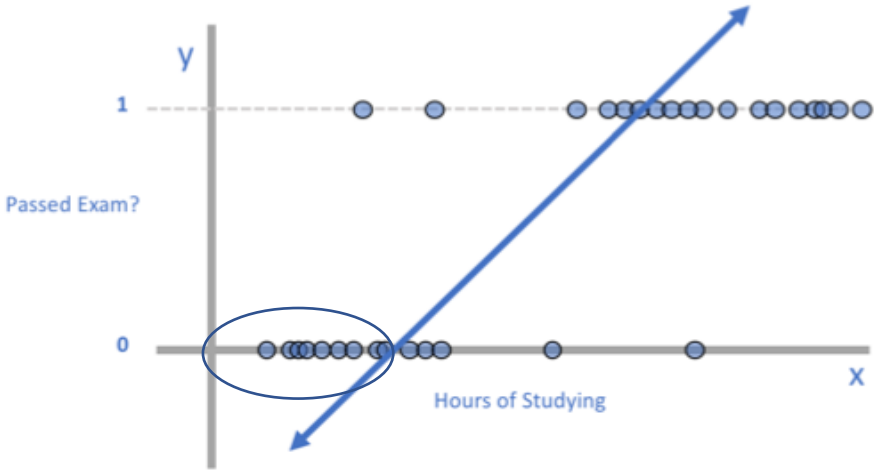
- Recall
  - Probabilities are bounded by $0 \leq p \leq 1$
  - If we were to look at students who studied this many hours, our model would predict a probability greater than 1
  - Likewise, here our model would predict a probability less than 0



Logistic Regression Classification

$$P(pass = 1) = \beta_0 + \beta_1 hours\ of\ studying + \varepsilon$$

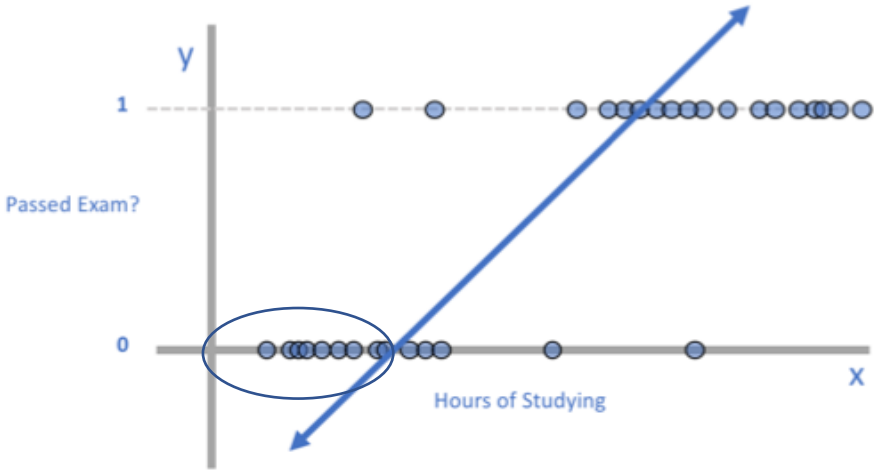
- Recall
  - Probabilities are bounded by $0 \leq p \leq 1$
  - If we were to look at students who studied this many hours, our model would predict a probability greater than 1
  - Likewise, here our model would predict a probability less than 0
- In addition to violating the laws of probability, our model would not maintain normally distributed residuals (which is another requirement of a linear regression model)



Logistic Regression Classification

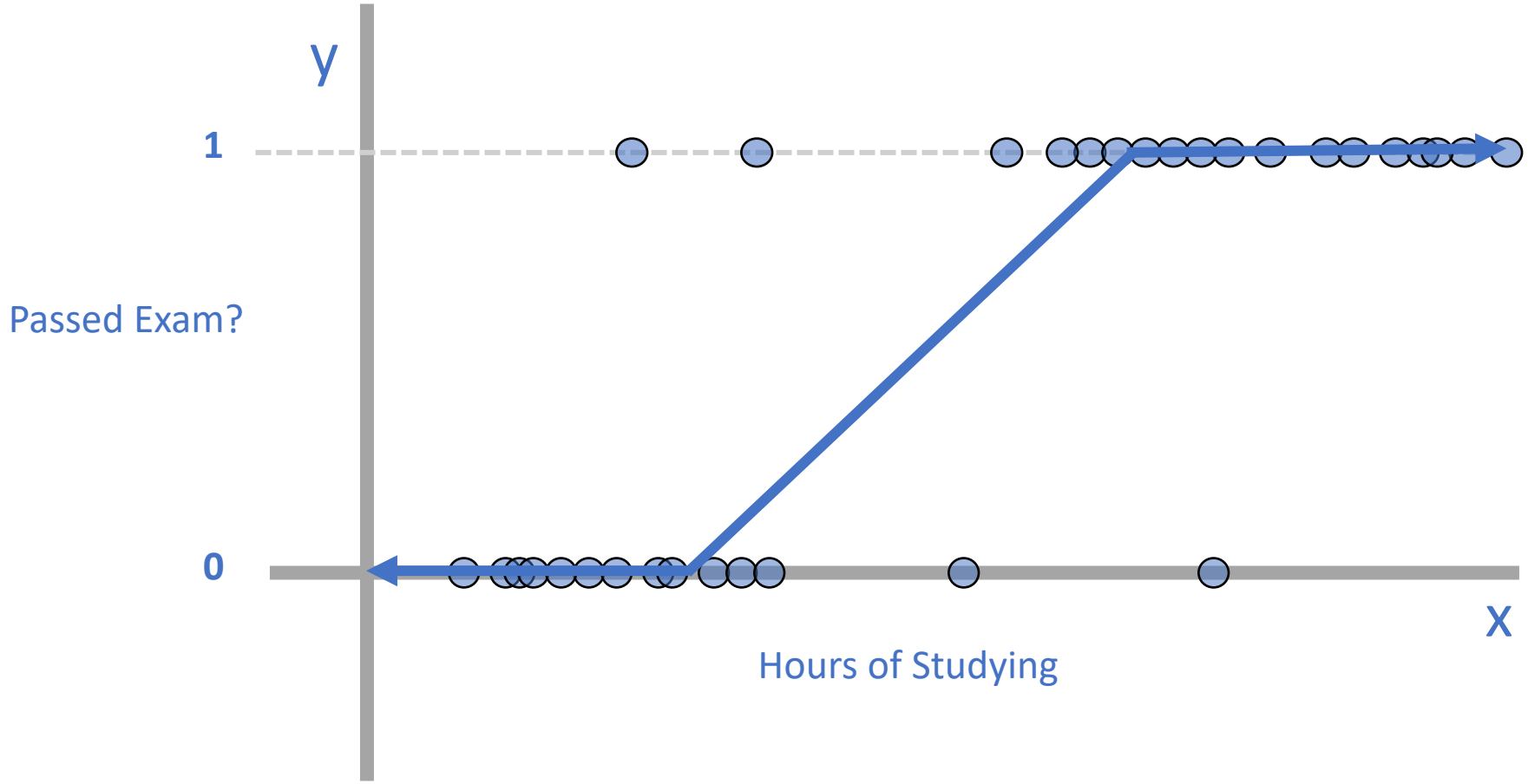
$$P(pass = 1) = \beta_0 + \beta_1 hours\ of\ studying + \ \varepsilon$$

