k-Nearest Neighbors (kNN)

Kristie Wirth
Jon Courtney
Frank Burkholder
Ryan Henning

galvanize

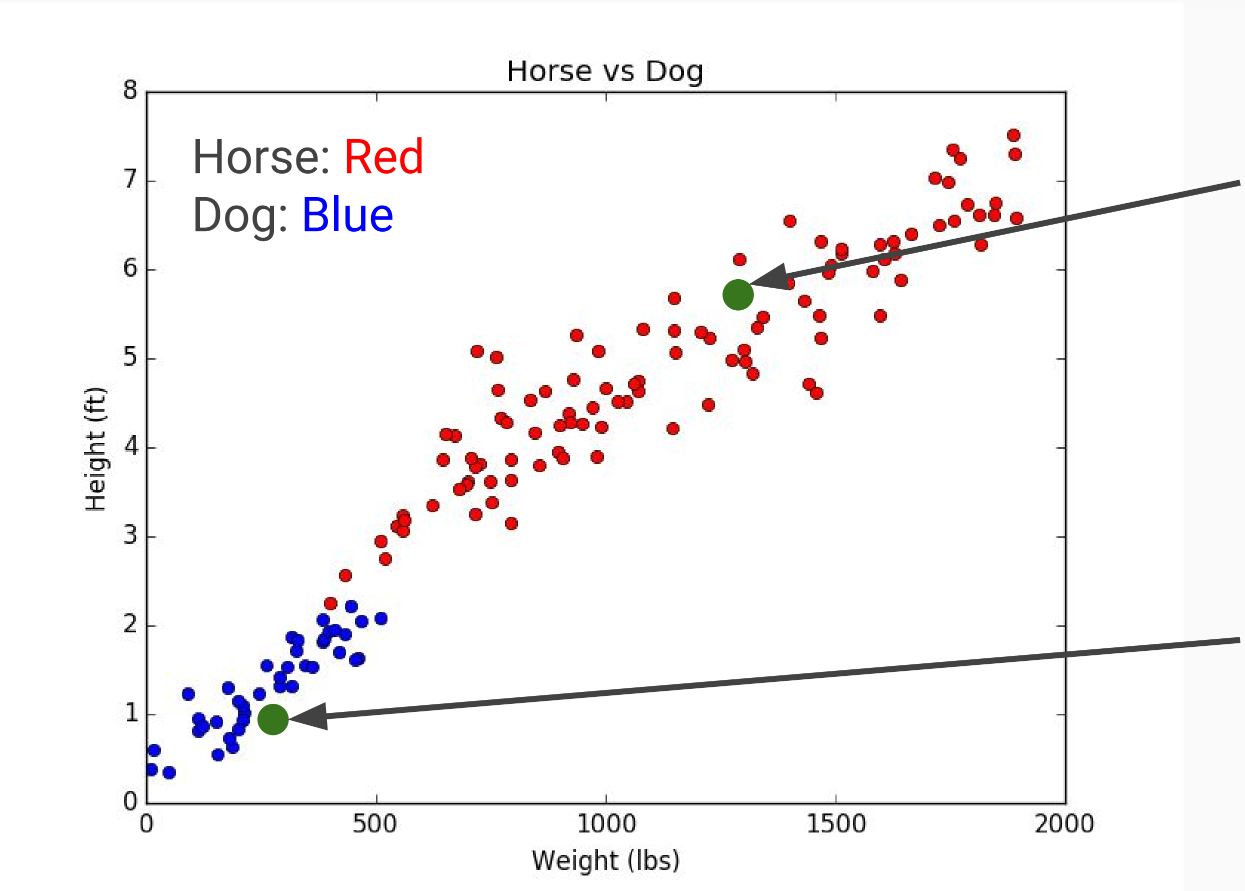
Learning Objectives

- Explain the difference between parametric and non-parametric models
- Know the hyperparameters for kNN and how the algorithm works
- Compute common distance metrics used in kNN
- Understand the problems posed by high-dimensional data
- Know the advantages & disadvantages of using kNN

Parametric vs. Non-Parametric Models

Parametric	Non-Parametric
Fixed model structure	Flexible model structure
Fixed type/number of parameters	Flexible type/number of parameters
Some assumptions about the data	Fewer assumptions about the data
Easy to understand & interpret	Harder to interpret
Can train the model very quickly	Model takes longer to train
Can handle smaller datasets	Needs larger amounts of data
Performance can be poor	Generally good performance
Typically less overfitting problems	Often problems with overfitting

