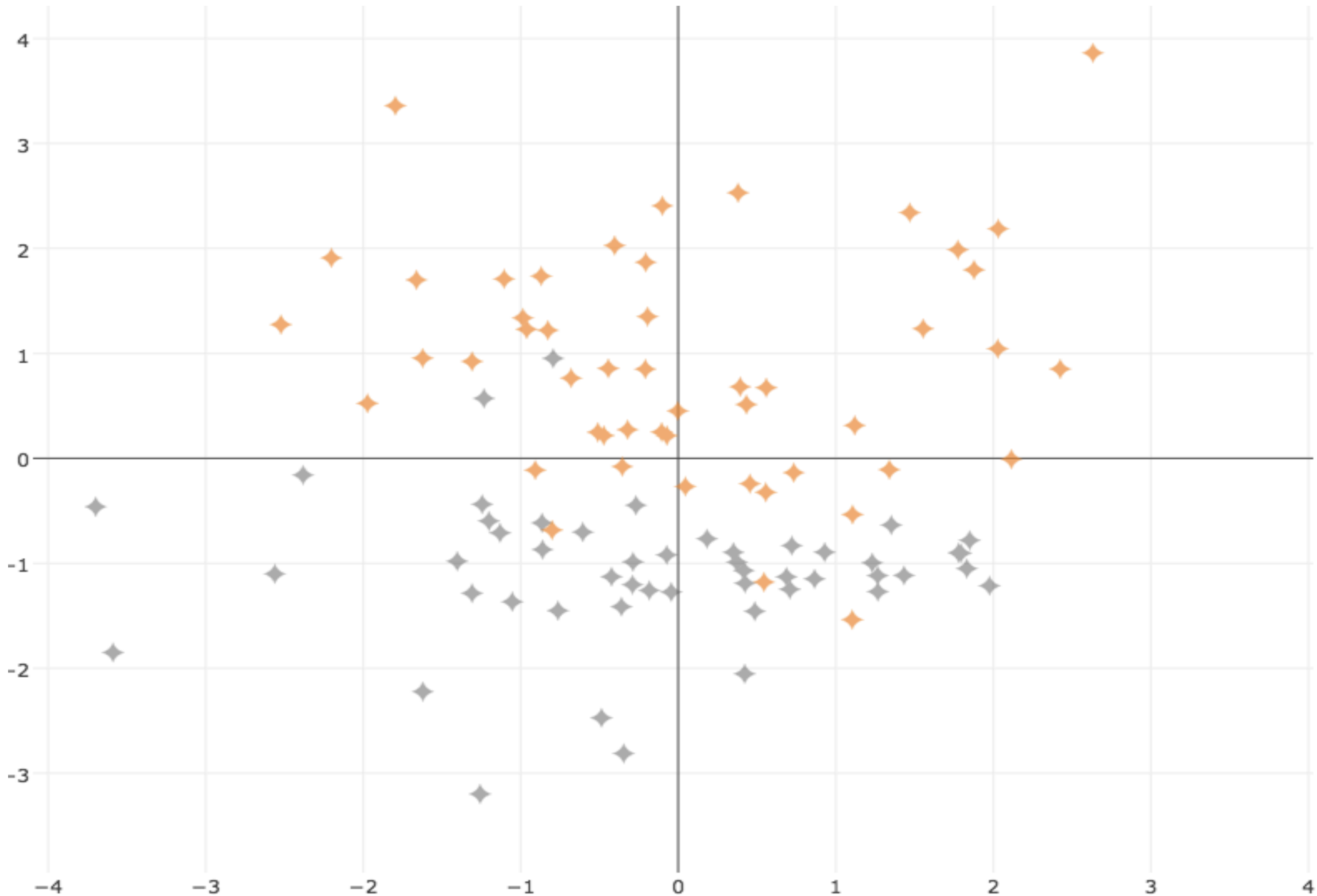
