KNN and Probability

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
  - k-NN is a supervised method for classification

Classification

- A few classification examples:
  - 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?
  - 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.
  - 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 k-NN
  - k-Nearest Neighbors
  - Naïve Bayes
  - Logistic Regression
  - Tree – based methods

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 85% 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.

Classification

A few issues to keep in mind when building a classifier

- **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.

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)
  - Uses 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 Brian’s weight
  - Who are the similar people?
  - Similar age, diet, height, waistline, activity level …
- Predict Brian’s IQ
  - Similar occupation, writing style, undergraduate degree, SAT score, …
- How do we calculate variously ranges in similarity?
  - Need some metric…
    - Distance

k-NN Approach

- Define a distance $d(x_1, x_2)$ between any 2 examples
  - Examples are essentially rows
  - 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

k-NN Approach

- Euclidean Distance Equation:

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

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

- p & z are the current data and q is new, think of p & z as the training and q as the test.

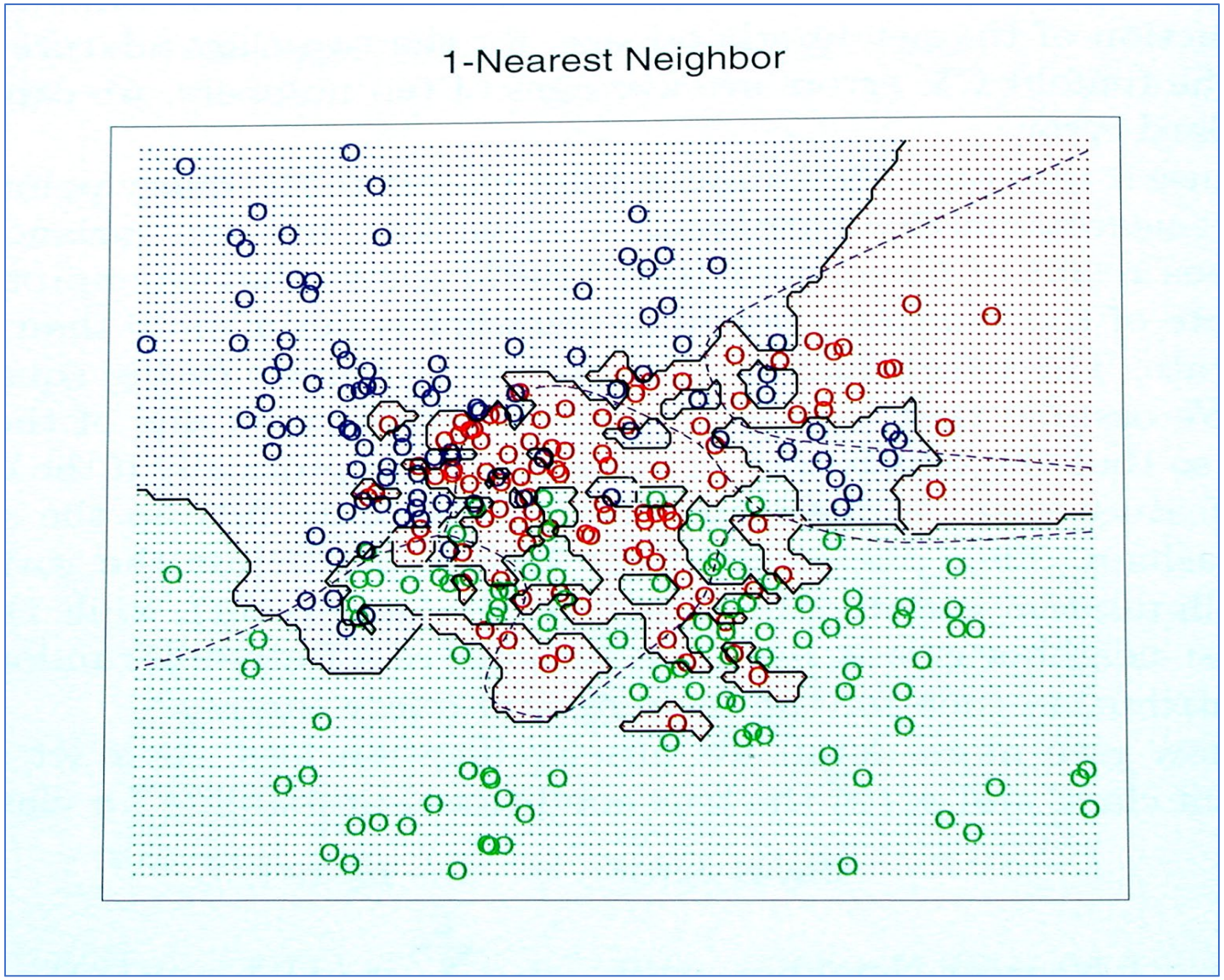
Subtracting then squaring row p with q. Then summing and taking the sqrt.

Do the same with z and compare

	x1	x2	x3	x4	x5	x6	x7	x8	Result
q	3	2	2	5	6	2	1	5	
p	4	3	3	6	4	4	5	6	
Euclidean	1	1	1	1	4	4	16	1	5.39
z	3	2	2	5	6	2	1	5	
Euclidean	0	0	0	0	0	0	0	0	0

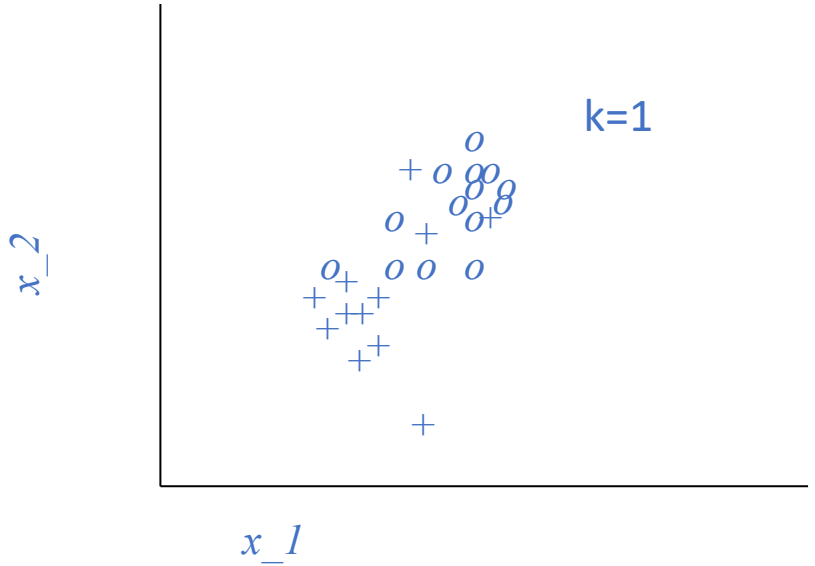
kNN Decision Boundaries

- kNN can learn complex decision boundaries



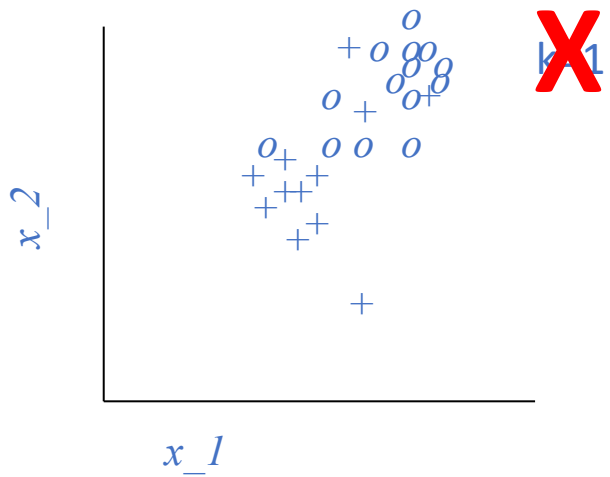
kNN

- 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



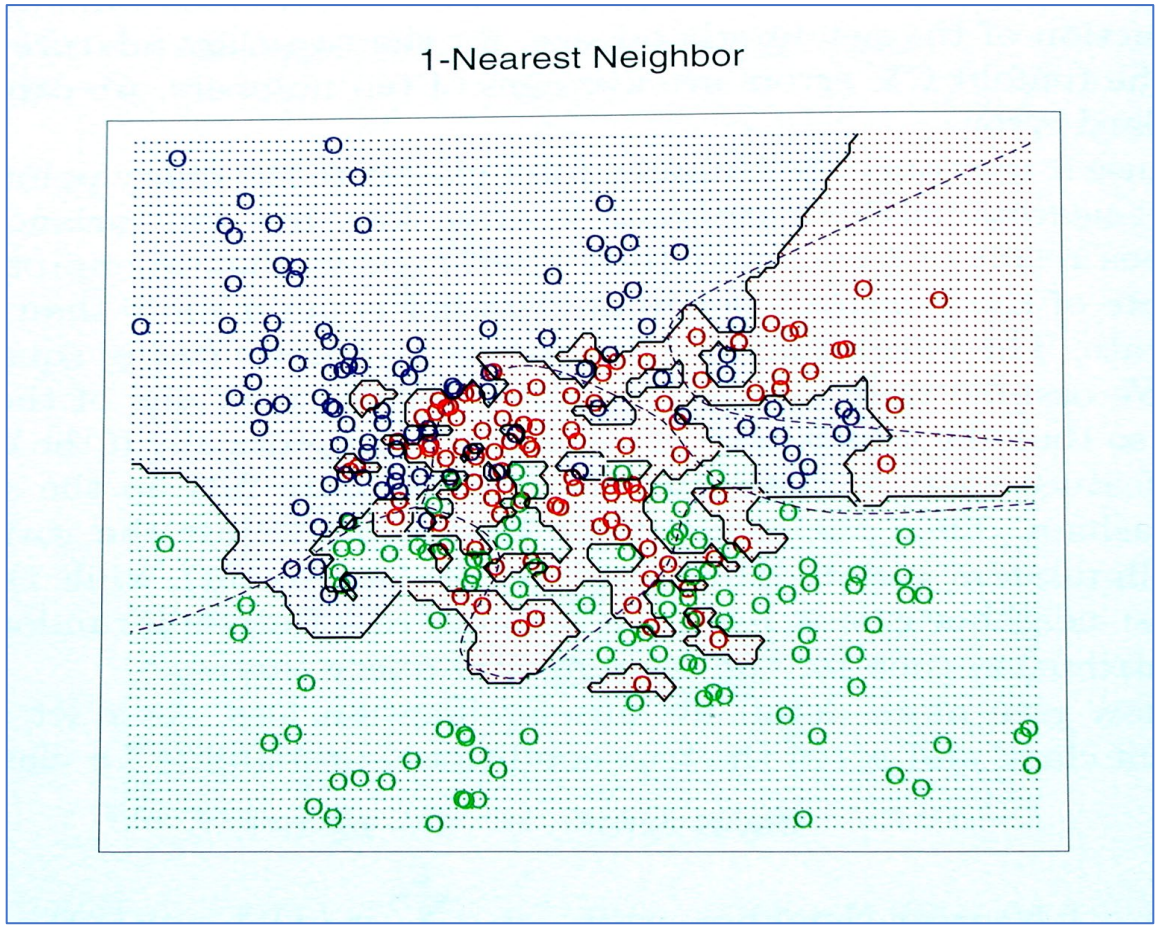
kNN

- 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?