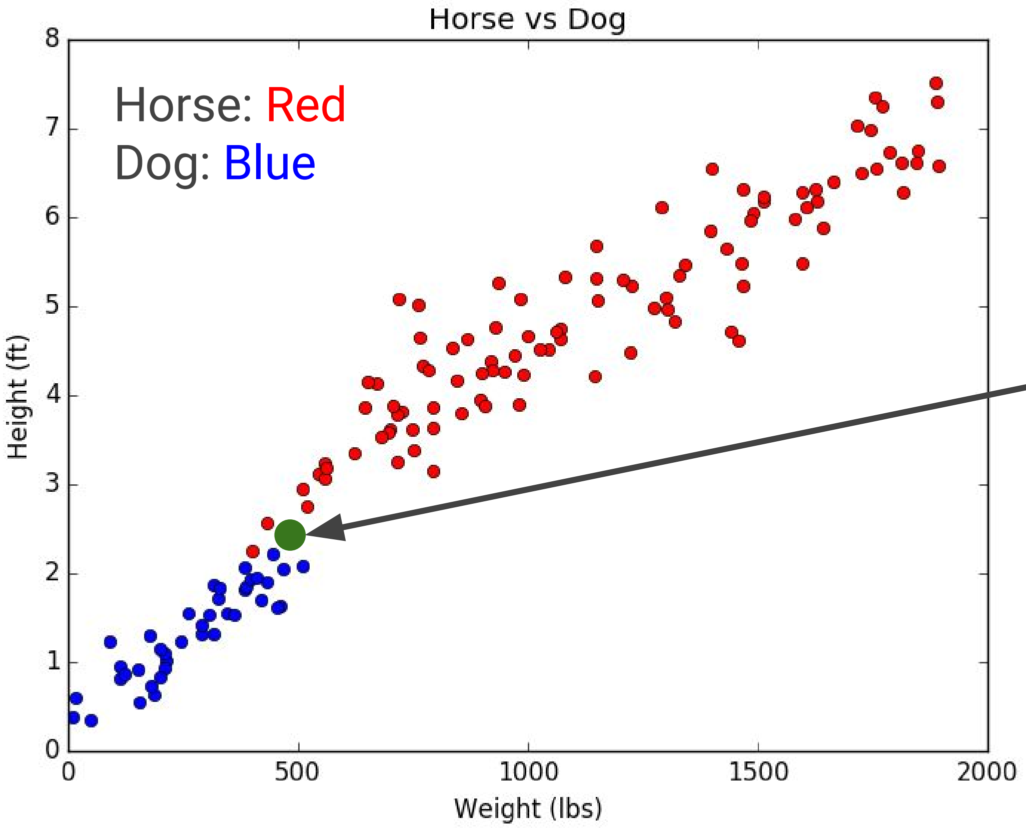
Horses and Dogs Data



New datapoint:
Is it a horse or a dog?

New datapoint:
Is it a horse or a dog?

Horses and Dogs Data



New datapoint:
Is it a horse or a dog?

What is kNN?

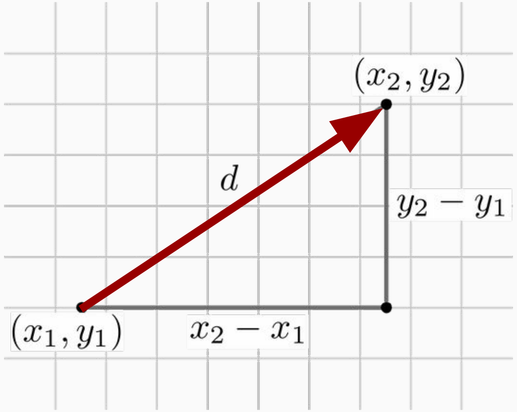
- Uses information about similar datapoints to predict information about a given datapoint
- Example 1: Predicting type of animal (dog or horse) based on animals with similar characteristics (**classification**)
- Example 2: Predicting the selling price of a house based on houses with similar characteristics (**regression**)

Overview of kNN Algorithm

1. Choose a value for the hyperparameter k - how many neighbors do you want to look at for a given data point?
2. Calculate the distance all data points are from each other
3. Find the closest k points to each data point, i.e. its neighbors
4. Make a prediction for each data point
    a. For classification, assign a data point's category based on what category the majority of its neighbors are (e.g., if 3 neighbors are dogs and 1 neighbor is a horse, then you classify that point as a dog)
    b. For regression, calculate a data point's value by taking the average value of its neighbors