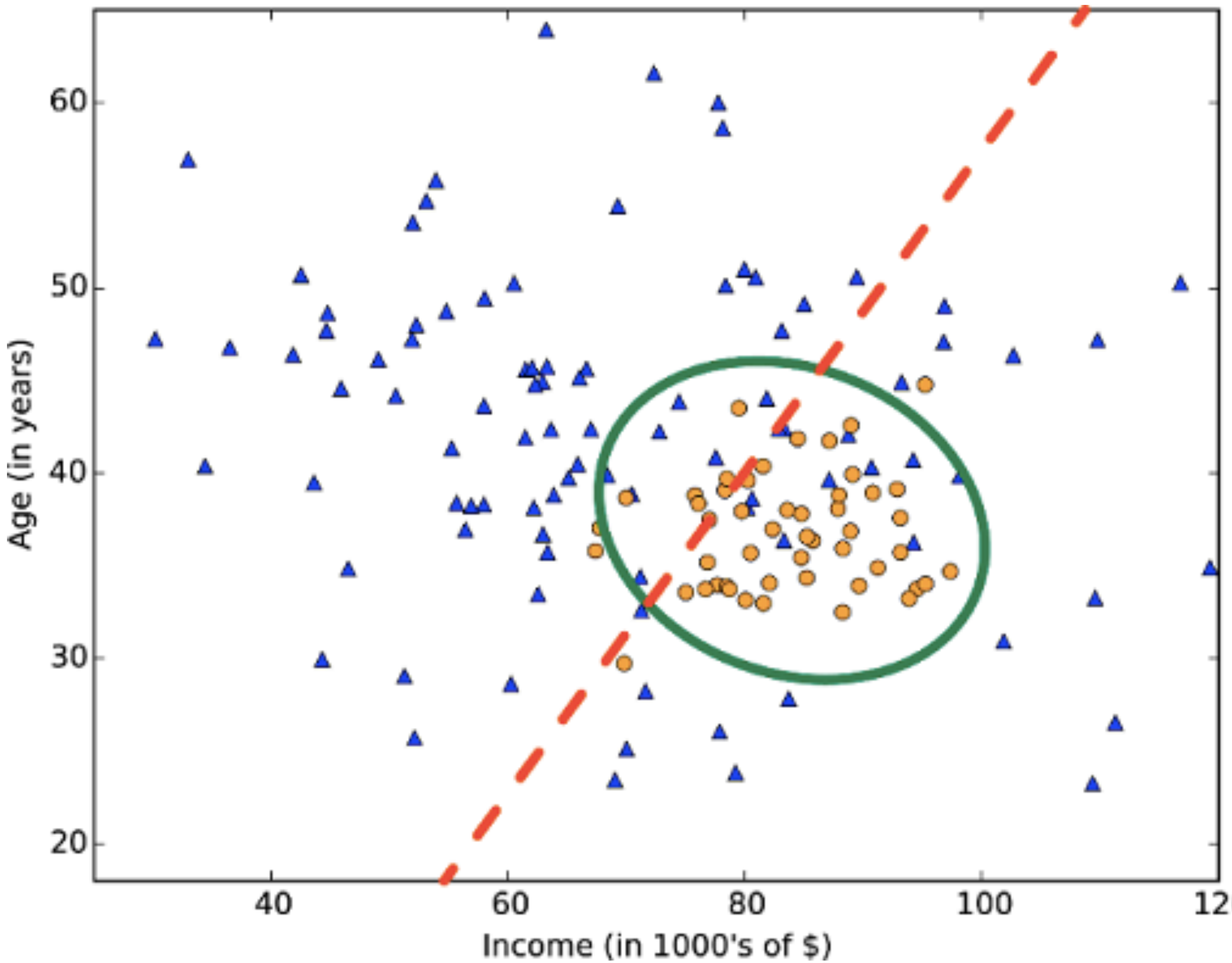
k Nearest Neighbors