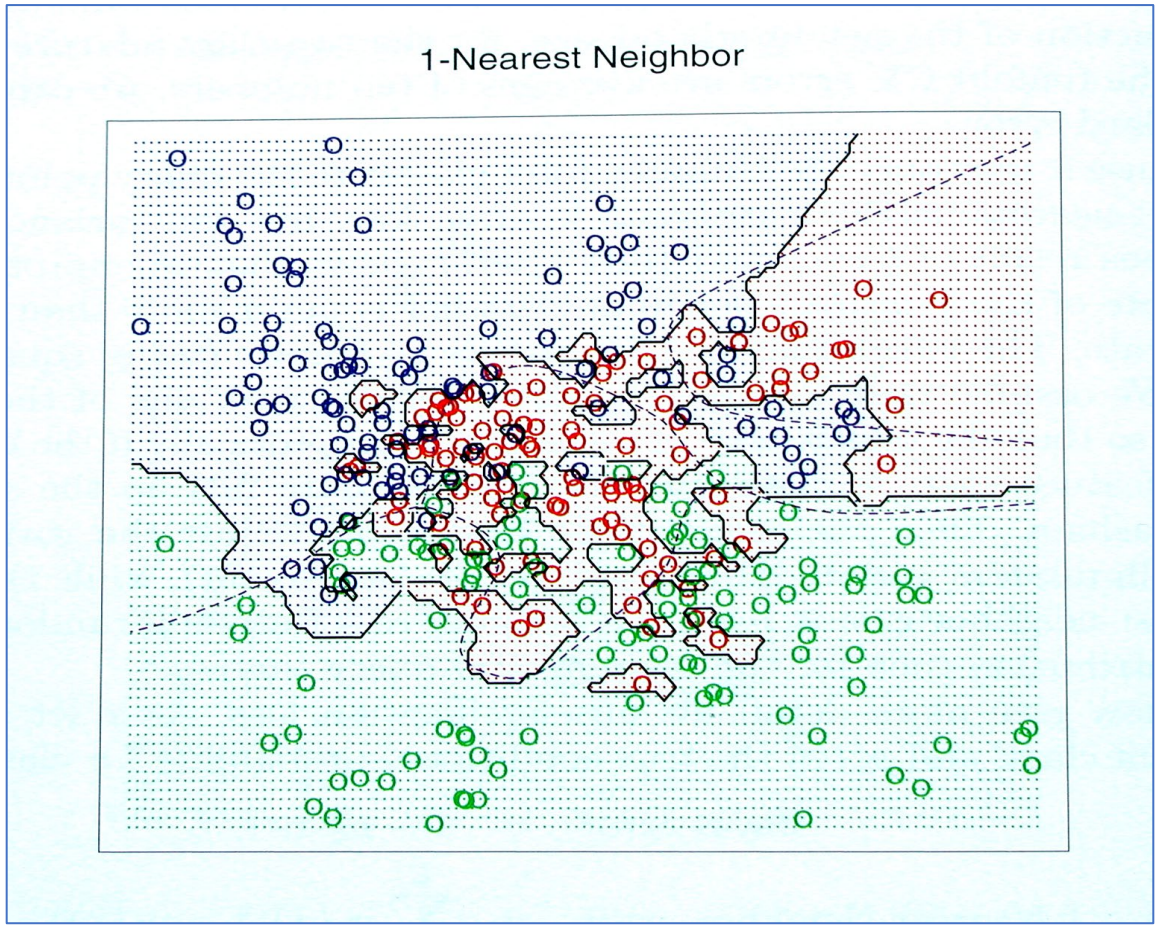
kNN Decision Boundaries

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



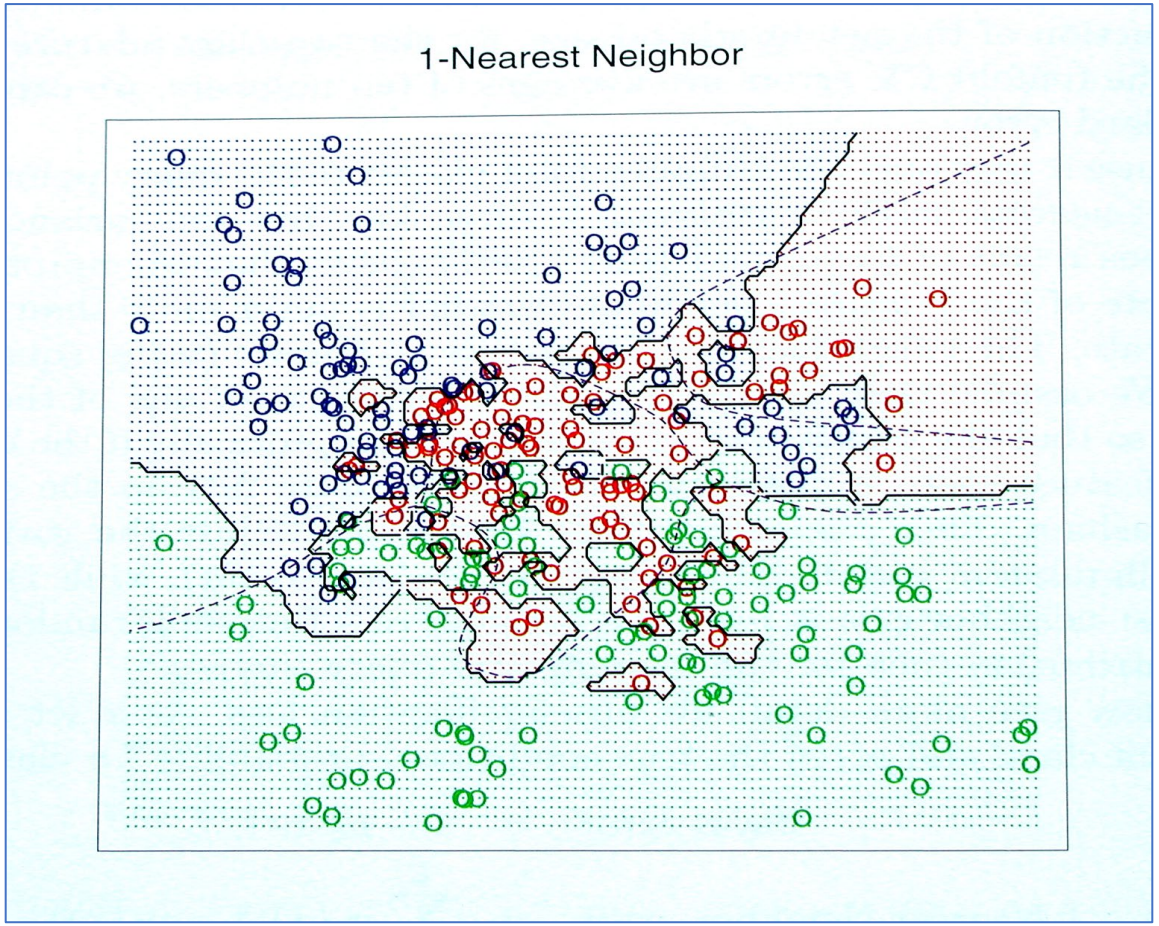
kNN Decision Boundaries

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



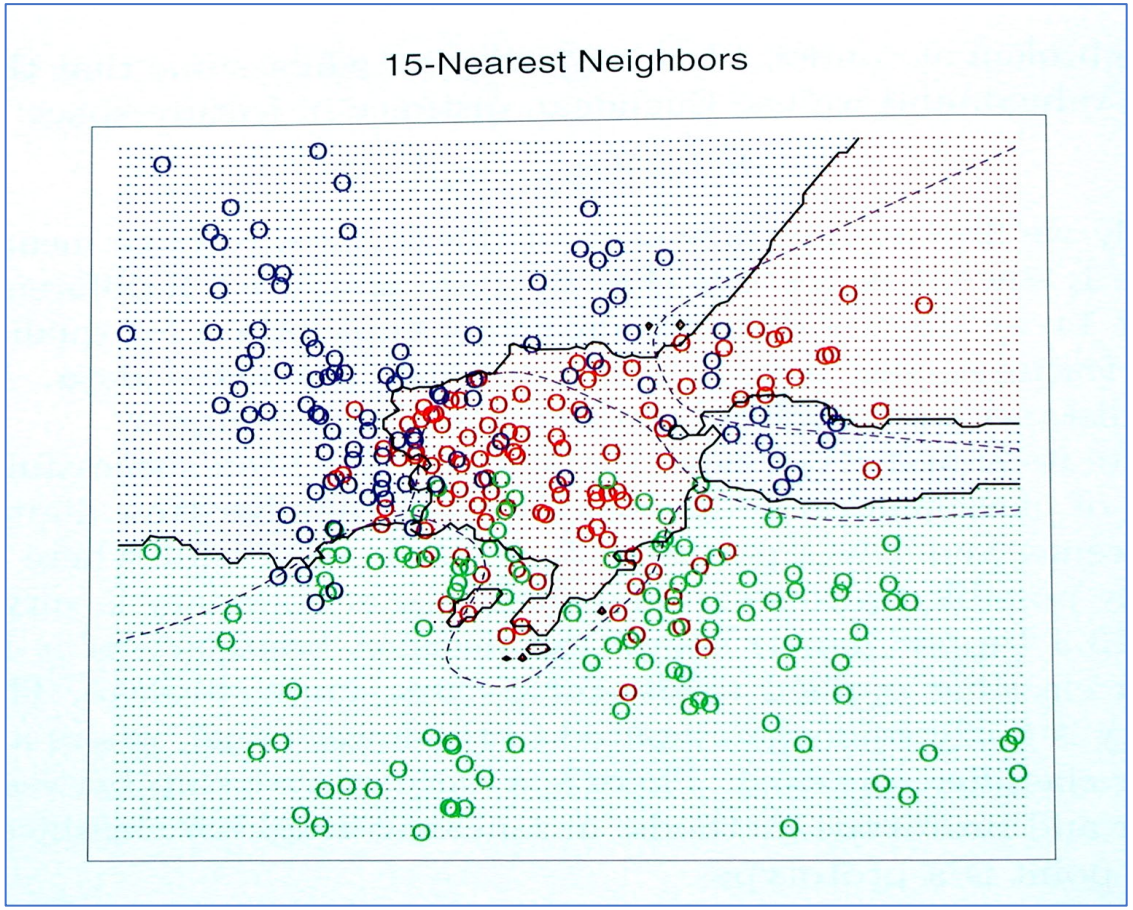
kNN Decision Boundaries

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



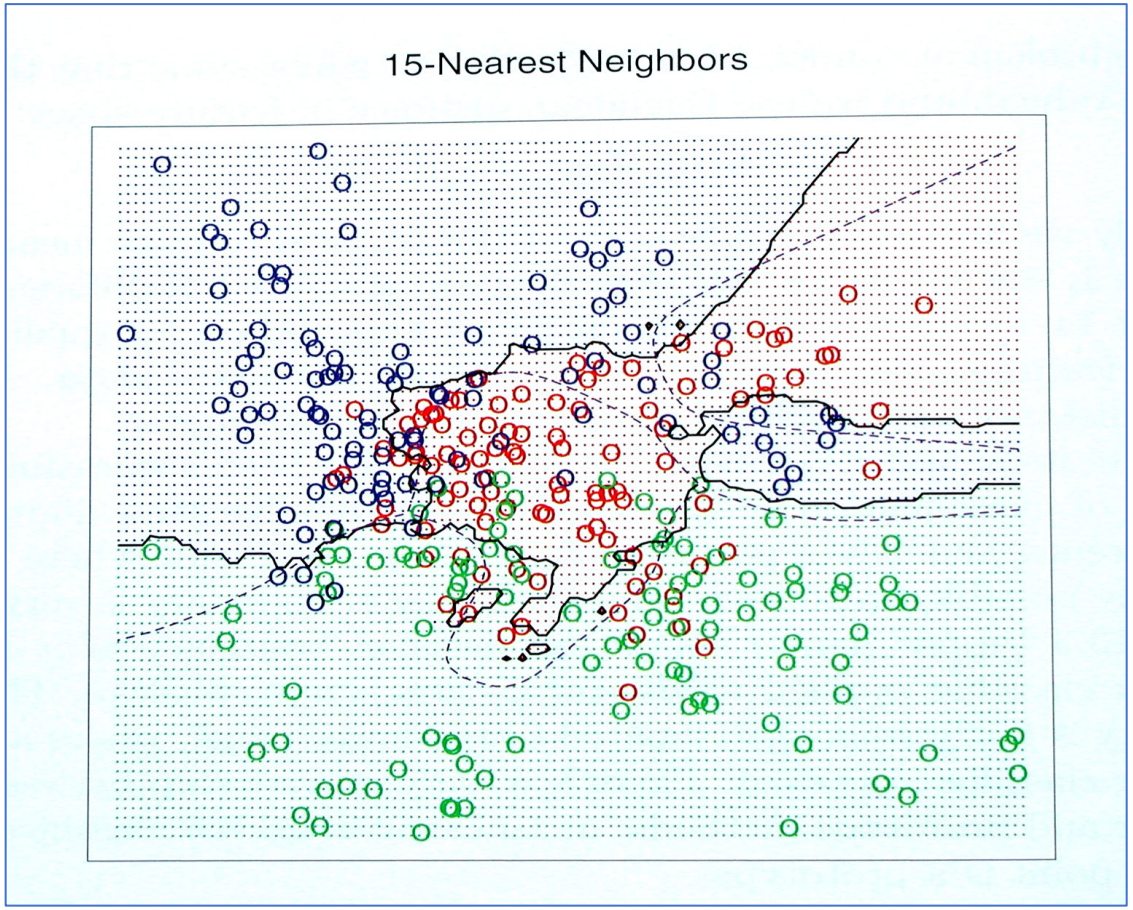
kNN Decision Boundaries