- Recall
  - Probabilities are bounded by $0 \leq p \leq 1$
  - If we were to look at students who studied this many hours, our model would predict a probability greater than 1
  - Likewise, here our model would predict a probability less than 0
- In addition to violating the laws of probability, our model would not maintain normally distributed residuals (which is another requirement of a linear regression model)
- What can we do to fix this?



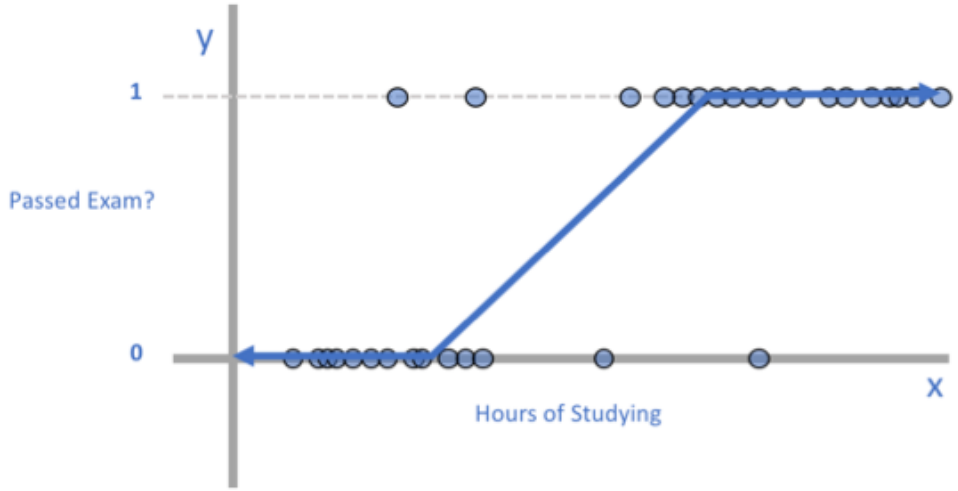
Logistic Regression Classification

- We could cap the probabilities at 0 and 1



Logistic Regression Classification

- Fixing the prior approach
  - We need to somehow constrain p such that $0 \leq p \leq 1$
  - We know $P(pass) = f(hours\ of\ studying)$ but the linear function didn't work
  - Let's try to develop a new function $f(hours\ of\ studying)$ that satisfies this criteria



Logistic Regression Classification

- Two Steps
  1. It must always be positive (since $0 \leq p(pass)$
     - $|x|$?
     - x^2 ?
     - What about $p(pass) = e^{\beta_0+\beta_1*hours\ of\ studying}$?
     - This works, but there are times when it would be greater than 1
  2. It must always be less than 1 ($p(pass) \leq 1$)
     - If you think about proportions, any number that is divided by a number slightly greater than it will give us a number smaller than 1
     - What if we just add 1 to the denominator?
     - $p(pass) = \frac{(e^{\beta_0+\beta_1*hours\ of\ studying})}{(e^{\beta_0+\beta_1*hours\ of\ studying})+1}$
     - Note that we could have added any small number (ε) and this condition would have been met, but we use 1 for reasons that will become clear shortly

Logistic Regression Classification

- The previous expression:

$$p(pass) = p = \frac{(e^{\beta_0 + \beta_1 * hours\ of\ studying})}{(e^{\beta_0 + \beta_1 * hours\ of\ studying}) + 1}$$

- After applying some algebra, can be re-written as:

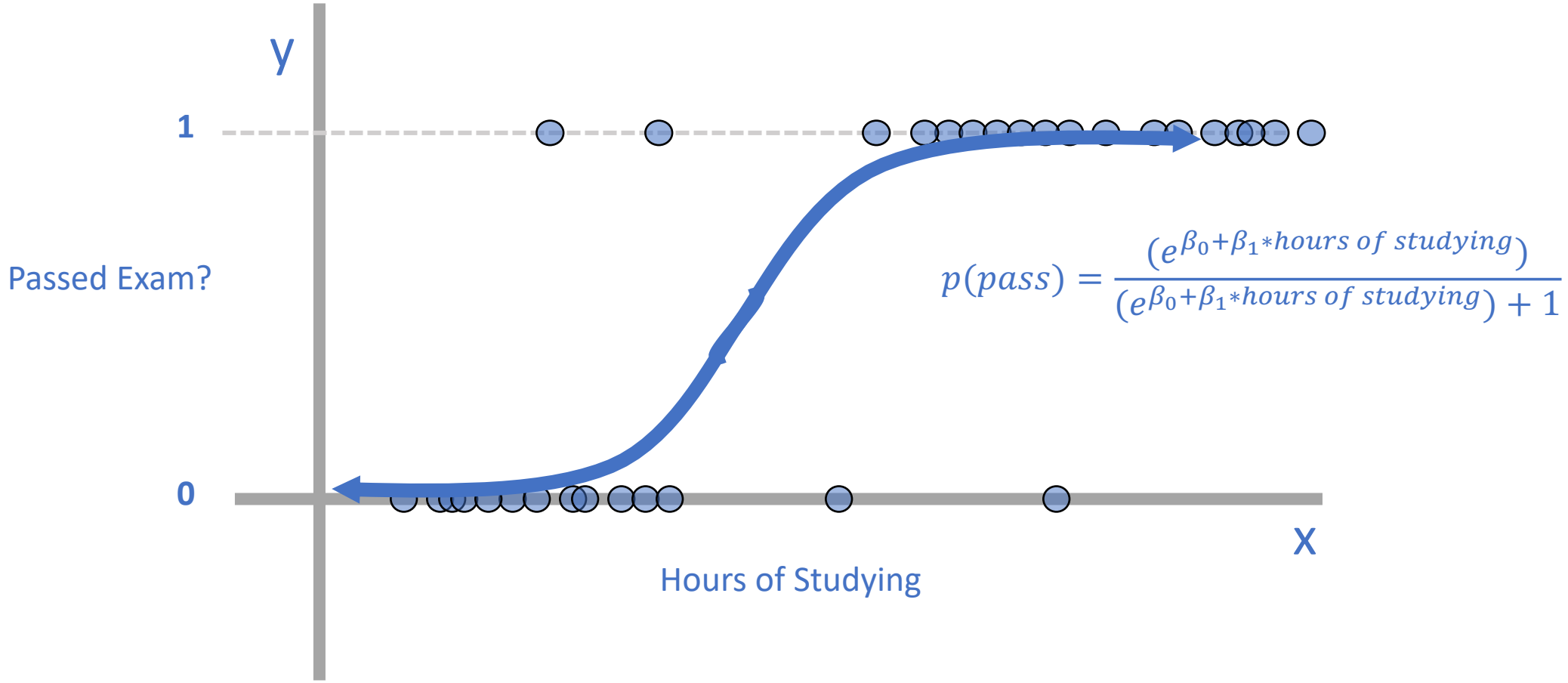
$$\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 * hours\ of\ studying$$

Logistic Regression Classification

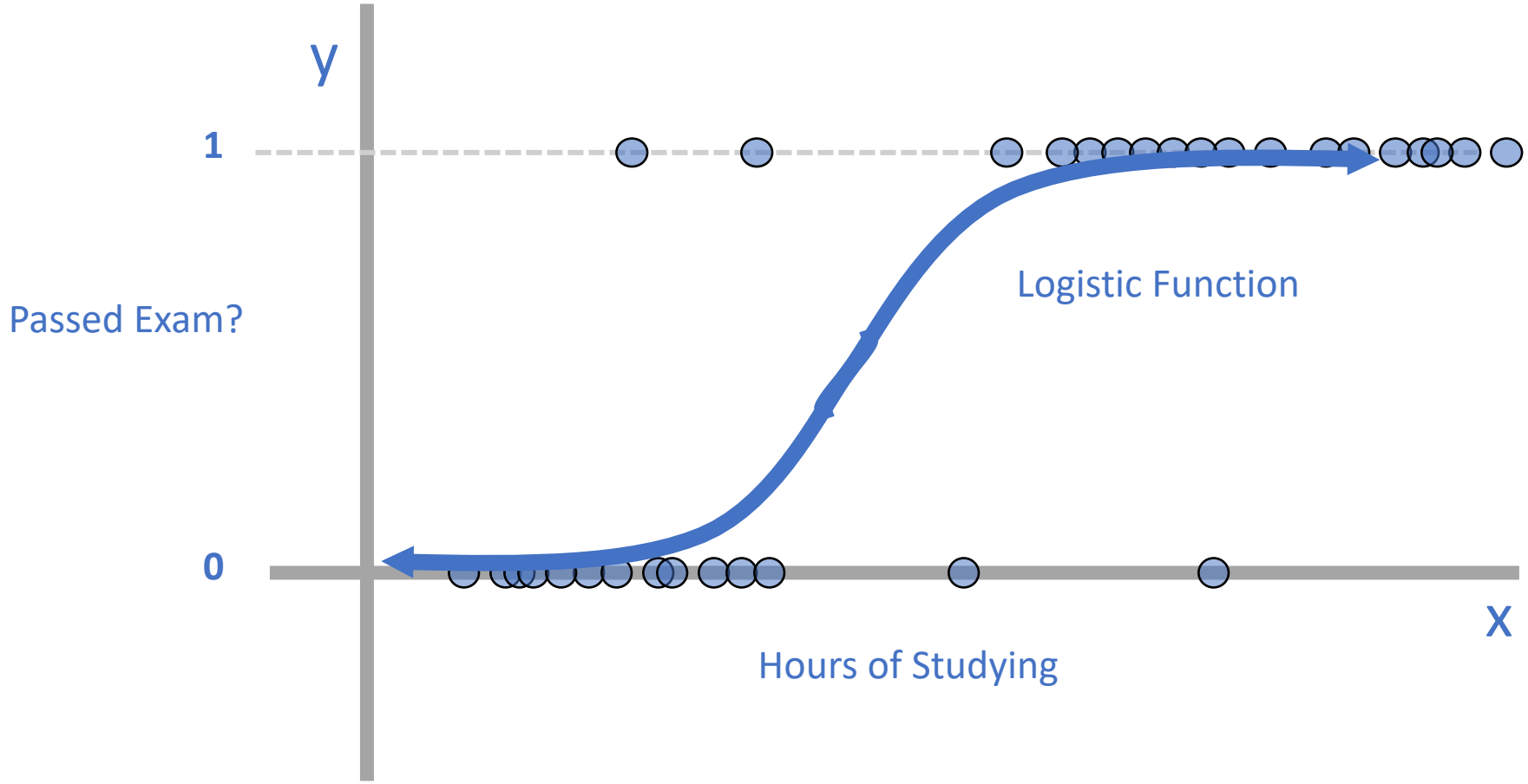
$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 * hours\ of\ studying$$