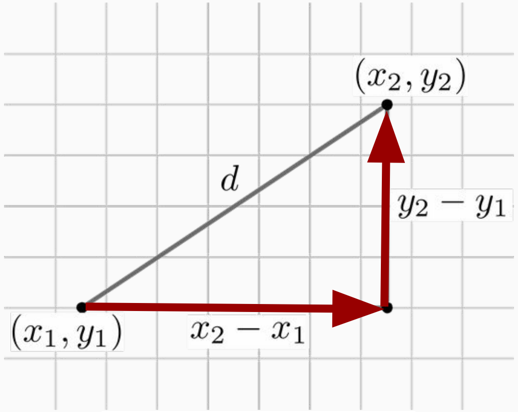
Ways to Measure Distance

Euclidean Distance



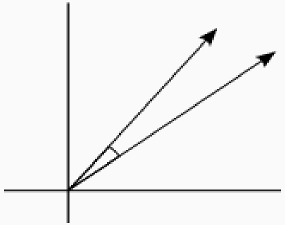
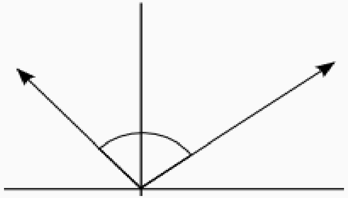
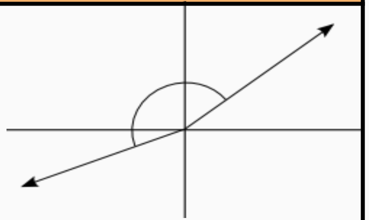
$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Manhattan Distance



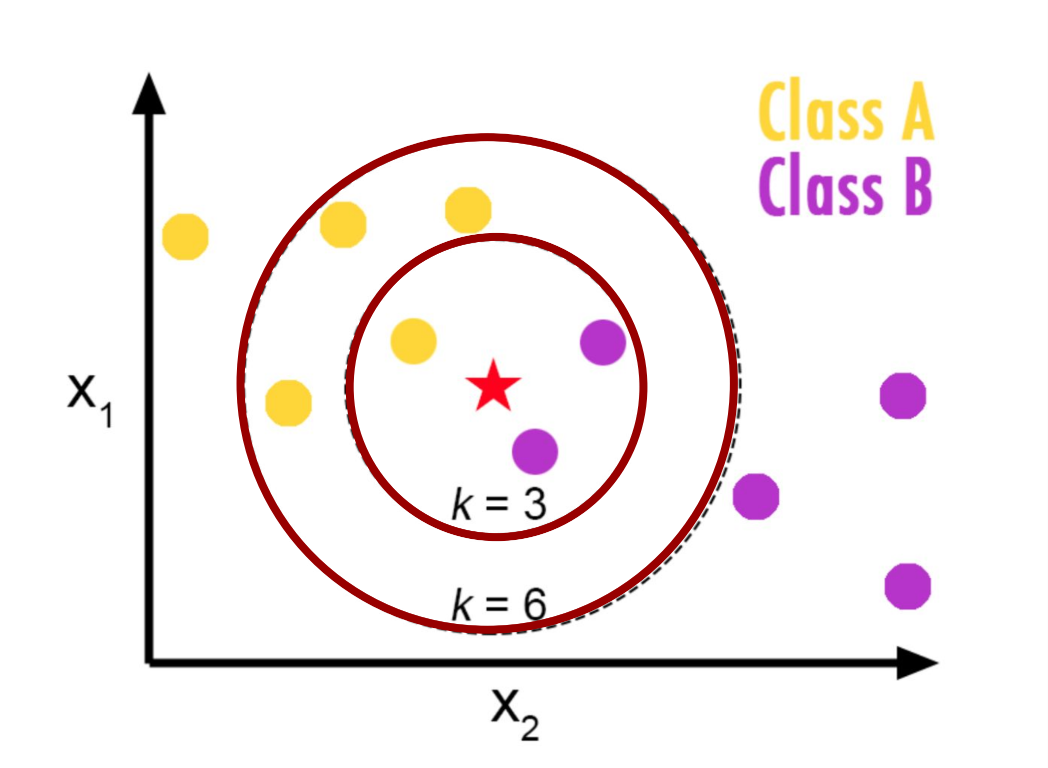
$$\sum_{i=1}^{n}|p_i - q_i|$$

Ways to Measure Distance

Cosine Similarity	1	0	-1
Cosine Distance (1 - Cosine Similarity)	**0**	**1**	**2**
	 Same direction	 Right angle/90 degrees	 Opposite directions

$$1 - \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|_2\|\mathbf{B}\|} = 1 - \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