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



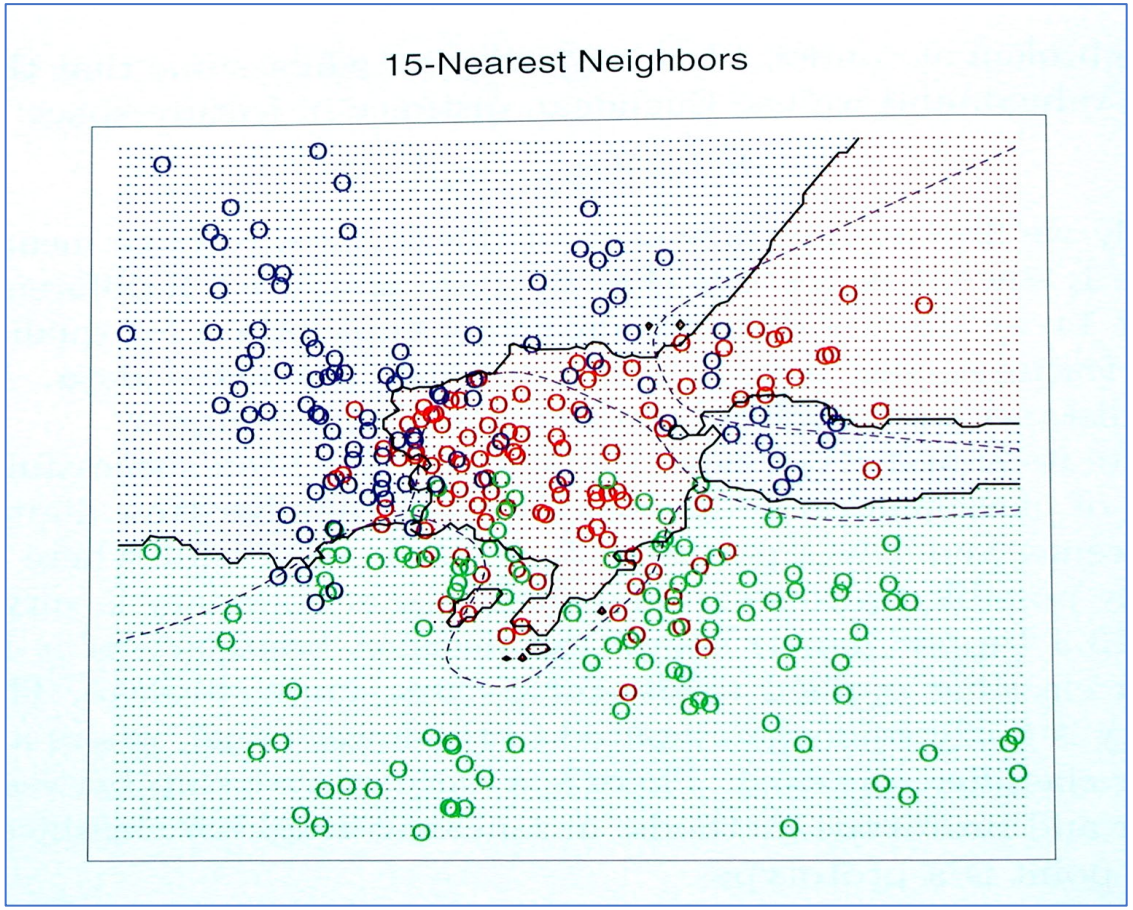
kNN Decision Boundaries

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



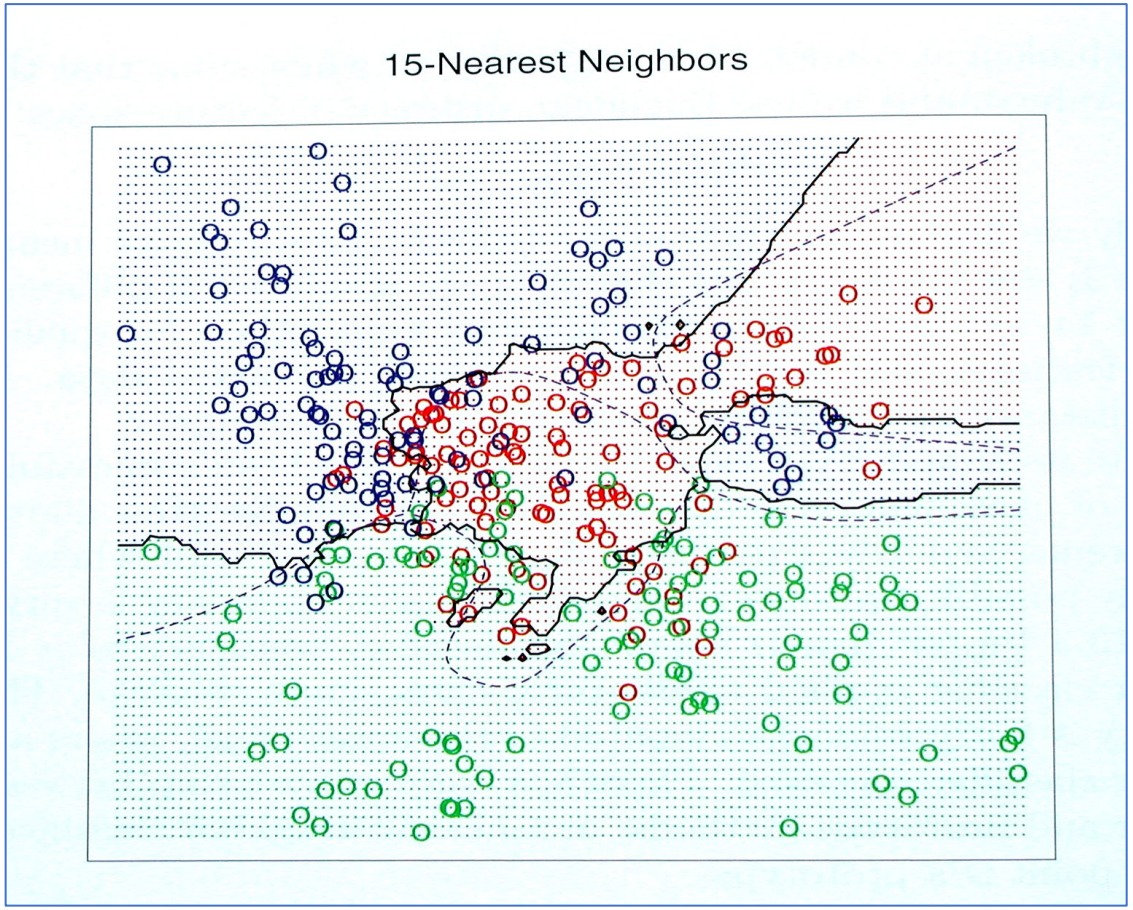
kNN Decision Boundaries

- 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



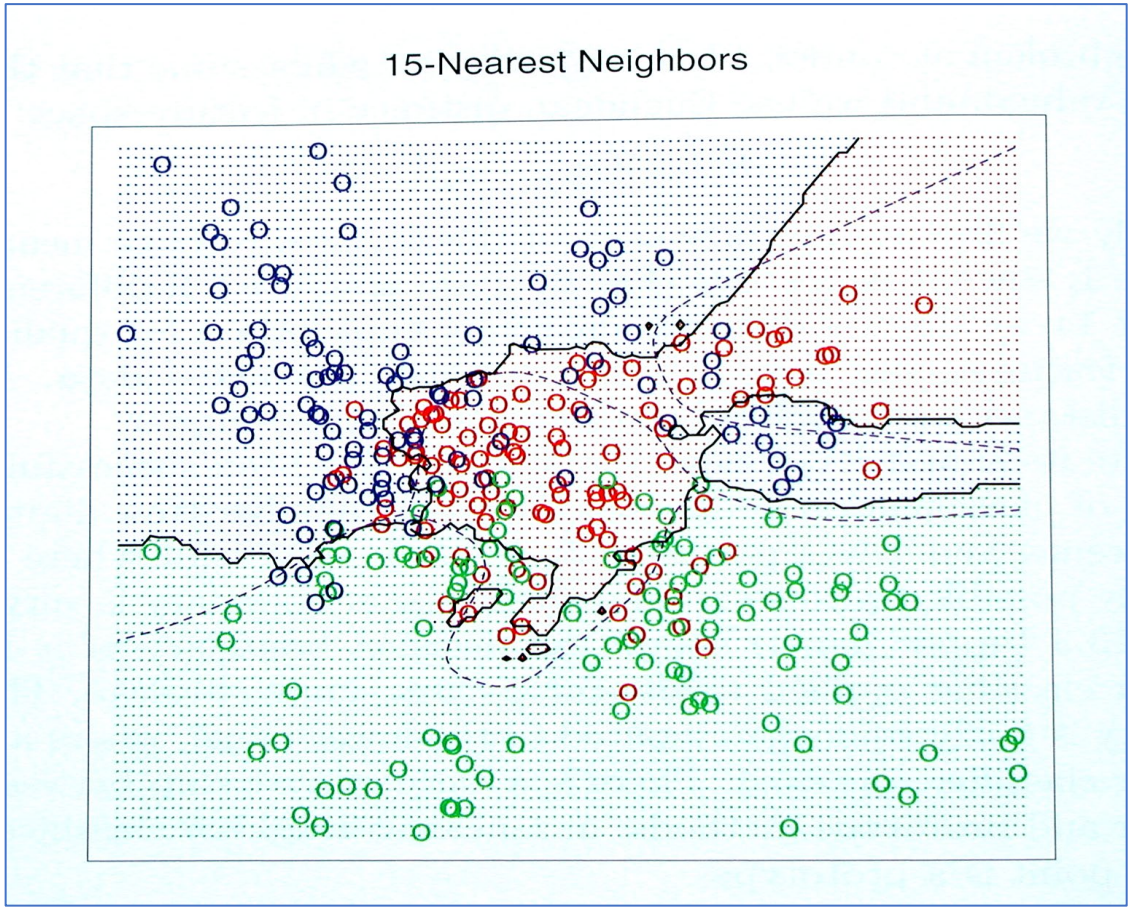
kNN Decision Boundaries

- 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?