- Does this look familiar?
  - Yes!
  - It's in the form of a standard linear model
- So, even though the probability of a student passing is not a linear function of study-hours, the simple transformation is a linear function of study-hours
- This is the equation used in logistic regression

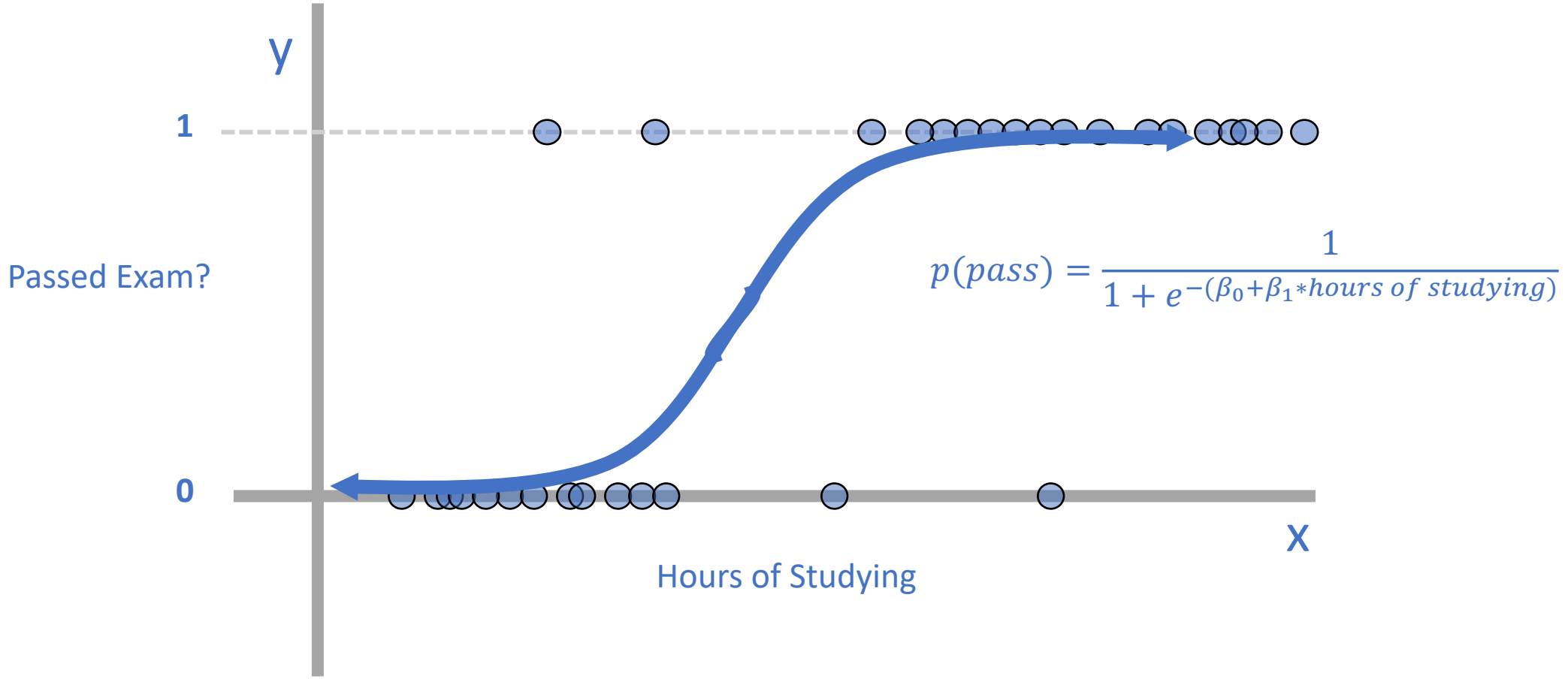
Logistic Regression



Logistic Regression

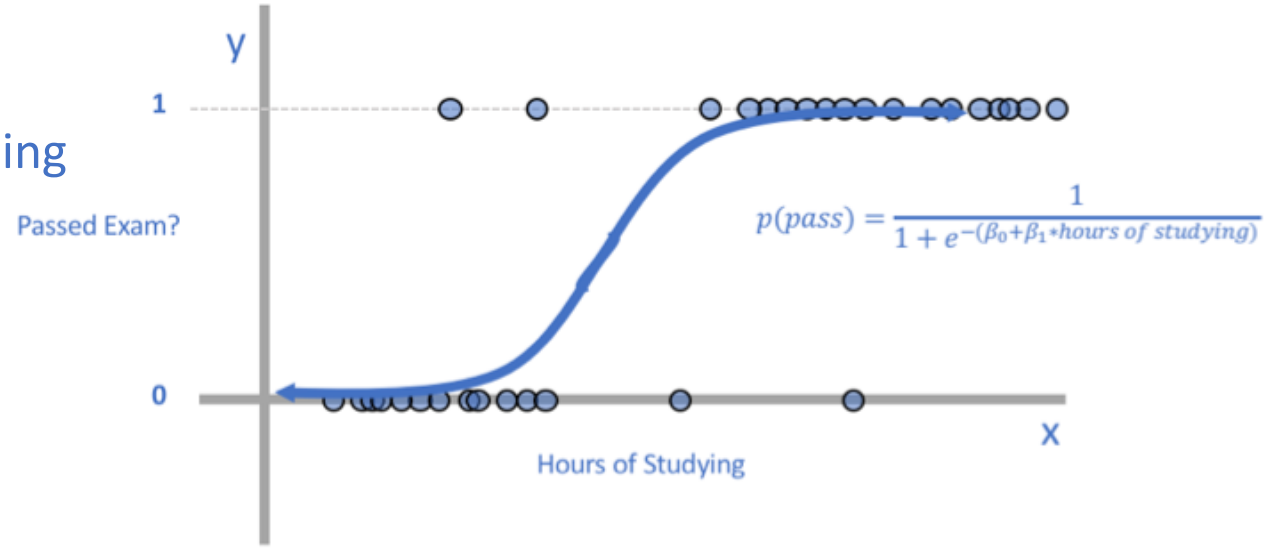