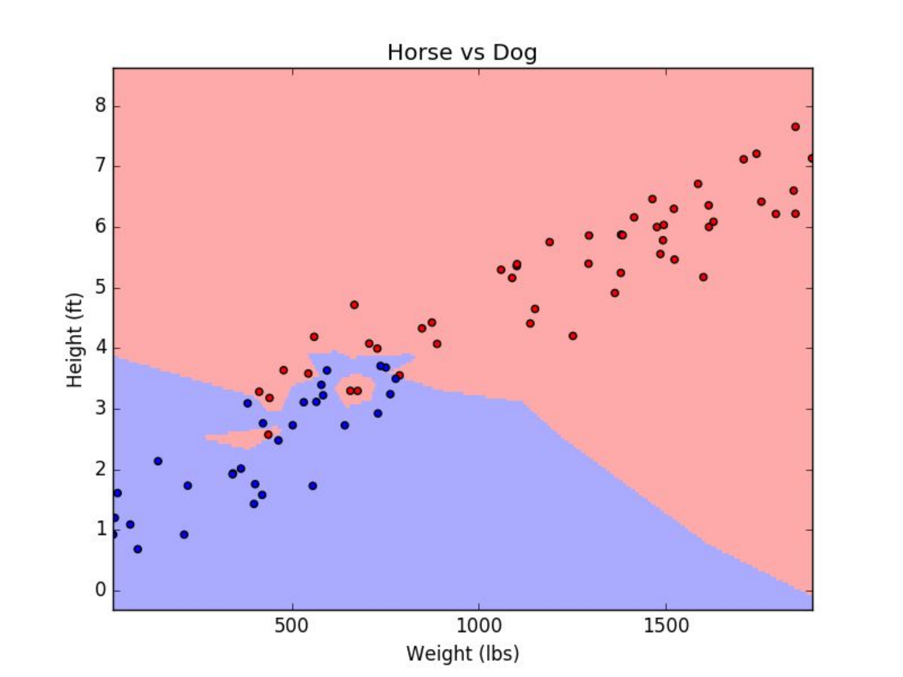
Choosing a Value for k



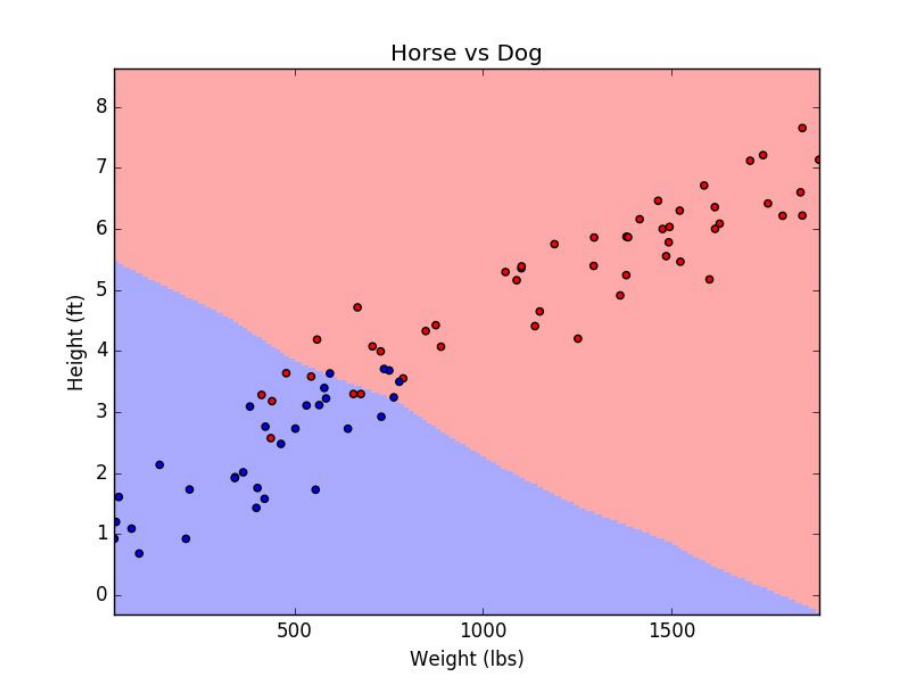
- Common starting point: k = √n
- Use cross validation to find the best value for k - try different values and score how well they do for your data

What are the Effects of Different Values of k?

k=1



k=50



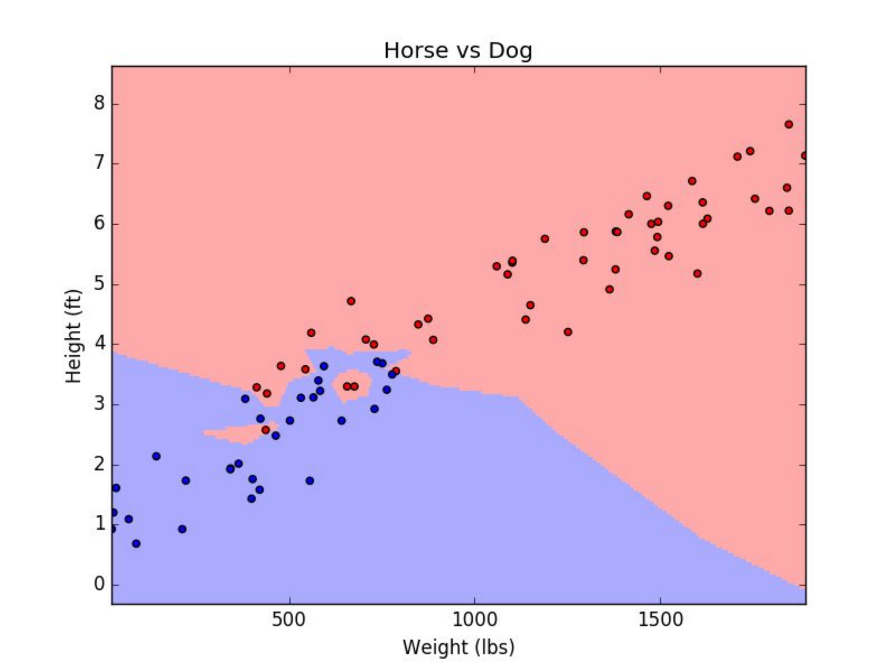
Smaller values for k = more likely to overfit