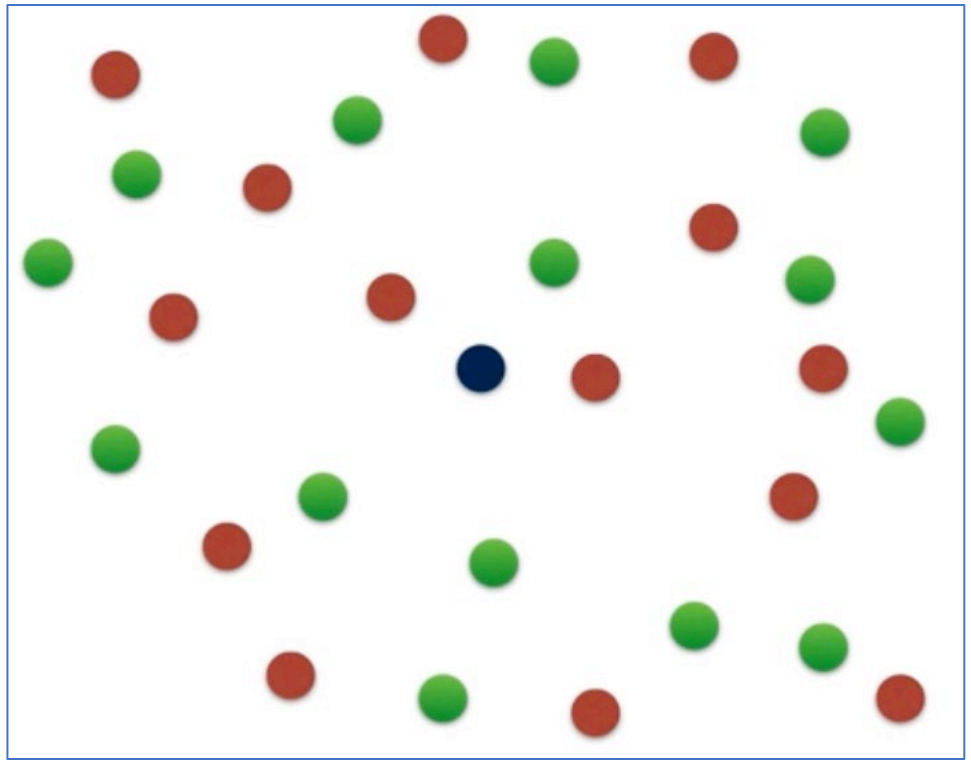
kNN Decision Boundaries

- 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



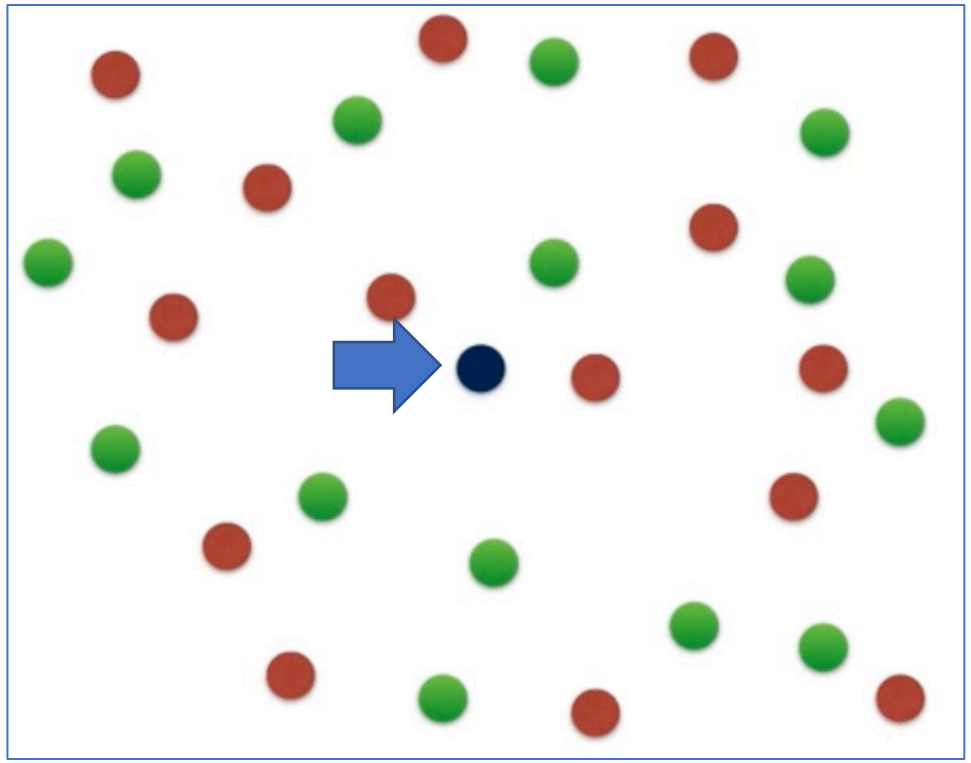
k-NN Graphical Example

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



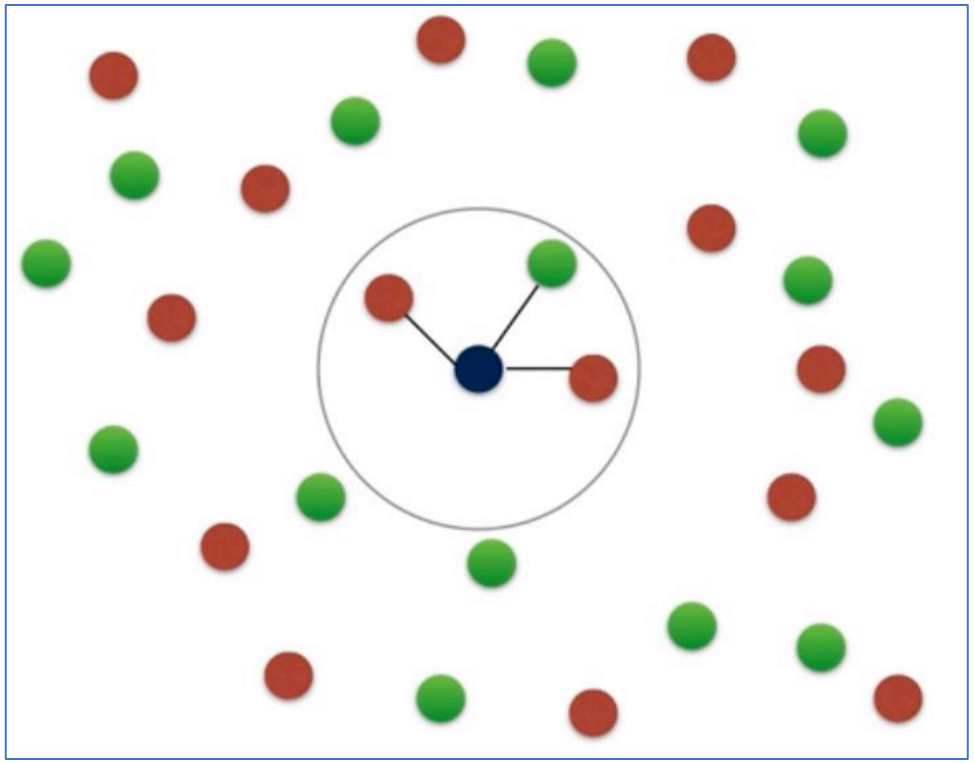
k-NN Graphical Example

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



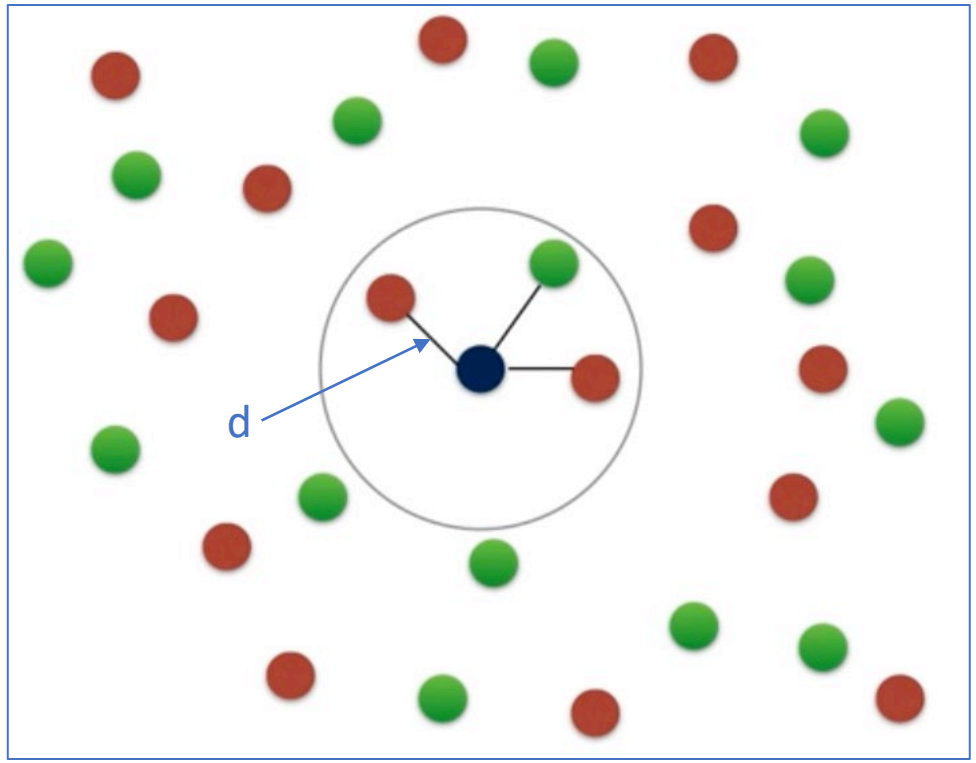
k-NN Graphical Example

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



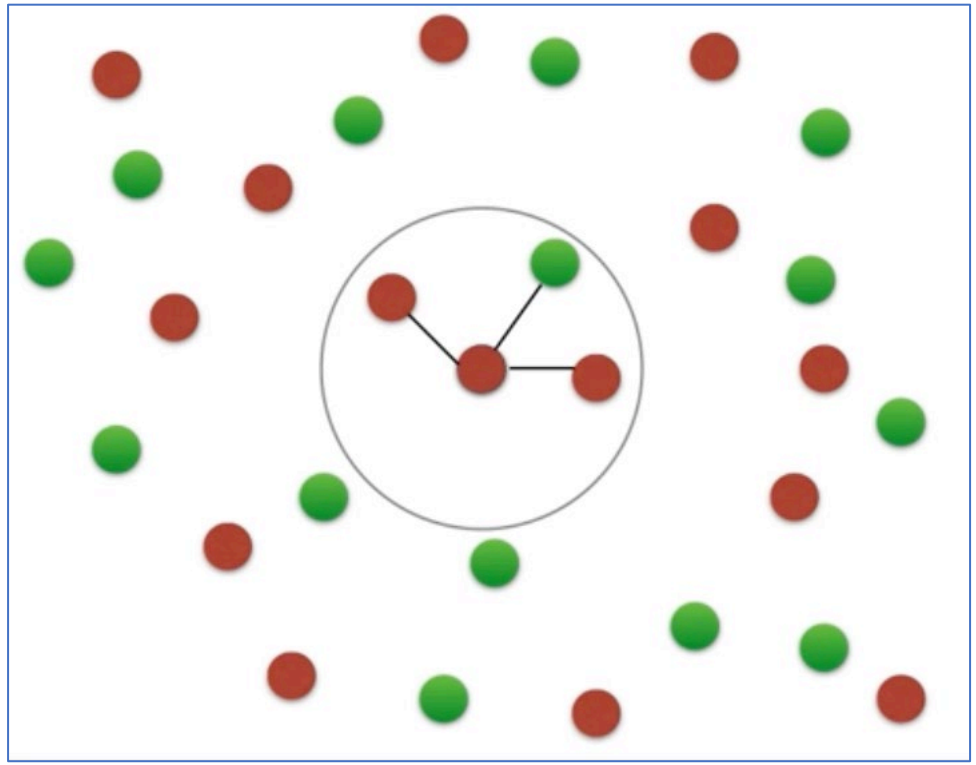
k-NN Graphical Example

- 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



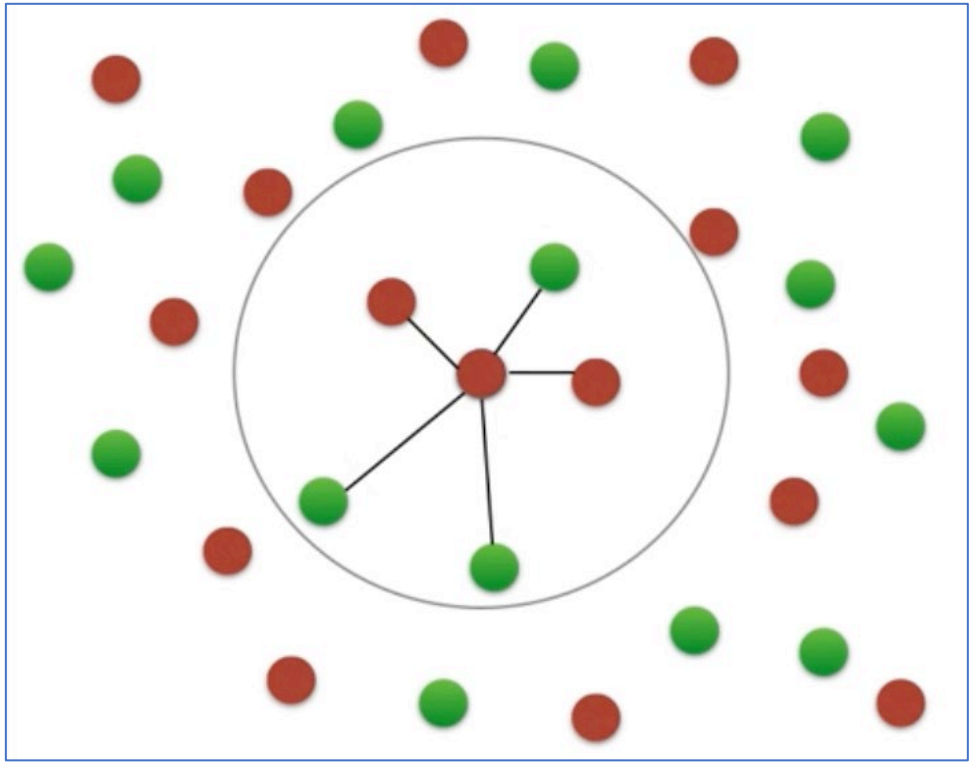
k-NN Graphical Example

- 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



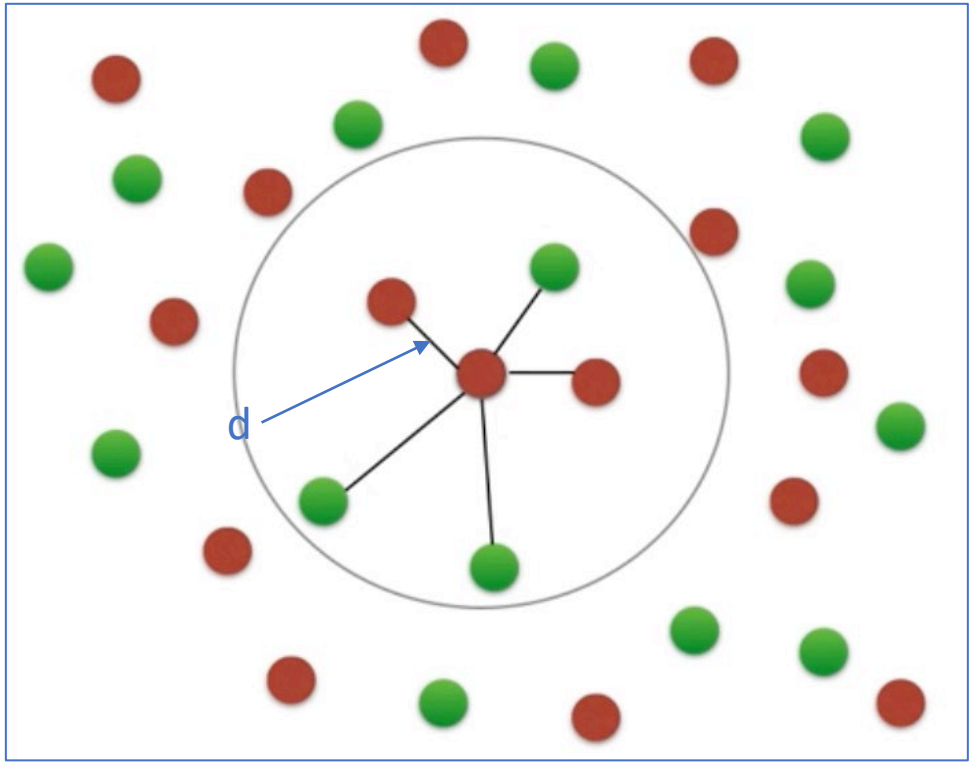
k-NN Graphical Example

- 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



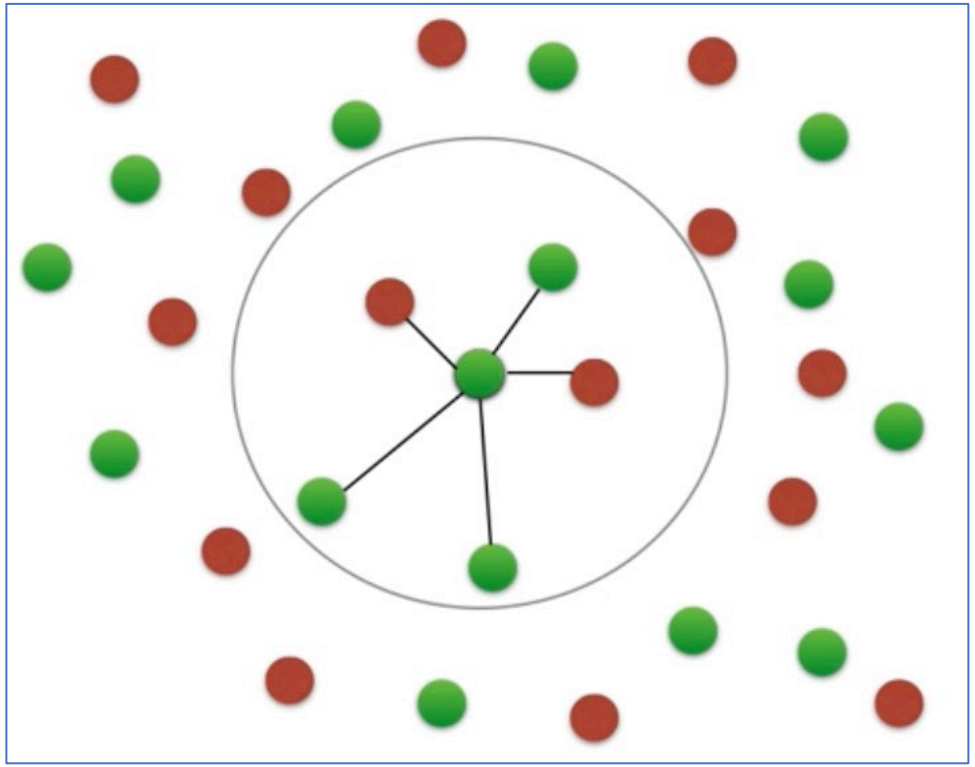
k-NN Graphical Example

- 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



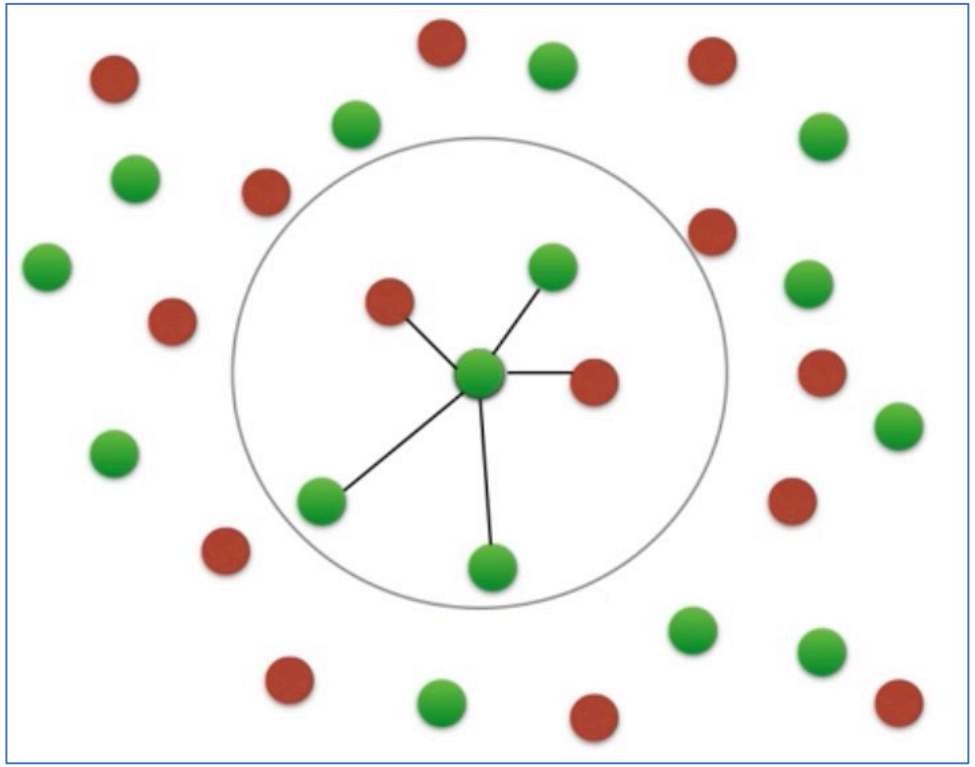
k-NN Graphical Example

- 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?