Logistic Regression

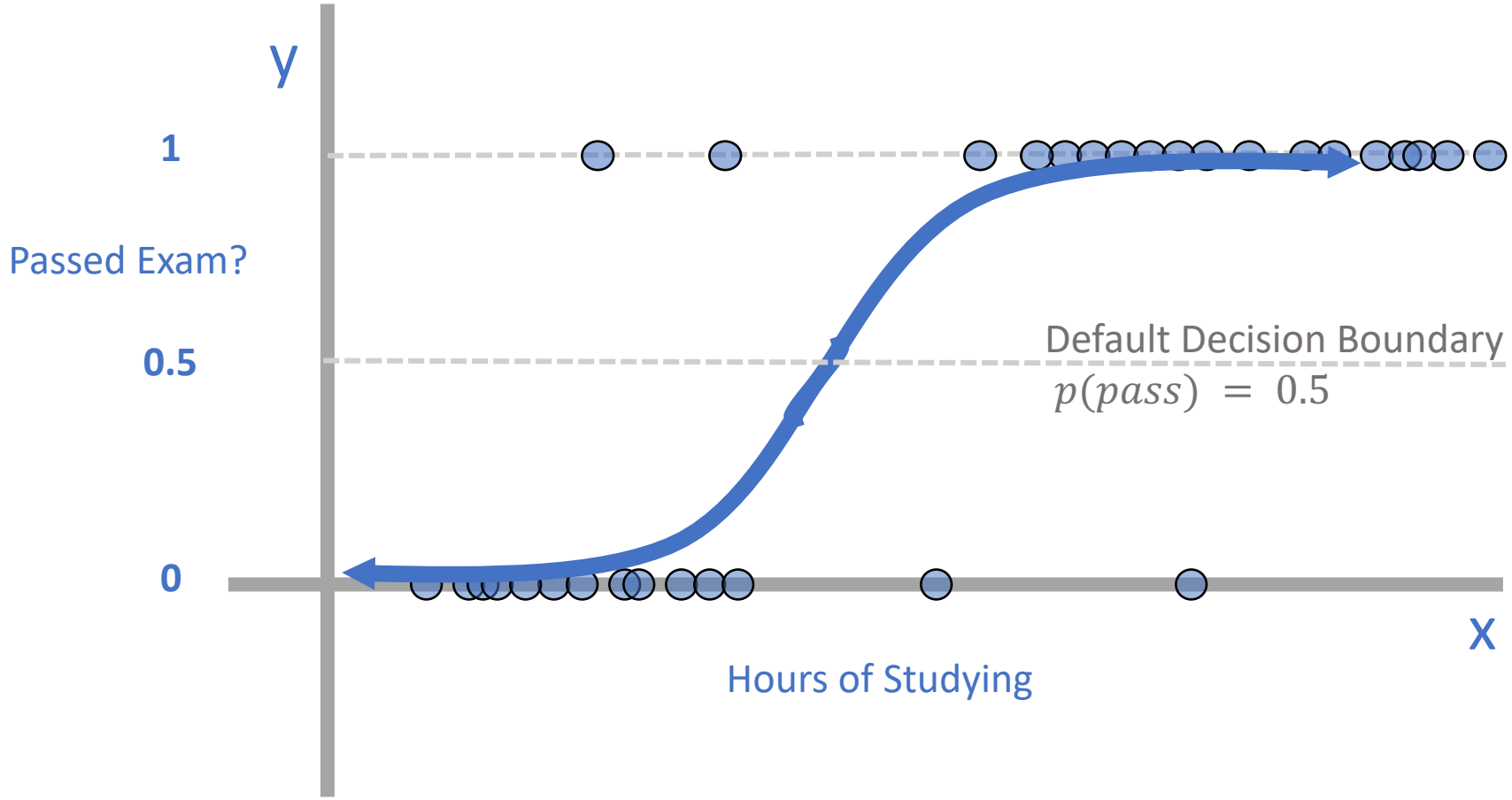


Logistic Regression Classification

- Note that the probability of passing is now between 0 and 1
  - $0 \leq p \leq 1$
- We now have a continuous function
- As study-hours approach 0, the probability of passing goes (asymptotically) to zero
- As study-hours approach infinity, probability of passing goes (asymptotically) to 1