school

Classify a new observation



How about here?



k Nearest Neighbors

kNN is a simple yet powerful algorithm capable of handling such non-linearities. Invented in the 1950s, it is at times referred to as the first machine learning method.

PSEUDOCODE:

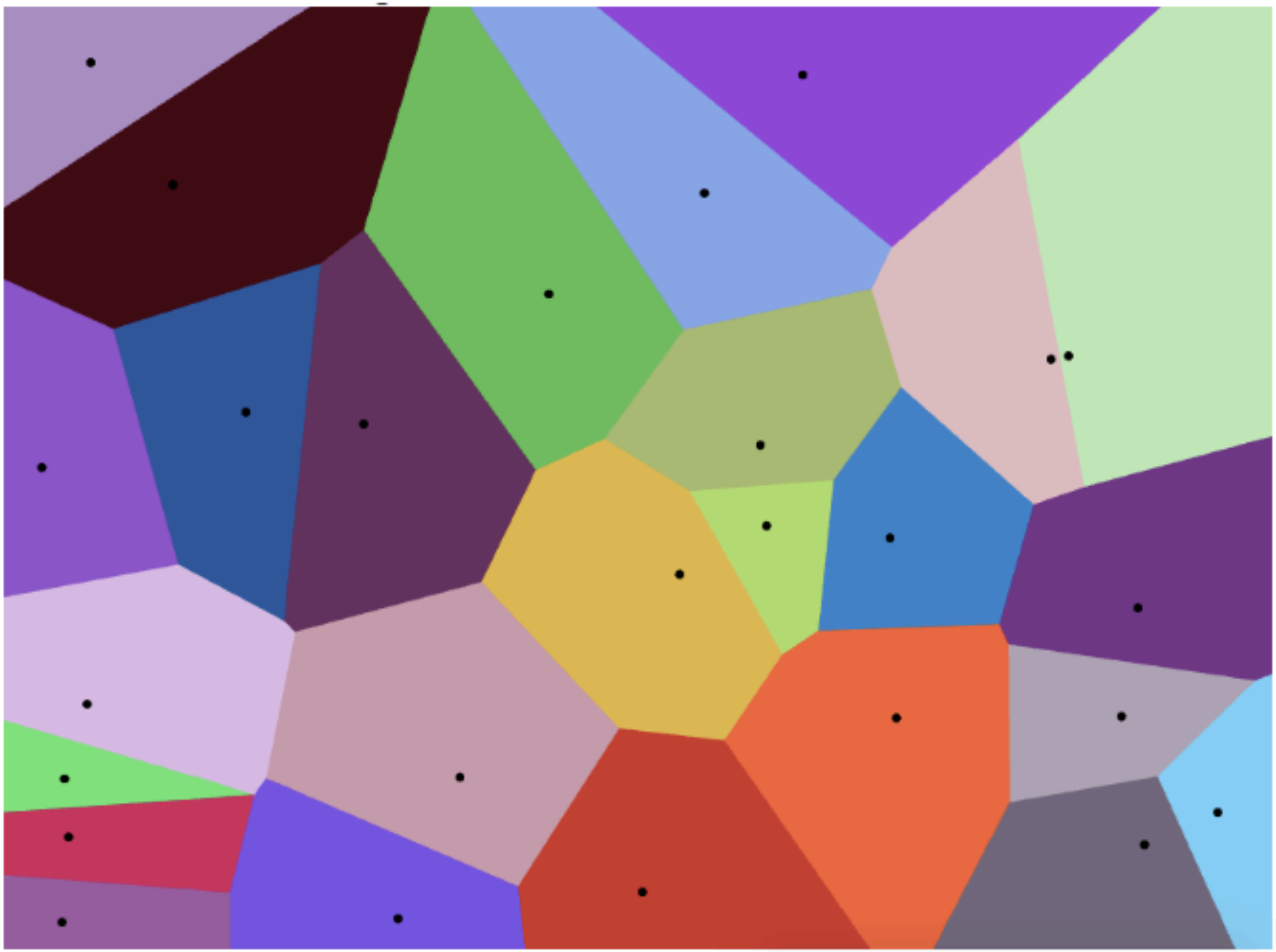
- Find k nearest neighbor to point of interest
- Count how many of those k neighbors are of each class
- Classify the point of interest as the majority class

kNN, is therefore ...

A Non-parametric learner – As it makes no assumptions about the distribution of the training data, which is great!

A lazy learning algorithm – As it makes no generalizations on the data during training, instead postpones effort to when having to fit test data.

kNN decision boundaries



See iPython notebook

kNN assumptions

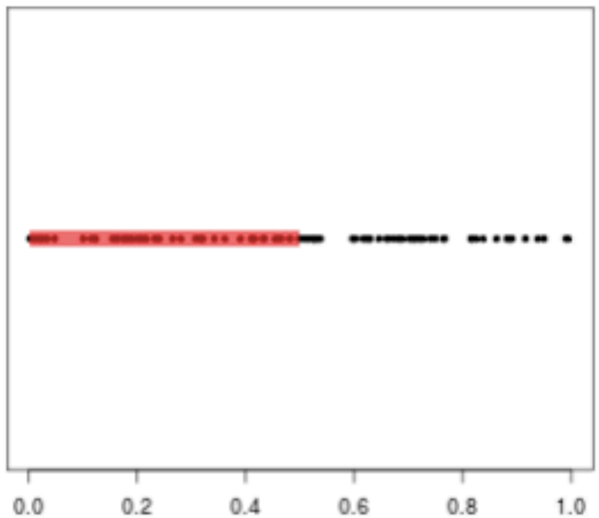
kNN builds on the assumption that your training data describes a feature space. Your training set has to cover this feature space densely and preferably uniformly.