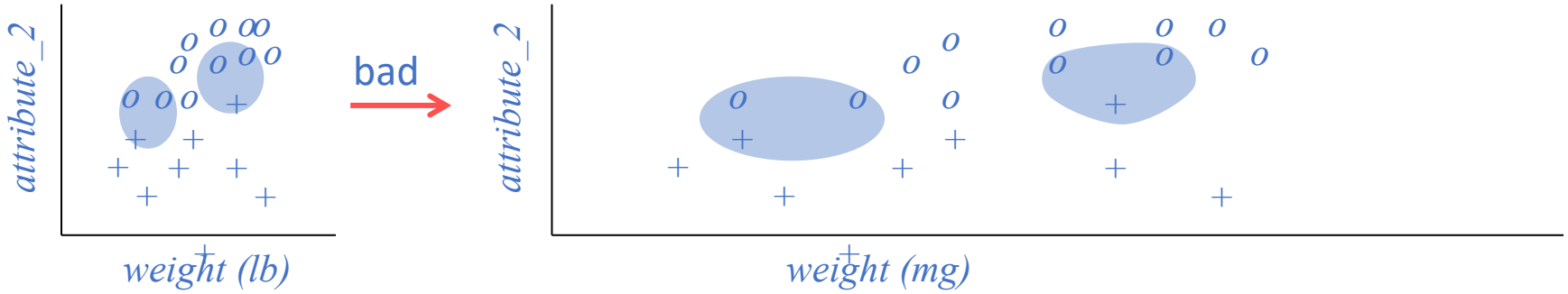


How to choose “k”

- 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
- Balance between large and small k

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
    - Rescale weights so that standard deviation = 1



More kNN Details

- 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 - method that generalizes data in the testing (deployment) phase, rather than during the training phase – designed to be continuously updating as new data comes in
  - A benefit of lazy learning is that it can quickly adapt to changes,
    - Think Netflix recommendations, new options are appearing constantly so have a static training set it's really valuable.
  - 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 (starts to normalize to 1)
- 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

Switch to R

Understanding Probability Classifiers

- To be able to unpack the probability classifiers, 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

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

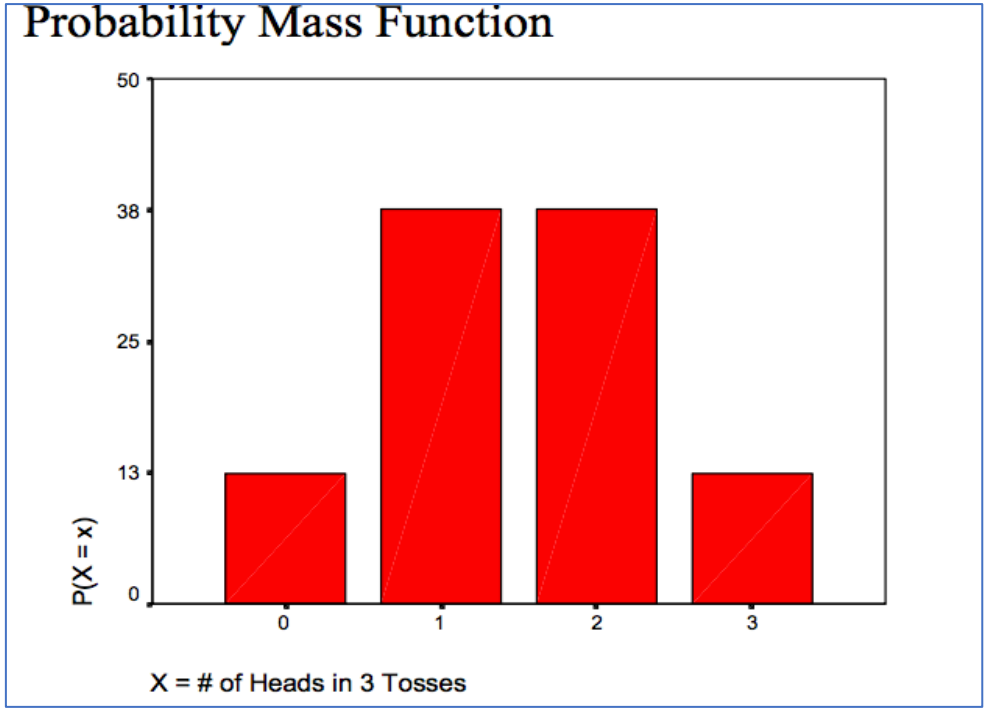
- Example
  - Therefore, the probability distribution for the number of heads occurring in three coin tosses is

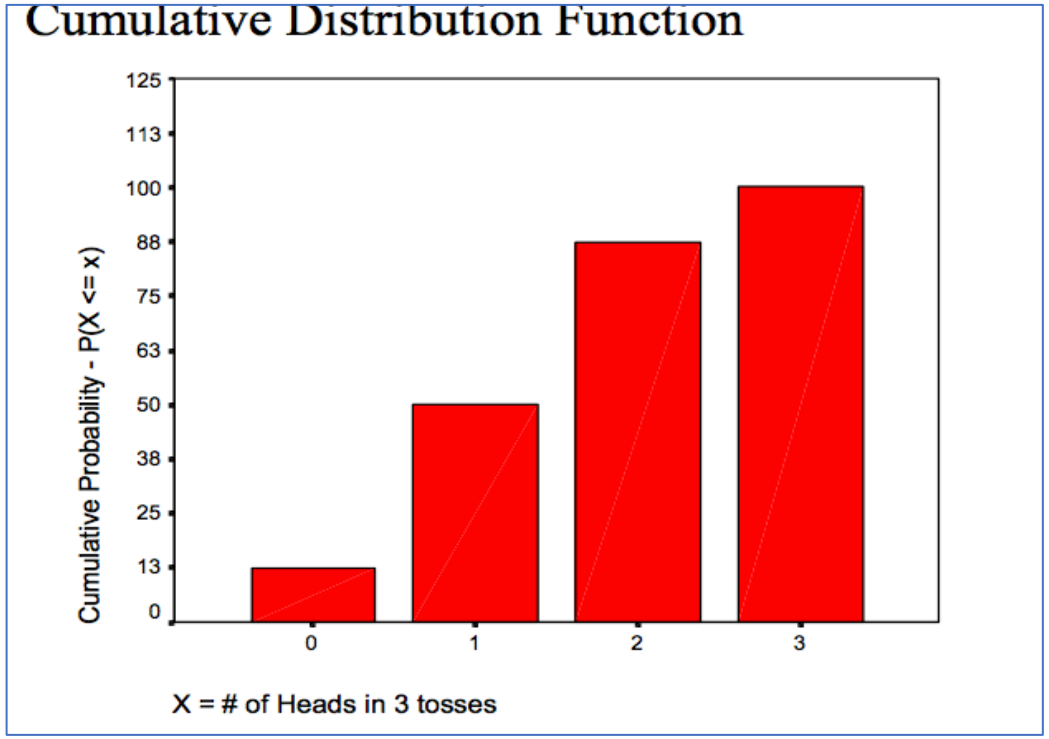
Count of Heads (X)	p(x)	F(x)
0	1/8	1/8
1	3/8	4/8
2	3/8	7/8
3	1/8	1

Discrete Probability Example

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

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





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

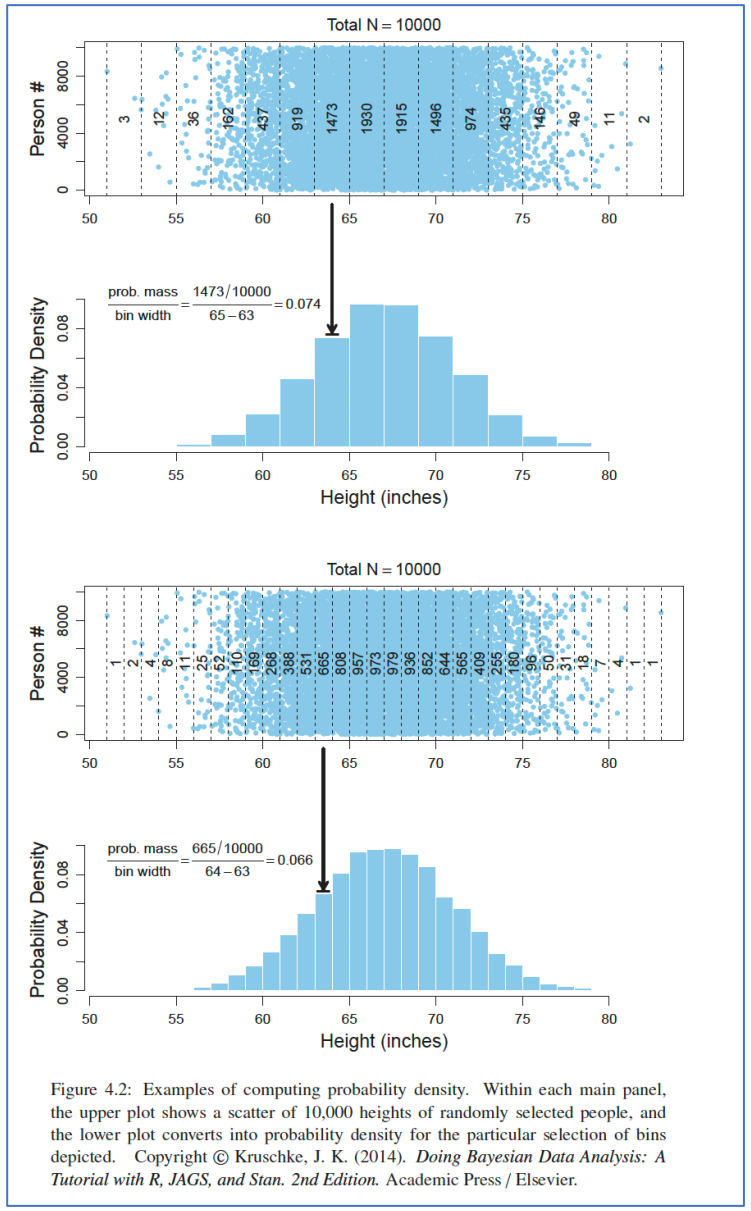


Figure 4.2: Examples of computing probability density. Within each main panel, the upper plot shows a scatter of 10,000 heights of randomly selected people, and the lower plot converts into probability density for the particular selection of bins depicted. Copyright © Kruschke, J. K. (2014). *Doing Bayesian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

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

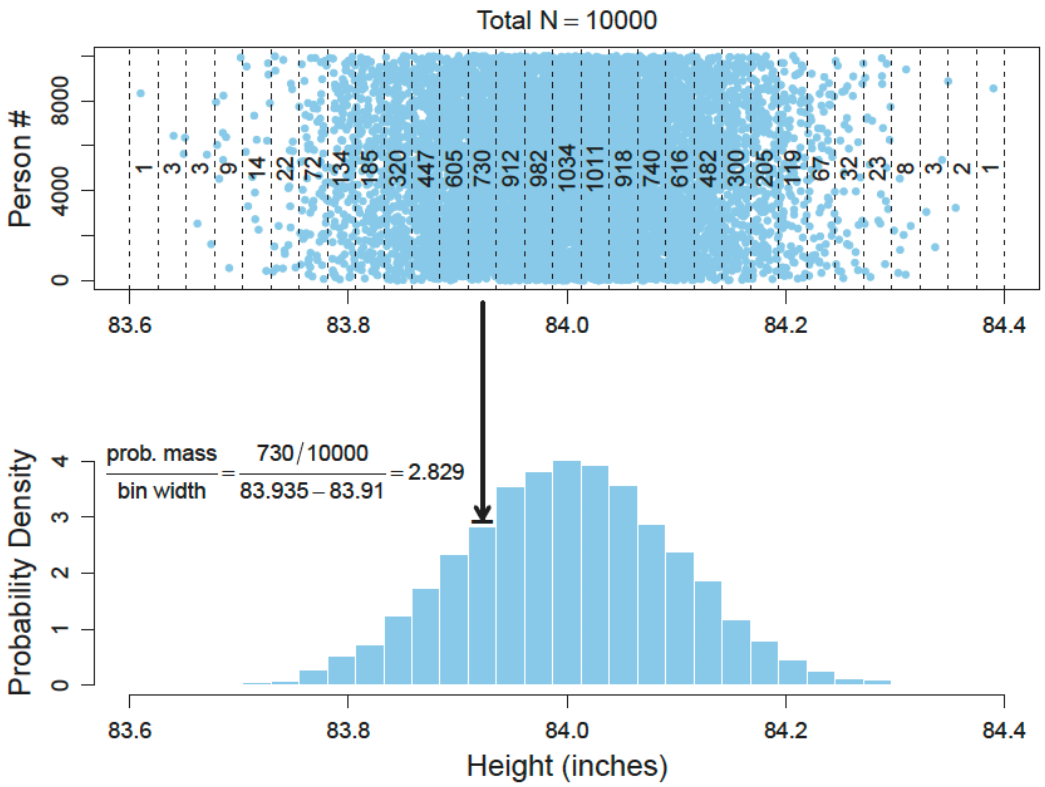


Figure 4.3: Example of probability density greater than 1.0. Here, all the probability mass is concentrated into a small region of the scale, and therefore the density can be high at some values of the scale. The annotated calculation of density uses rounded interval limits for display. (For this example, 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.

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

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

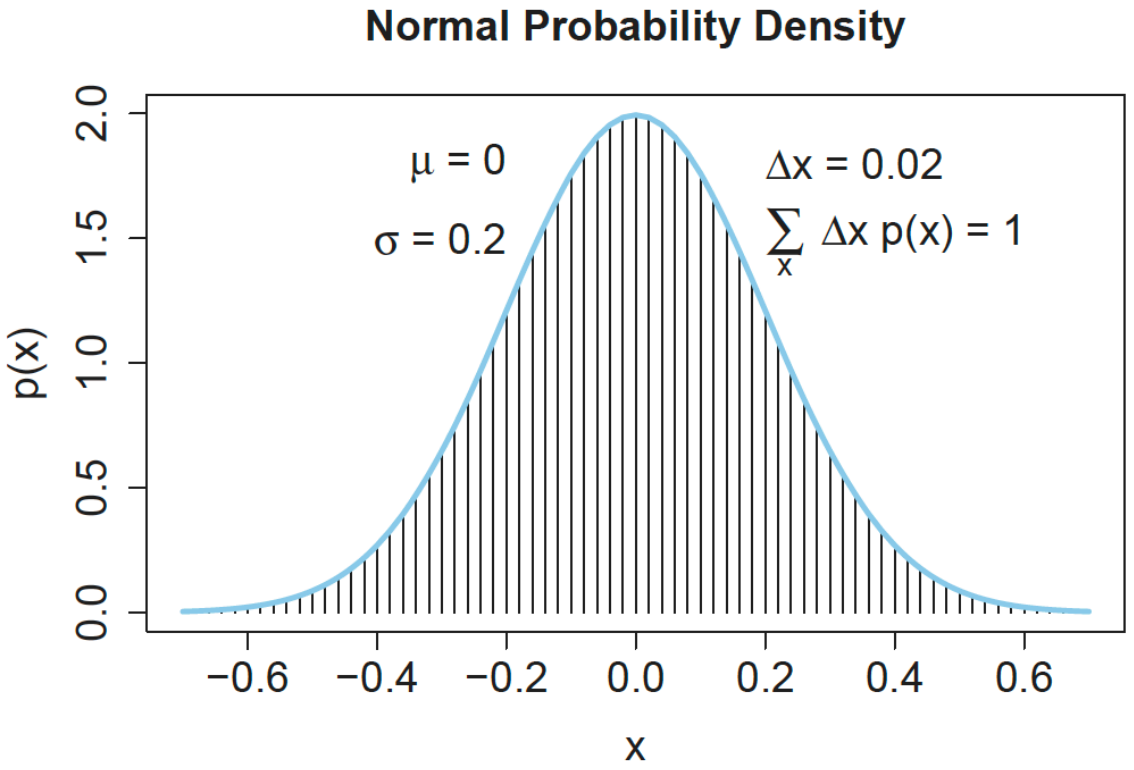
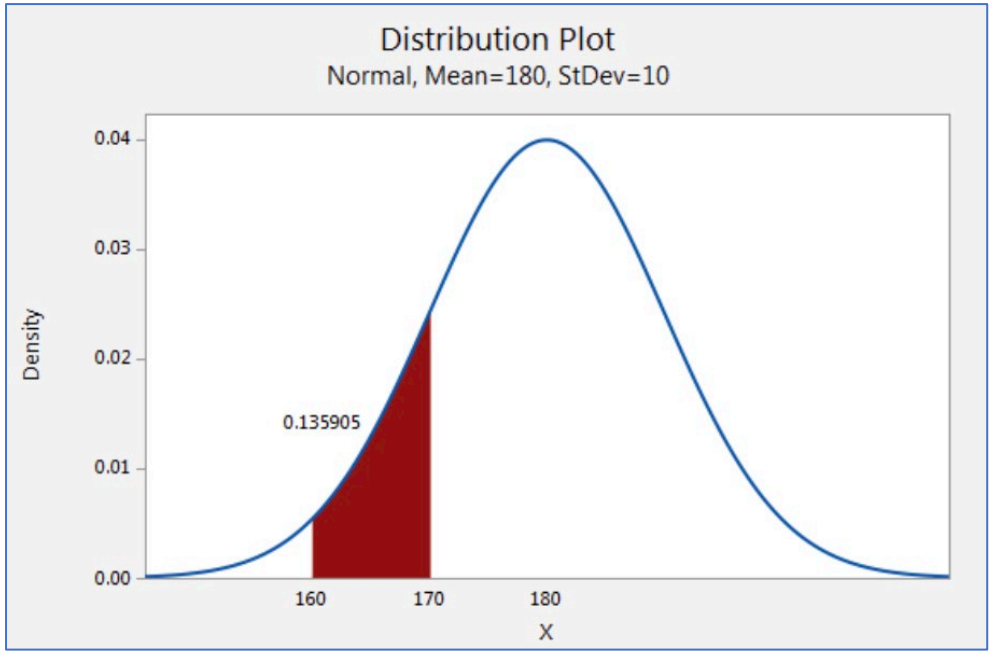