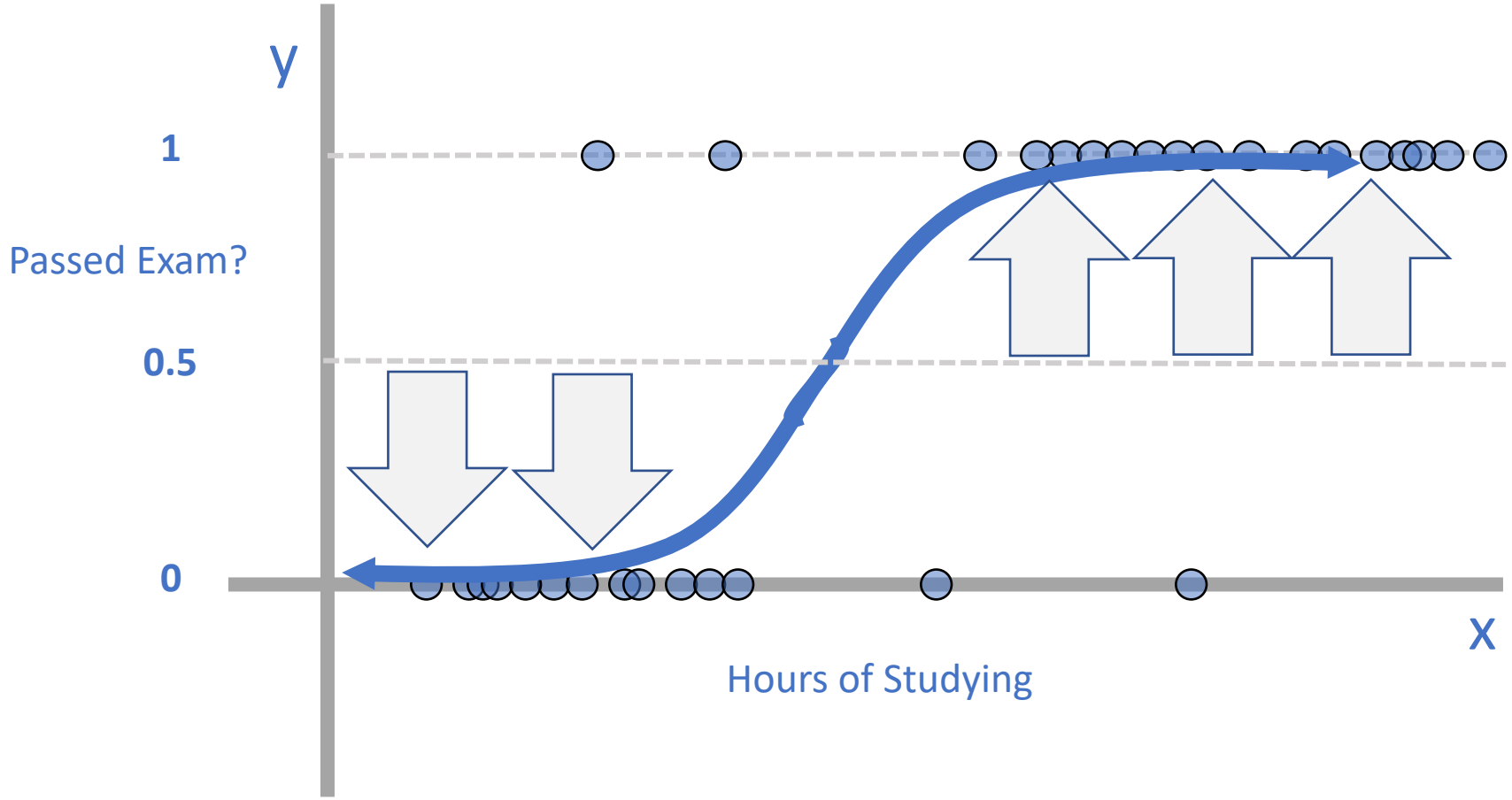


Decision Boundary



Logistic Regression Classification

Decision Boundary



Interpreting Binary Logistic Regression Model Output

- Interpreting the coefficients from a Logistic Regression model is different from a Linear Regression model
- Consider these results from a fitted logistic regression model
  - $\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 * study\_hrs$
  - $\beta_0 = -2.524$
  - $\beta_1 = 0.614$

Interpreting Binary Logistic Regression Model Output

- Interpreting the coefficients from a Logistic Regression model is different from a Linear Regression model
- Consider these results from a fitted logistic regression model
  - $\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 * study\_hrs$
  - $\beta_0 = -2.524$
  - $\beta_1 = 0.614$
  - $\ln\left(\frac{p}{1-p}\right) = -2.524 + 0.614 * study\_hrs$

Interpreting Binary Logistic Regression Model Output

- Interpreting the coefficients from a Logistic Regression model is different from a Linear Regression model
- Consider these results from a fitted logistic regression model
  - $\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 * study\_hrs$
  - $\beta_0 = -2.524$
  - $\beta_1 = 0.614$
  - $\ln\left(\frac{p}{1-p}\right) = -2.524 + 0.614 * study\_hrs$
  - For every additional unit increase in $study\_hrs$, $\ln\left(\frac{p}{1-p}\right)$ increases by 0.614 units?