Higher values of k = less complex model

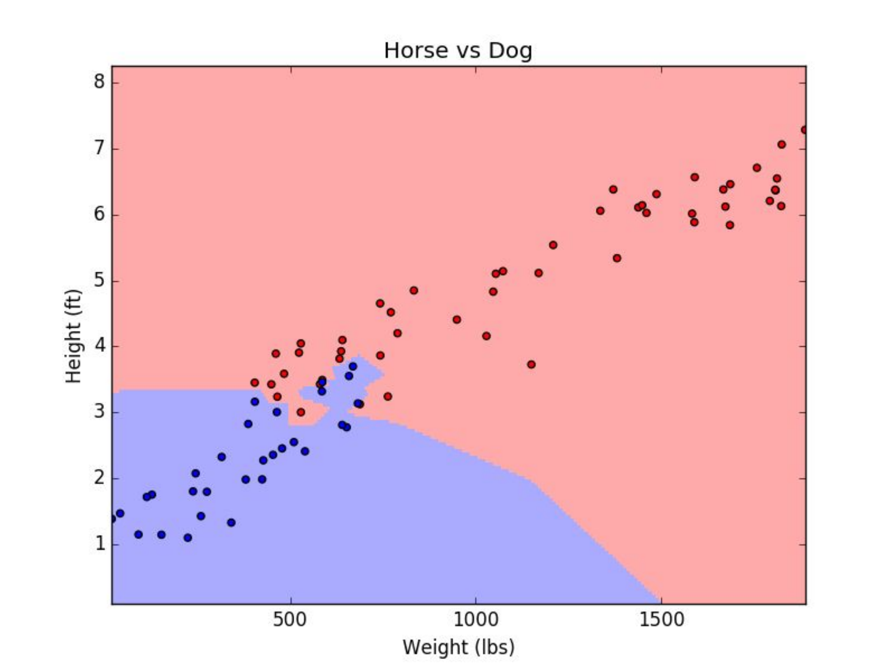
Smaller k = Higher Model Variance

Each training dataset is randomly generated from the same population.