Figure 4.4: A normal probability density function, shown with a comb of narrow intervals. The integral is approximated by summing the width times height of each interval. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press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

Example - Continuous Normal Distribution

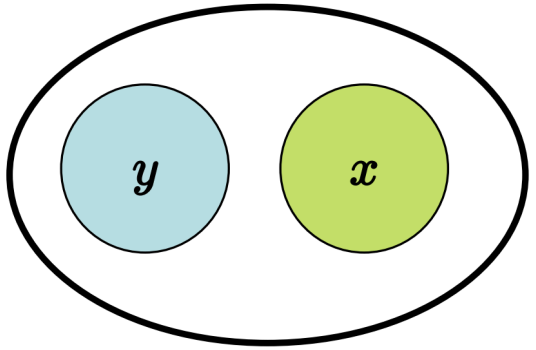
- Example of the continuous distribution of weights
  - The continuous normal distribution can describe the distribution of weight of adult males.
  - For example, you can calculate the probability that a man weighs between 160 and 170 pounds.
  - The area of this range is 0.136; therefore, the probability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

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

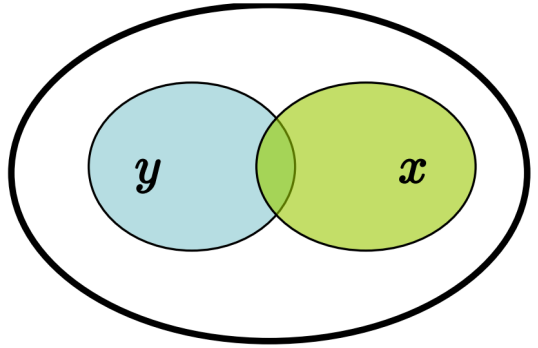


Joint Probability

- 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.

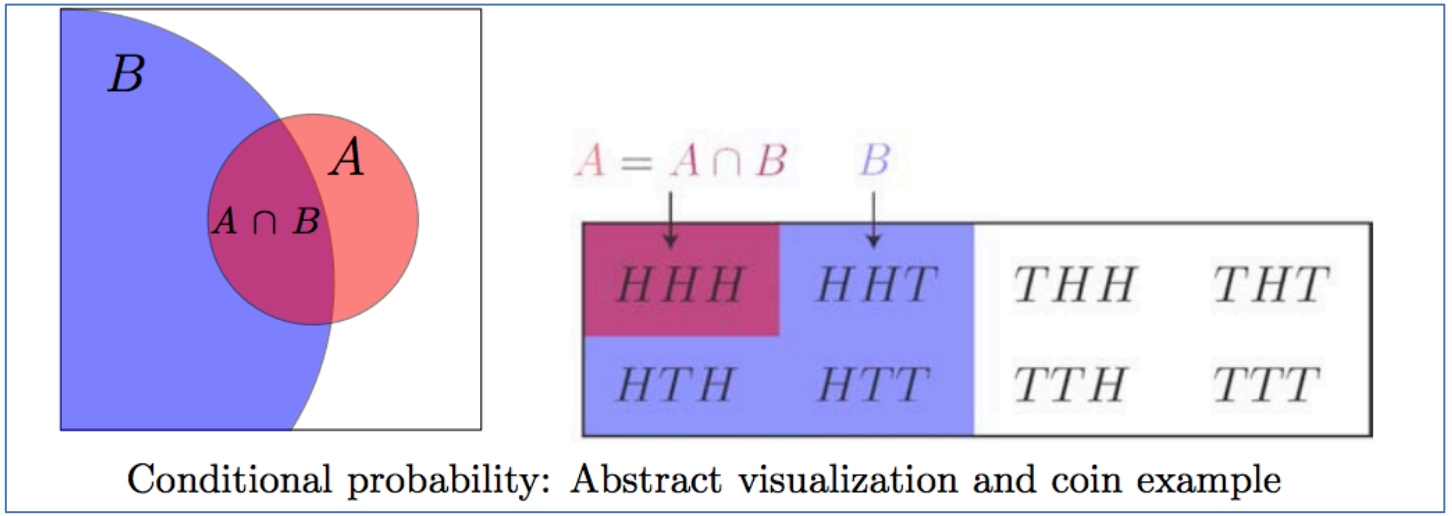
Eye Color	Hair Color				Marginal
	Black	Brunette	Red	Blond	(Eye Color)
Blue	.03/.36 = .08	.14/.36 = .39	.03/.36 = .08	.16/.36 = .45	.36/.36 = 1.0

Conditional Probability 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":

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

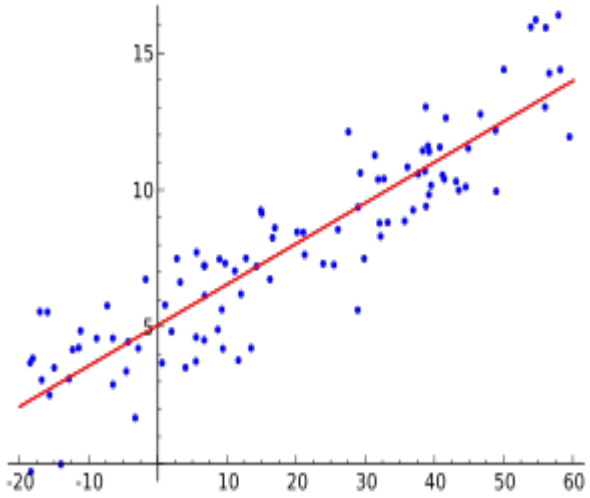
  - Probability of rolling a dice and getting "3":

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

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

Regression Review

Linear regression finds the straight line, called the **least squares regression line** or LSRL, that best represents observations in a data set. Suppose *Y* is a dependent variable, and *X* is an independent variable. Then, the equation for the regression line would be: $\hat{y} = b_0 + b_1 x$



Linear regression equation

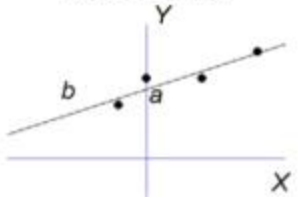
(without error)

$$\hat{Y} = bX + a$$

predicted values of *Y*

slope = rate of increase/decrease of Y hat for each unit increase in *X*

Y-intercept = level of *Y* when X is *0*.



Assumptions

- Linear relationship between dependent and independent - Plot
- Multicollinearity – occurs when independent variables are not independent – checked with VIF
- Auto-correlation – Occurs when residuals are not independent from each other – stock prices example- Durbin-Watson d tests
- Homoscedasticity – Error term along the regression line are equal – Scatter Plot – can convert the dependent variable.

Regression

- Works best with numeric/continuous data to increase the inferential power
- We are trying to model or explain the relationship between a single variable *Y (response, output, dependent)* and one or more X_1….. *(predictor, input, independent, explanatory, regessors..etc.)*
- *Y* must be a continuous variable but X categorical, continuous,
- Centering (scale/Zscores) variables so that the predictors have mean 0, is often recommended. The intercept term is then interpreted as **the expected value of** Yi **when the predictor values are set to their means**. Otherwise, the intercept is interpreted as the expected value of Yi when the predictors are set to 0, which may not be a realistic or interpretable situation (e.g. what if the predictors were height and weight?).

What Does r^2 (Coefficient of Determination) Mean?

This statistic quantifies the proportion of the variance of one variable “explained” (in a statistical sense, not a causal sense) by the other.

As an example at data set in R (.2697) = 27% of the variability in trunk Height is explained by Girth

```
Call:
lm(formula = Height ~ Girth, data = trees)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5816  -2.7686   0.3163   2.4728   9.9456

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  62.0313     4.3833  14.152 1.49e-14 ***
Girth         1.0544     0.3222   3.272  0.00276 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.538 on 29 degrees of freedom
Multiple R-squared:  0.2697,    Adjusted R-squared:  0.2445
F-statistic: 10.71 on 1 and 29 DF,  p-value: 0.002758
```

R Regression Output: Coefficients

- Formula Call? Regression equation
- Estimate? Coefficients for the regression equation
- Standard Error?

  Measures the average amount that the coefficient estimates vary from the actual average value of our response variable, can be used to generate confidence intervals.

- P value?

  A small p-value indicates that it is unlikely we will observe a relationship between the predictor (Girth) and response (Height) variables due to chance, a p-value of .05 or less is considered statistical significant

R Regression Output

- Residual Standard Error

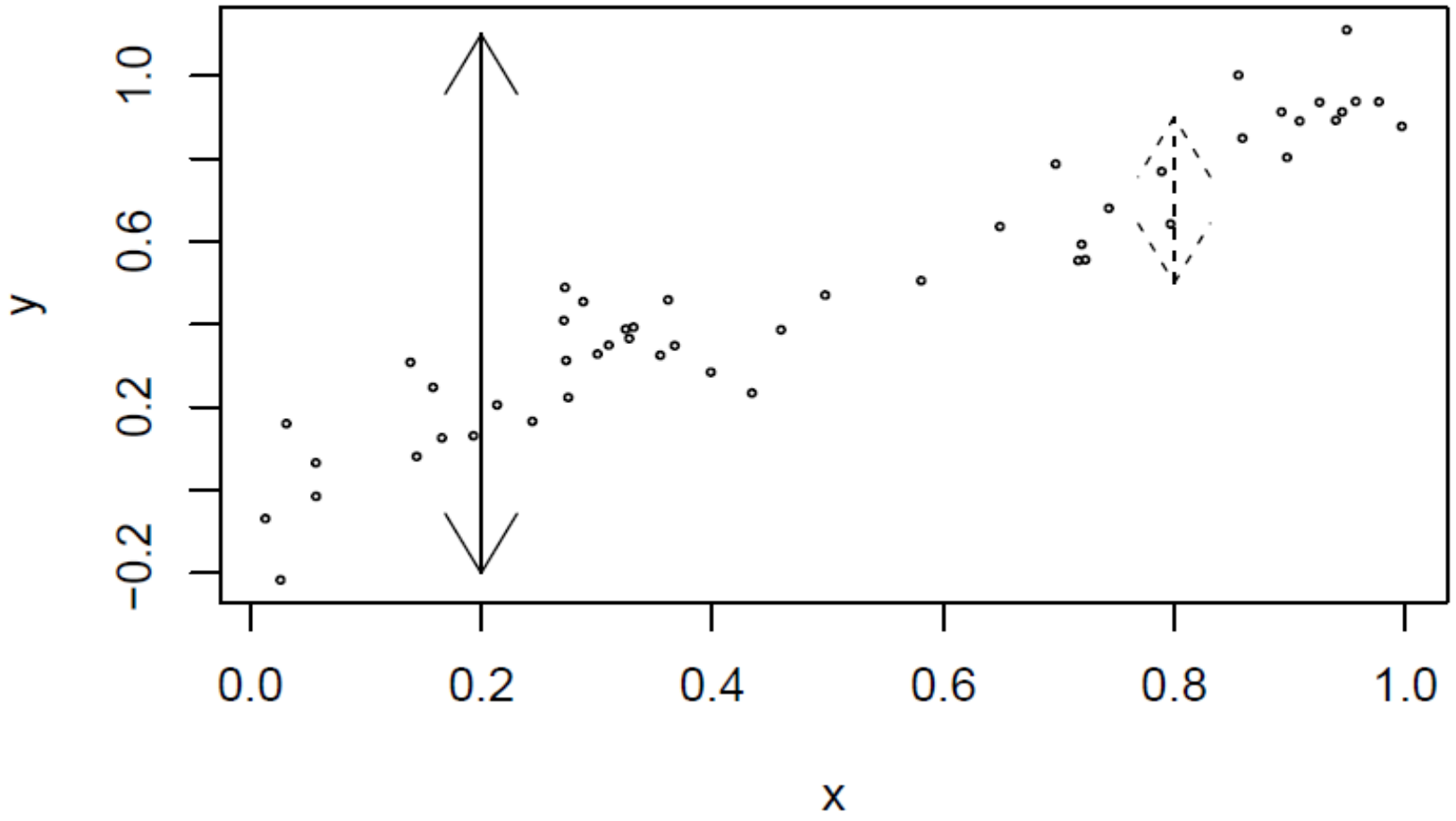
  The average amount that the response will deviate from the **true regression line.**
  Used to evaluate our model.
  In our example we have a residual error of 5.538 in the predication of tree height.
  The average tree height is 76 meaning we could be off by roughly 7%.