Interpreting Binary Logistic Regression Model Output

- Interpreting the coefficients from a Logistic Regression model is different from a Linear Regression model
- Consider these results from a fitted logistic regression model
  - $\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 * study\_hrs$
  - $\beta_0 = -2.524$
  - $\beta_1 = 0.614$
  - $\ln\left(\frac{p}{1-p}\right) = -2.524 + 0.614 * study\_hrs$
  - For every additional unit increase in $study\_hrs$, $\ln\left(\frac{p}{1-p}\right)$ increases by 0.614 units?
  - But what does that mean?

Interpreting Binary Logistic Regression Model Output

- If we have

$$y^* = logit = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 * study\_hrs$$

- y^* is called the "logit" function
  - A logit is defined as the log base e (log) of the odds
- Exponentiating both sides of equation results in:

$$\frac{p}{1-p} = e^{(\beta_0 + \beta_1 * study\_hrs)}$$

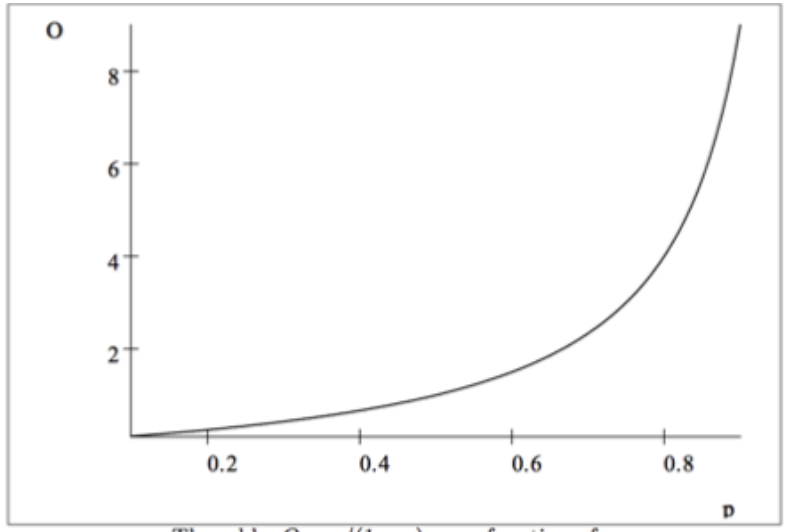
- We can exponentiate each of the coefficients to generate their corresponding odds ratios

Interpreting Binary Logistic Regression Model Output

- Odds

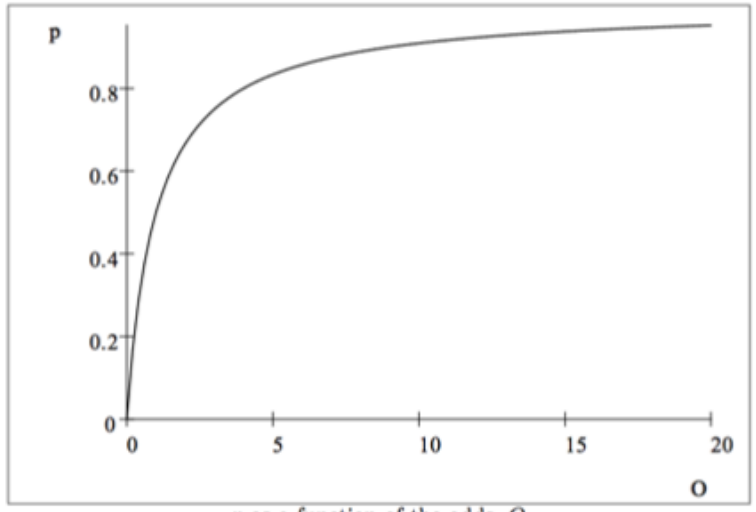
$$Odds(O) = \frac{P}{1 - P}$$

  - The odds ratio tells you how a unit increase or decrease in the corresponding input variable affects the odds of passing the test
    - When the odds ratio is greater than 1, it describes a positive relationship
    - An odds ratio less than 1 implies a negative relationship