k=1



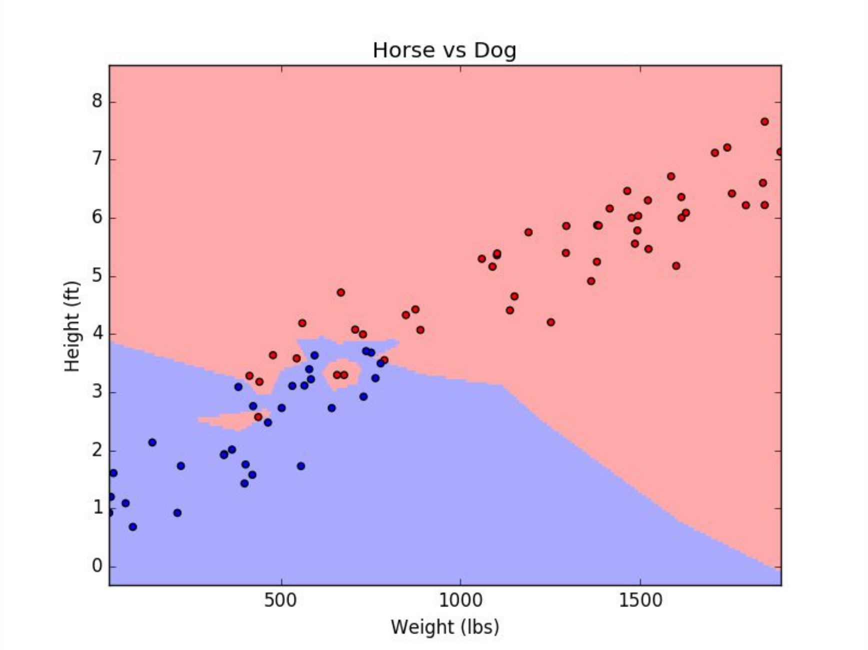
k=1



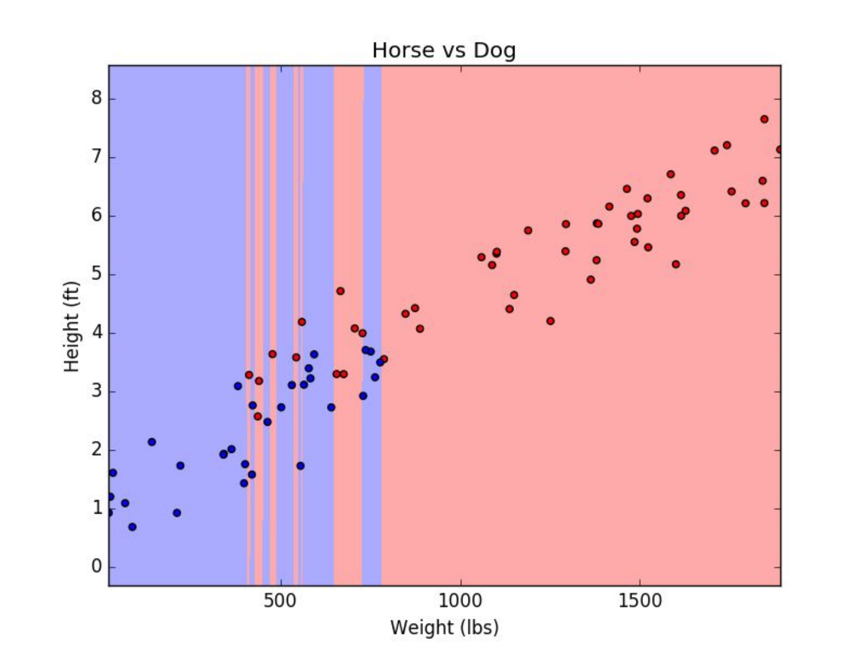
Don’t Forget to Scale Your Data!