- Multiple R-squared?

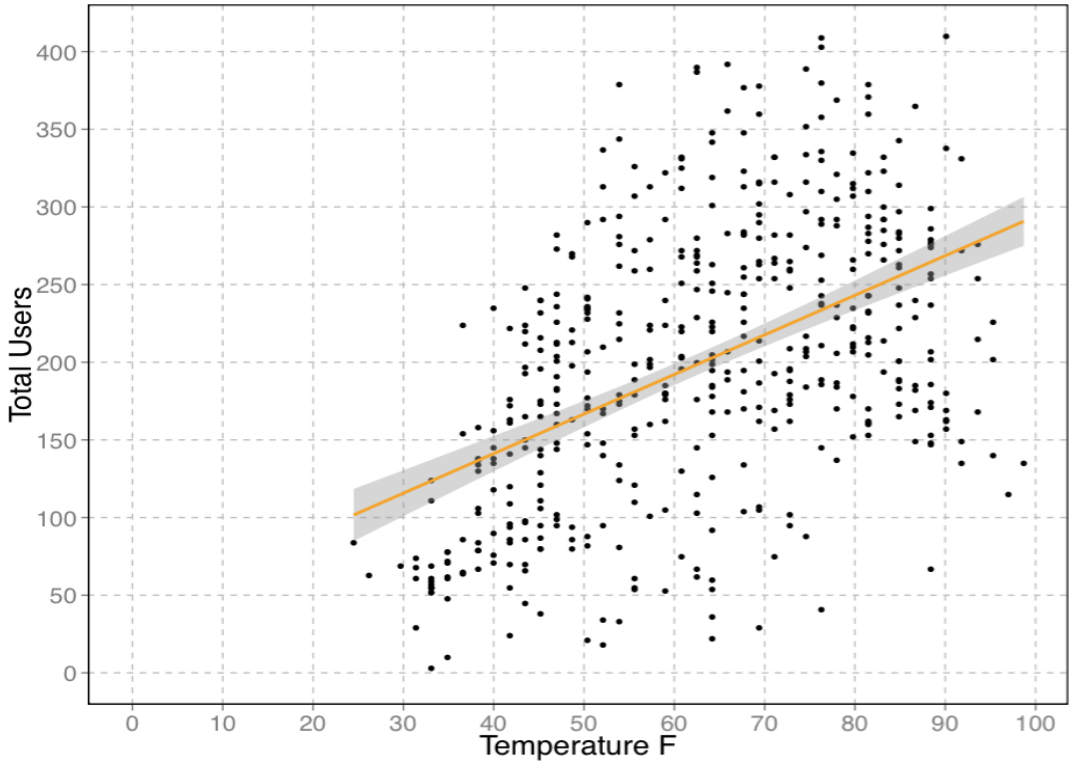
  Provides a measure of how well the model is fitting the actual data, percentage of variance explained by the predictors, ours example is 27%

Visual

GOODNESS OF FIT



Bike Share Output



Another Definition: Functional Approximation

- What is a functional approximation problem?
  - Target variable: Dependent: What we are trying to predict
  - Other Variables: Independent: Using to Predict
- Functional approximation is a approach that uses the other variables we have access to approximate the dependent and does so through the function development
- We will use regression and which assumes that we have a numeric target variable, for classification it's often a bi-variate or class level variable

Assessment Measures

- Assessing Regression Models: MSE, RMSE and MAE
  - MSE – The difference between the predicted values and the actual values squared
  - RSME – Same as above only the square root is taken to put the error back in terms of the dependent variable
    - Can also normalize the RSME to the range of the data in order to be able to compare RSME outputs that include different data ranges
  - MAE – The same approach only taking the absolute value instead of squaring

Equations

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - x|$$

$$NRMSE = \frac{RMSE}{X_{obs,\max} - X_{obs,\min}}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}$$

Linear Regression

- Best used in situations where the data being deploy lacks a high level of complexity
- Trains very fast but isn't able to create complex decision boundaries when data is heterogeneous.
- Also as compared to most ML approaches OLS does not have a built in function that can control for overfitting
  - However not all hope is lost, we can use forward stepwise regression – which we discussed in Intro to DS or
  - Ridge/Penalized Regression

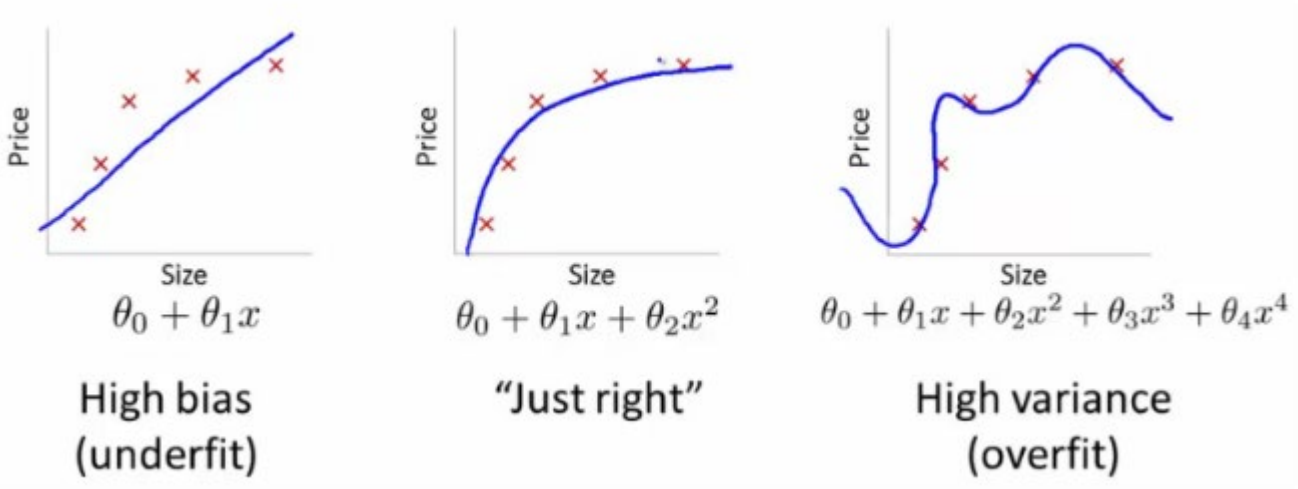
Regression

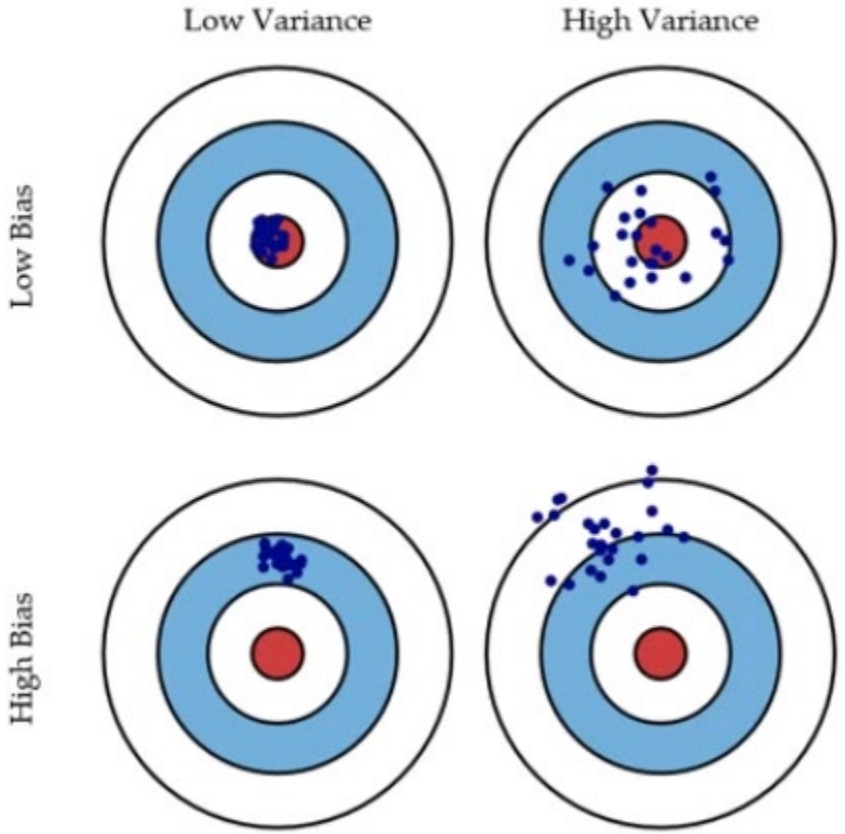
- Said another way basic linear regression has a Prediction Accuracy Problem:
  - Has a low bias (overfitting) but a high variance
  - This can be improved by injecting some level of bias into the equation by reducing the impact of certain coefficients
  - This can improve overall accuracy by reducing variance
- Draw Dart Board –
- Another issue is interpretation – with a large number of predictor variables and large data sets it often hard to identify variable importance and explain model outcomes

Regression and Sparsity

- Often we have more features than observations in the world of big data
  - What type of problem is this?
- So we strive to have Sparse models
  - What do we mean by Sparse?

Bias Versus Variance





Ridge Regression

- Injecting bias can be done through a regulator or utilization of a penalizing attribute. two common examples are Ridge and Lasso, let's start with Ridge

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i,$$ Basic Regression Equation

Ridge regression is like least squares but shrinks the estimated coefficients towards zero. Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$, the ridge regression coefficients are defined as

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Note that:

- When $\lambda = 0$, we get the linear regression estimate
- When $\lambda = \infty$, we get $\hat{\beta}^{\text{ridge}} = 0$
- For λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients

Ridge Penalty

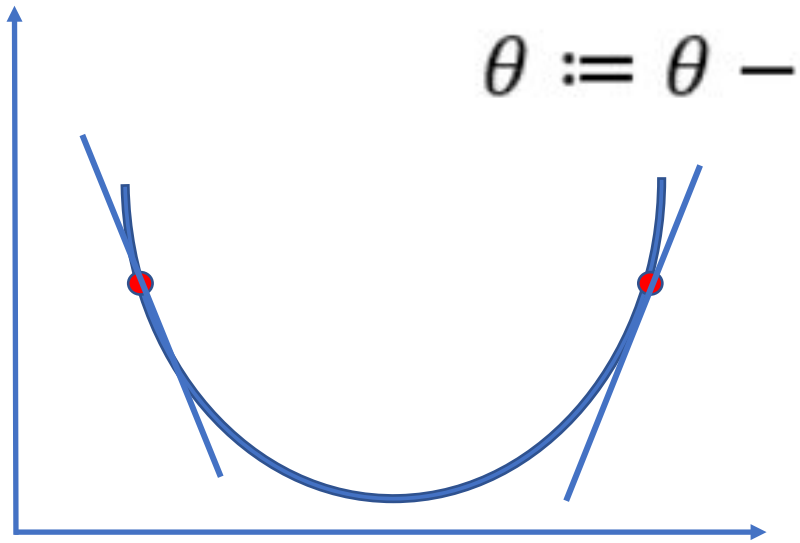
$$\lambda \underset{\alpha}{\text{or}} \sum_{j=1}^{p} \beta_j^2$$

Essentially the square of the Euclidean norm (magnitude) of β which is the vector of coefficients

- In this example if α is 0 we simply have normal regression equations
- If α is large or ∞ then the β approaches zero and only the constant term is available to predict y
- Essentially provides a weight on the squared residuals during the normal OLS process
- We want to train the regulator to fall between 0 and 1 using cross-validation in such a way that it minimize mean square error

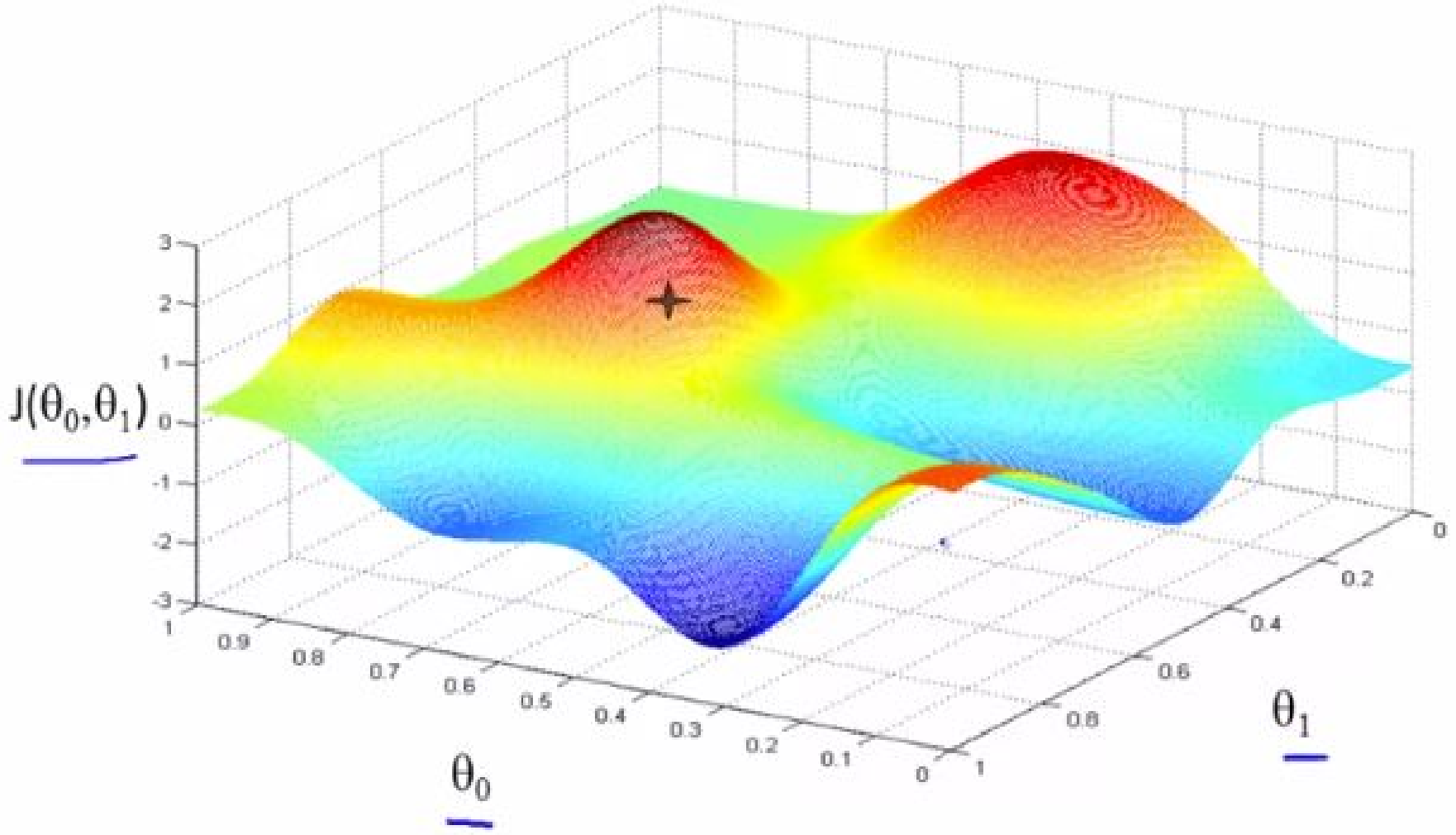
Gradient Decent

- Penalized regression methods use a very common algorithm called gradient decent
- It is essentially a step function that works to minimize some cost function through a iterative process



$$\theta := \theta - \alpha \frac{d}{d\theta} J(\theta)$$

Gradient Decent



Ridge regression is like least squares but shrinks the estimated coefficients towards zero. Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$, the ridge regression coefficients are defined as