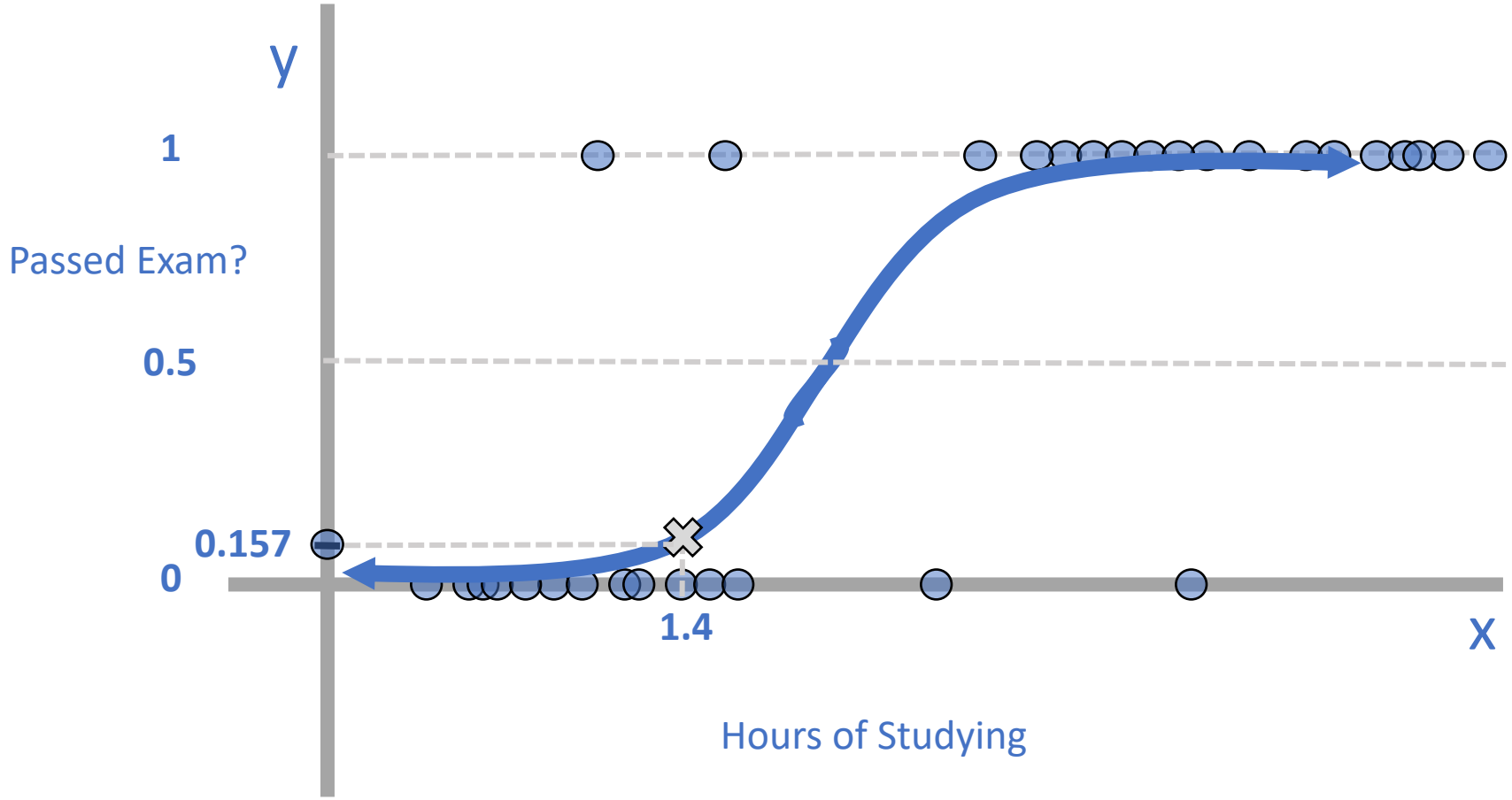
The odds, $O = p/(1-p)$, as a function of p

Note that $0 \le O$, and O is undefined for $p = 1$. Solving $O = \frac{p}{1-p}$ for p, $p = \frac{O}{(O+1)}$



p as a function of the odds, O

Visualizing the Logistic Regression Model



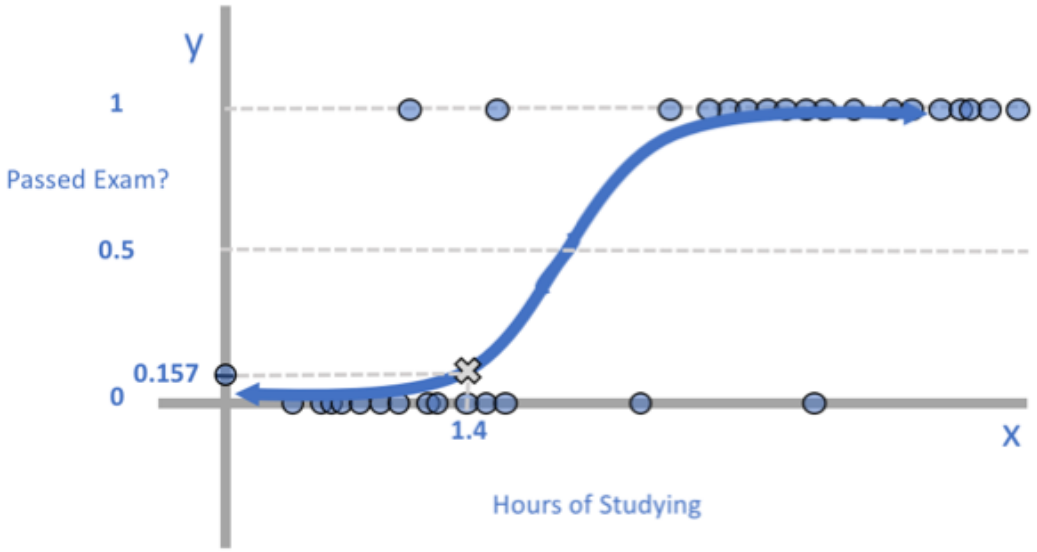
Logistic Regression Classification

- If a student only prepares 1.4 hours for the test
  - $study\_hours$=1.4
  - $\beta_0 = -2.54$
  - $\beta_1 = 0.614$

$$p(pass) = \frac{\left(e^{\beta_0+\beta_1*study_{hours}}\right)}{\left(e^{\beta_0+\beta_1*study_{hours}}\right)+1}$$

$$p(pass) = \frac{\left(e^{-2.54+0.614*1.4}\right)}{\left(e^{-2.54+0.614*1.4}\right)+1} = 0.157$$

- Our model predicts a probability of passing of 15.7%
- This falls below the 0.5 threshold and results in a prediction of failing the exam



Logistic Regression Example

Logistic Regression Summary

Exercise Model Comparison

Day 2 Recap

References

1. *Locally Weighted Learning* by Atkeson, Moore, Schaal
2. *Tuning Locally Weighted Learning* by Schaal, Atkeson, Moore
3. Kruschke, J. K. (2014). Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.
4. Machine Learning, Neural and Statistical Classification, Editors: D. Michie, D.J. Spiegelhalter, C.C. Taylor