galvanize

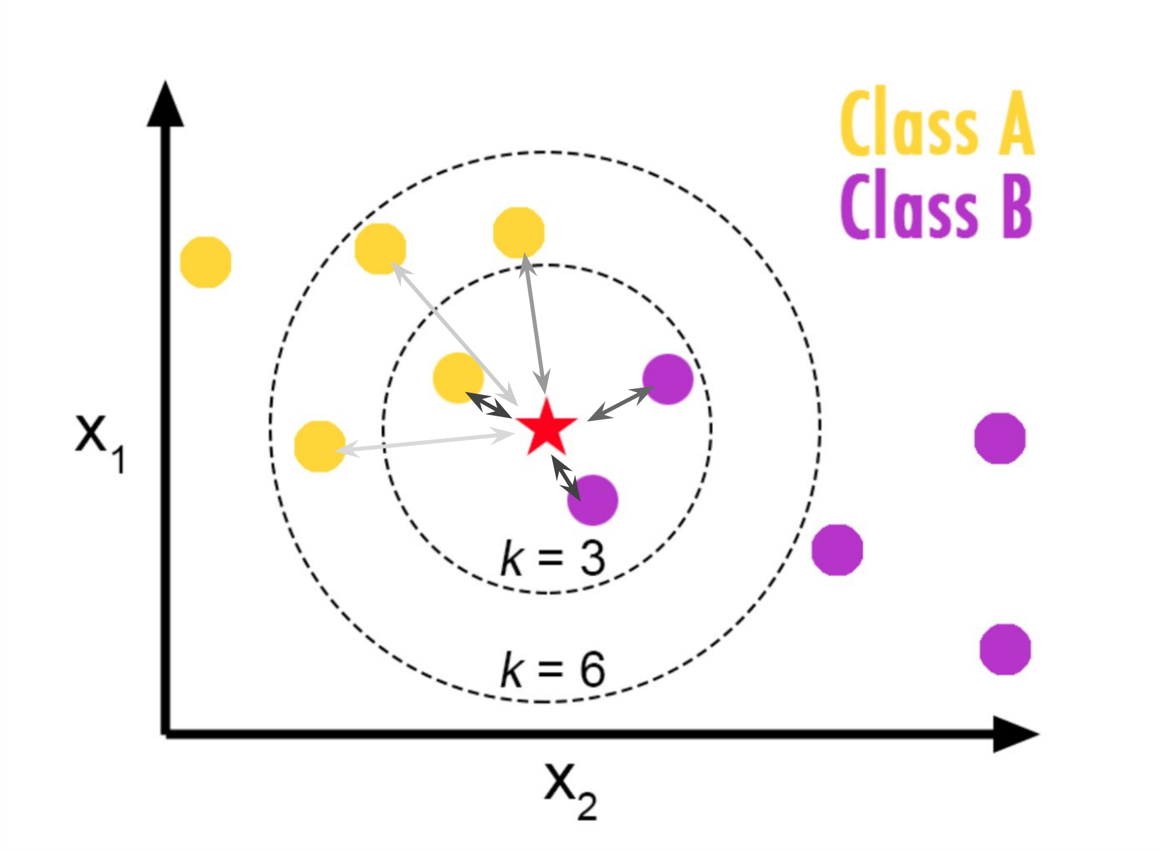
k=1, scaled features



k=1, original-scale features



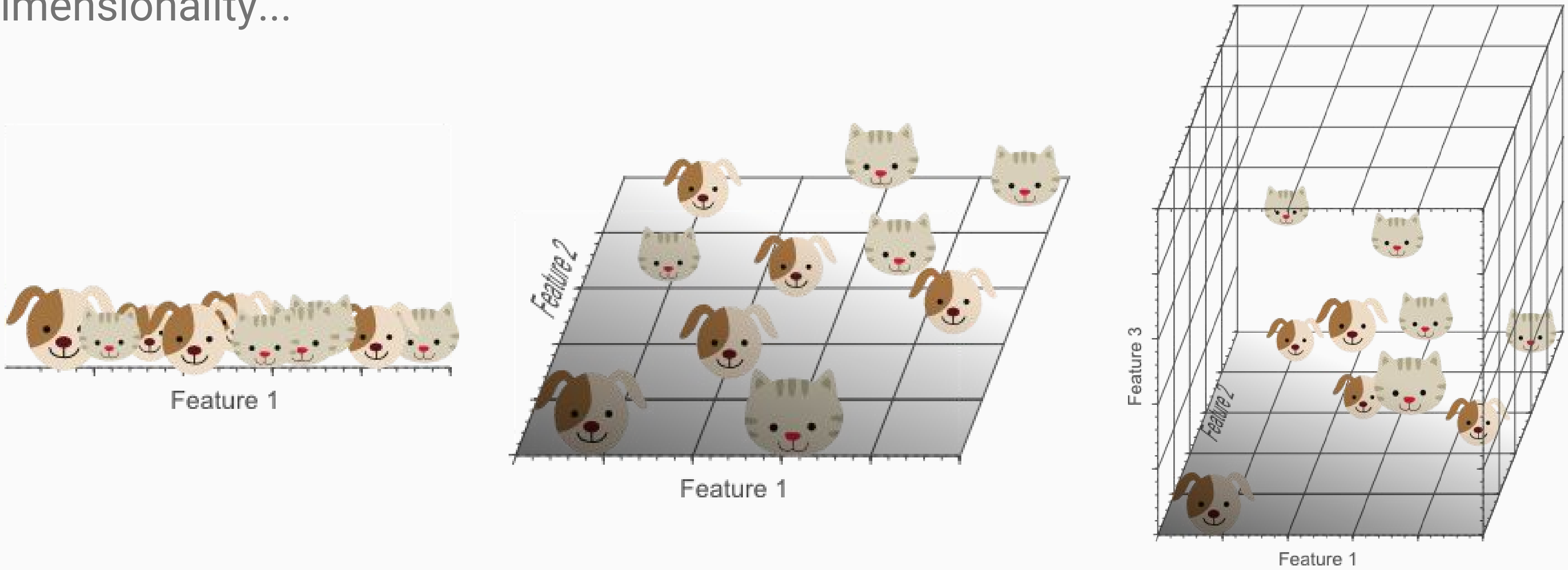
A Variation on kNN: Point Weighting



- Points that are closer are considered more important than points that are farther away
- The *i* point "votes" with a weight of 1 / distance $\frac{1}{d_i}$
- Shorter distance - higher weighted "vote" for its own category

Review: The Curse of Dimensionality

kNN does not work well in 5+ dimensions. Why is this true? Let’s review The Curse of Dimensionality...