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

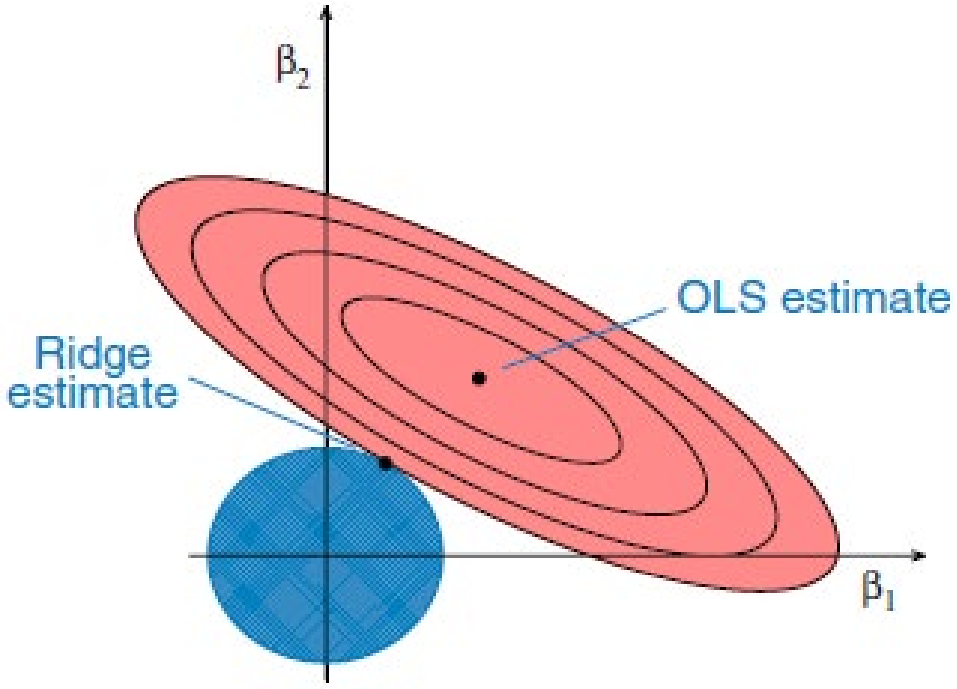
Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Note that:

- When $\lambda = 0$, we get the linear regression estimate
- When $\lambda = \infty$, we get $\hat{\beta}^{\text{ridge}} = 0$
- For λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients

L2 Norm or Euclidean Norm or Euclidean Length

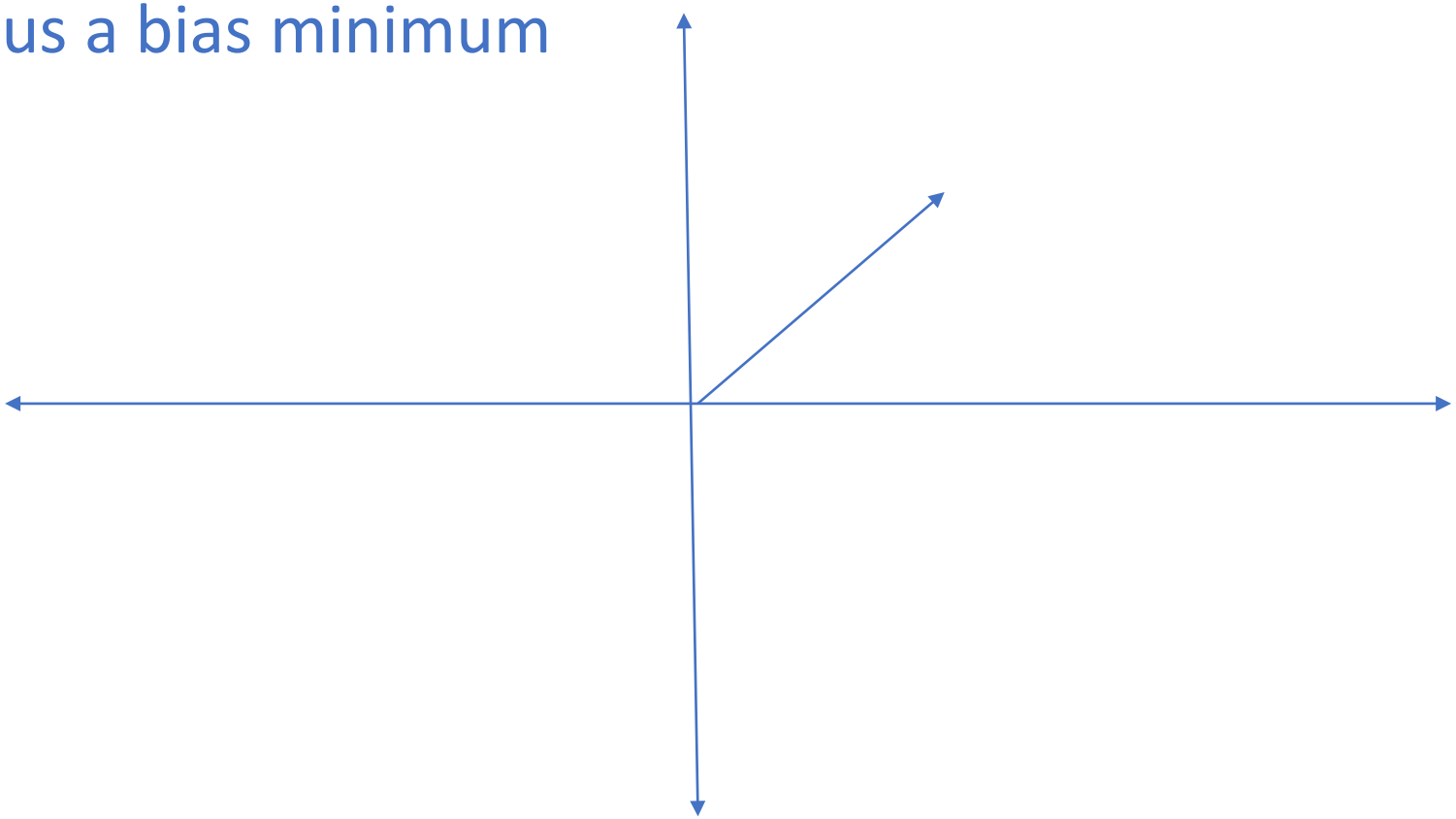
- Square of all the elements in matrix
- Sum the values together
- Take the square root
- Ridge also squares the final result of this, so we are taking the square of the Euclidean norm and multiply this number by the penalty to weight the coefficients

Ridge Visualization



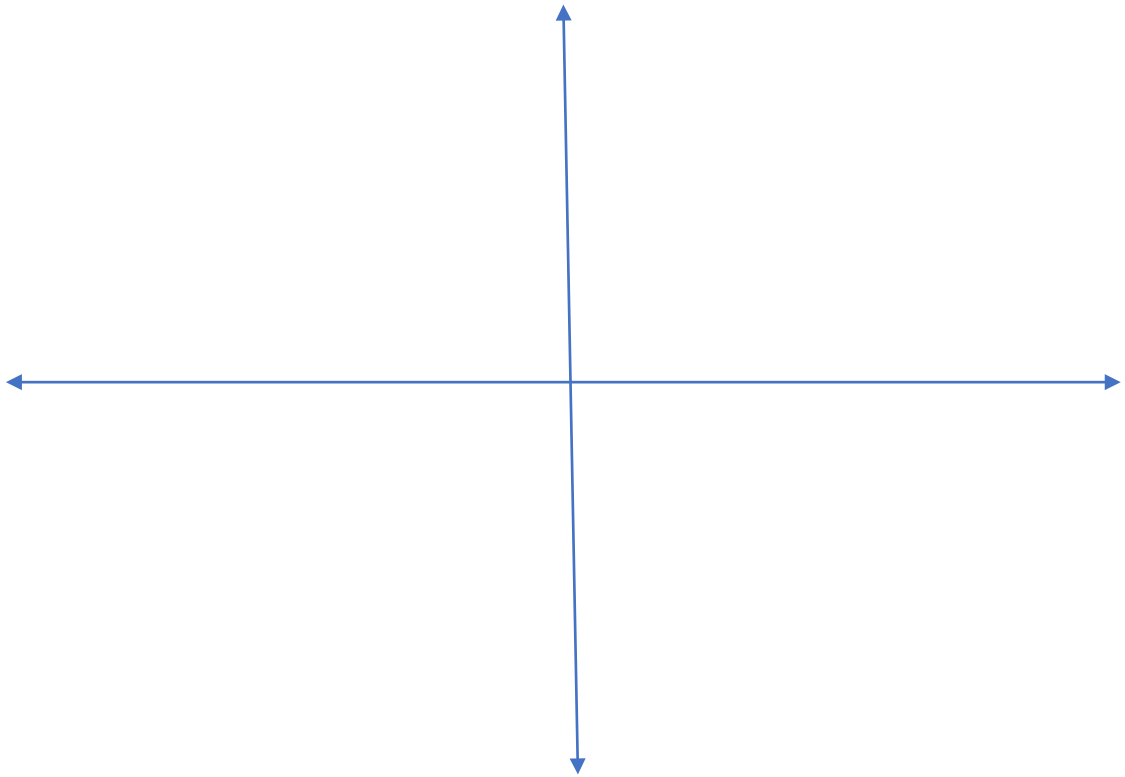
Measuring the Magnitude of Vectors, L2 and L1

- $\overrightarrow{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$
- L2 – Norm : $II\overrightarrow{\beta}\,II_2 = \sqrt{\beta_0{}^2 + \beta_1{}^2}$: What we used for ridge, essentially calculates the magnitude of the coefficient vector of the regression equation, gives us a bias minimum
- L1

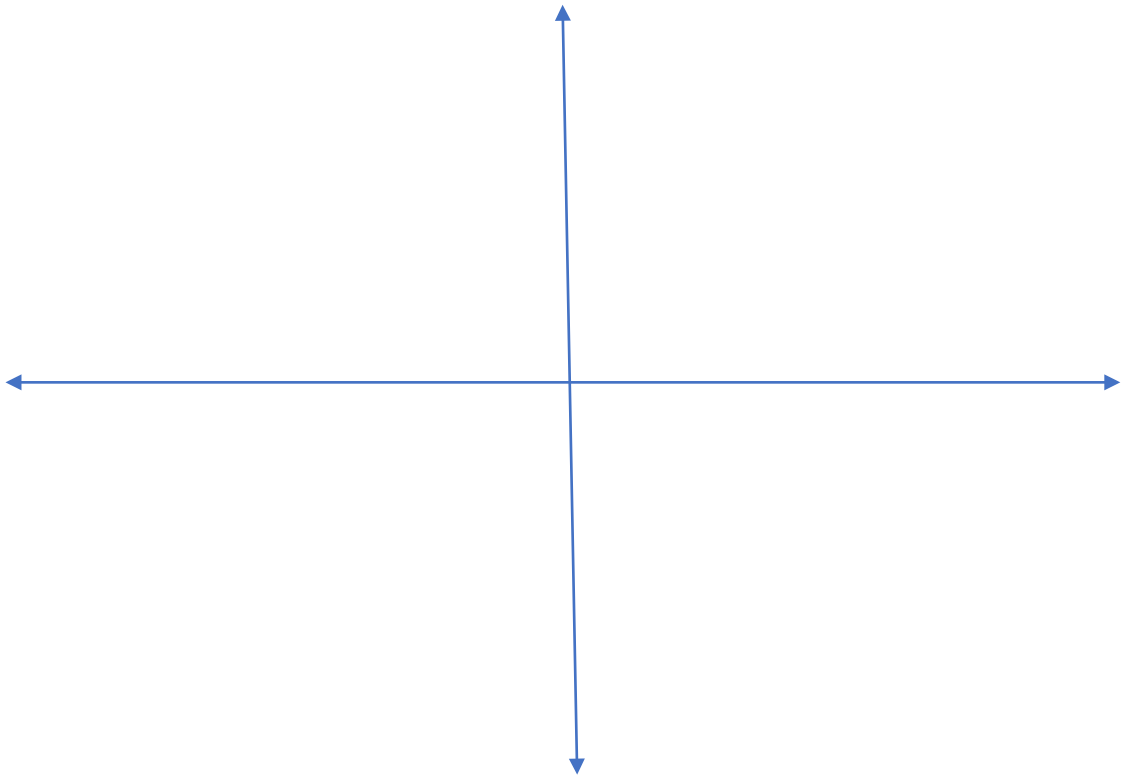


Measuring the Magnitude of Vectors, L2 and L1

- $\overrightarrow{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$
- L1 – Norm : $\|\overrightarrow{\beta}\|_1 = |\beta_0| + |\beta_1|$: What we used for ridge, essentially calculates the magnitude of the coefficient vector of the regression equation, gives us a bias minimum. Cartesian Distance or Euclidean Distance of our vector



Measuring the Magnitude of Vectors, L2 and L1



The lasso

The lasso[1] estimate is defined as

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

The only difference between the lasso problem and ridge regression is that the latter uses a (squared) ℓ_2 penalty $\|\beta\|_2^2$, while the former uses an ℓ_1 penalty $\|\beta\|_1$. But even though these problems look similar, their solutions behave very differently

Note the name "lasso" is actually an acronym for: Least Absolute Selection and Shrinkage Operator

[1] Tibshirani (1996), "Regression Shrinkage and Selection via the Lasso"