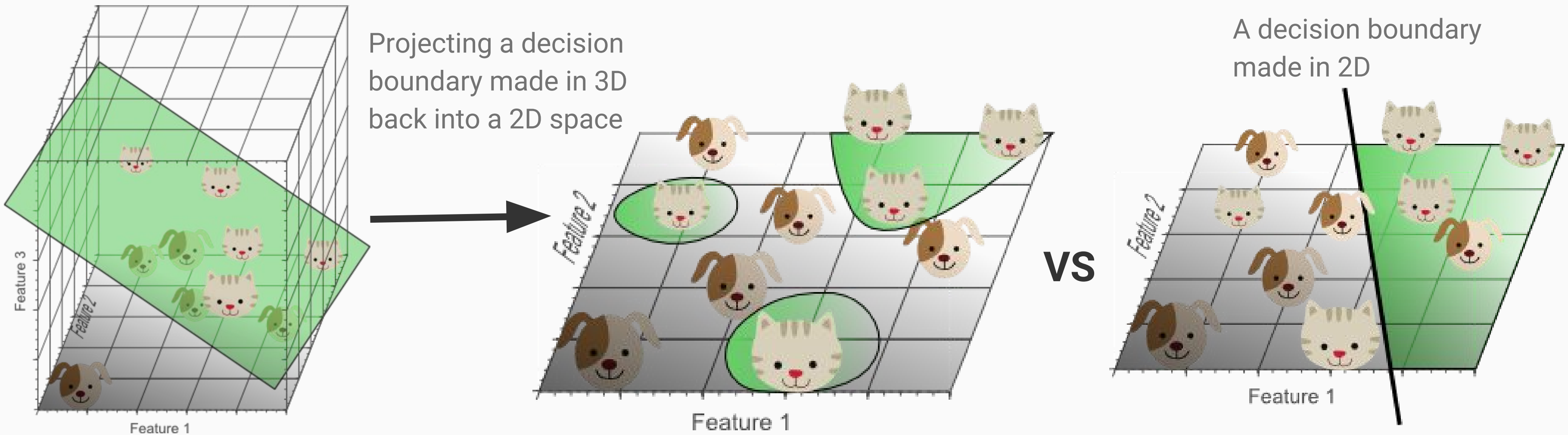


In higher dimensions, data is more sparse/spread out more.

http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Review: The Curse of Dimensionality

Sparse data often leads to overfitting in these higher dimensions.



Projecting a decision boundary made in 3D back into a 2D space

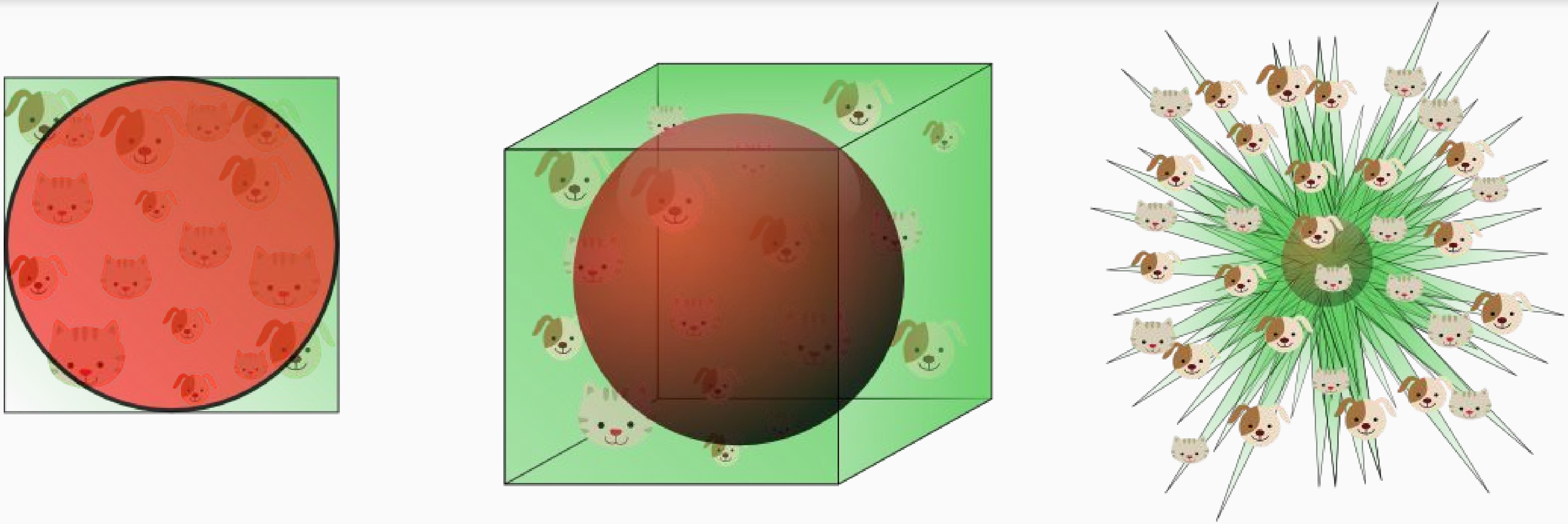
VS

A decision boundary made in 2D

If you are moving from a 1 feature space with 100 samples to a 10 feature space, you would need 100,000,000,000,000,000,000 samples to achieve the same sample density and thus avoid overfitting!

Also, the spareness is not evenly distributed over the entire space...

kNN & The Curse of Dimensionality



- Samples that are outside the red circle are farther away from the other points
- The volume of the red circle/hypersphere becomes smaller in higher dimensions
- In high dimensional space, most of the points are not in this red hypersphere
- Distance measurements between points (which are the basis of kNN) become so large they are almost meaningless

Advantages & Disadvantages of kNN

Advantages:

- Simple to implement
- The training phase is just storing the data in memory
- Works with any number of classes/categories
- Easy to add in additional data
- Only two hyperparameters - k & the distance metric

Disadvantages:

- Poor performance in high dimensions
- Very slow to run, especially with large datasets
- Categorical features don't work well with kNN

Check in Questions

- What is the difference between parametric and non-parametric models?
- What are the hyperparameters for kNN?
- Describe the steps in the kNN algorithm.
- What are some of the possible distance metrics used in kNN?
- How does the Curse of Dimensionality affect kNN?
- What are some advantages of using kNN?
- What are some disadvantages of using kNN?