As a result,

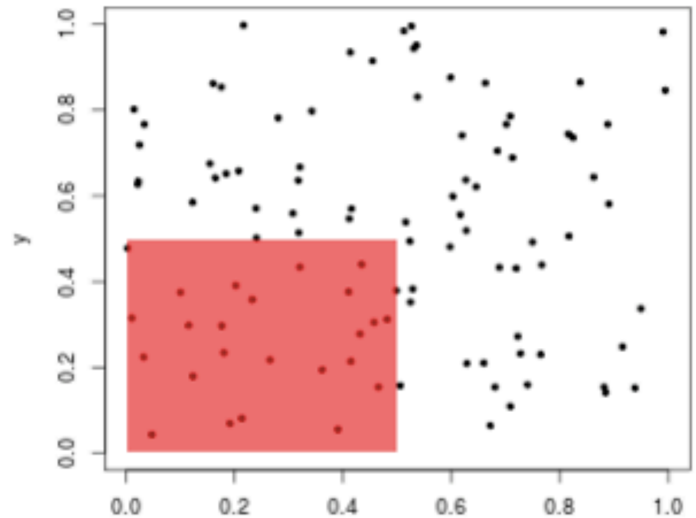
- Unbalanced classes pose problems
- What happens when your features space grows?
  Say hello to the "curse of dimensionality"

Curse of dimensionality

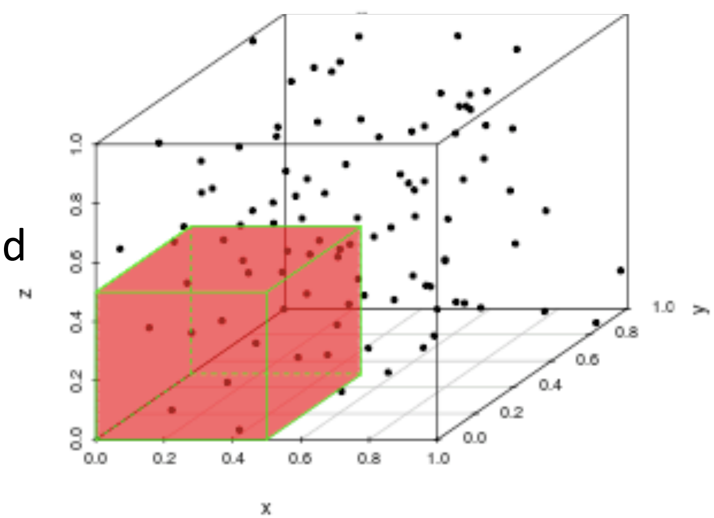
1d



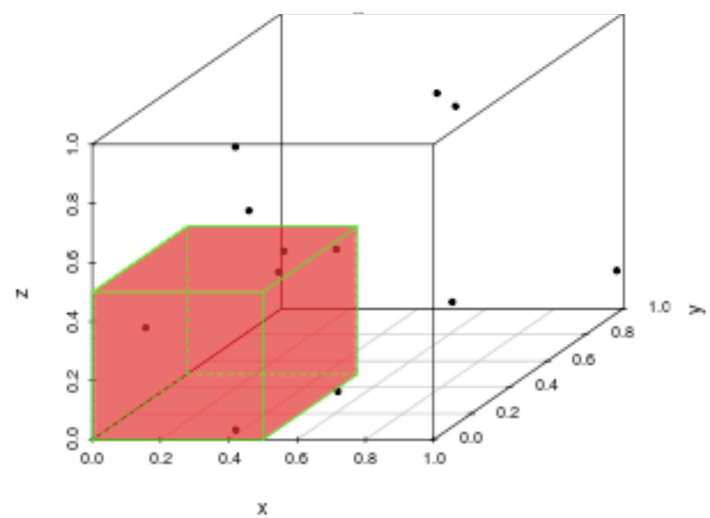
2d



3d



4d
t= 0



https://prpatil.shinyapps.io/cod_app/

Distance metrics

Inherent idea of distance. Most common choices are,

- Euclidean distance (L^2 norm)

$$\left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}} = \|\vec{x}\|_2$$

- Cosine similarity

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

- Other choices are:
  Manhattan distance, L^p norm, L^infinity norm, Hamming distance etc

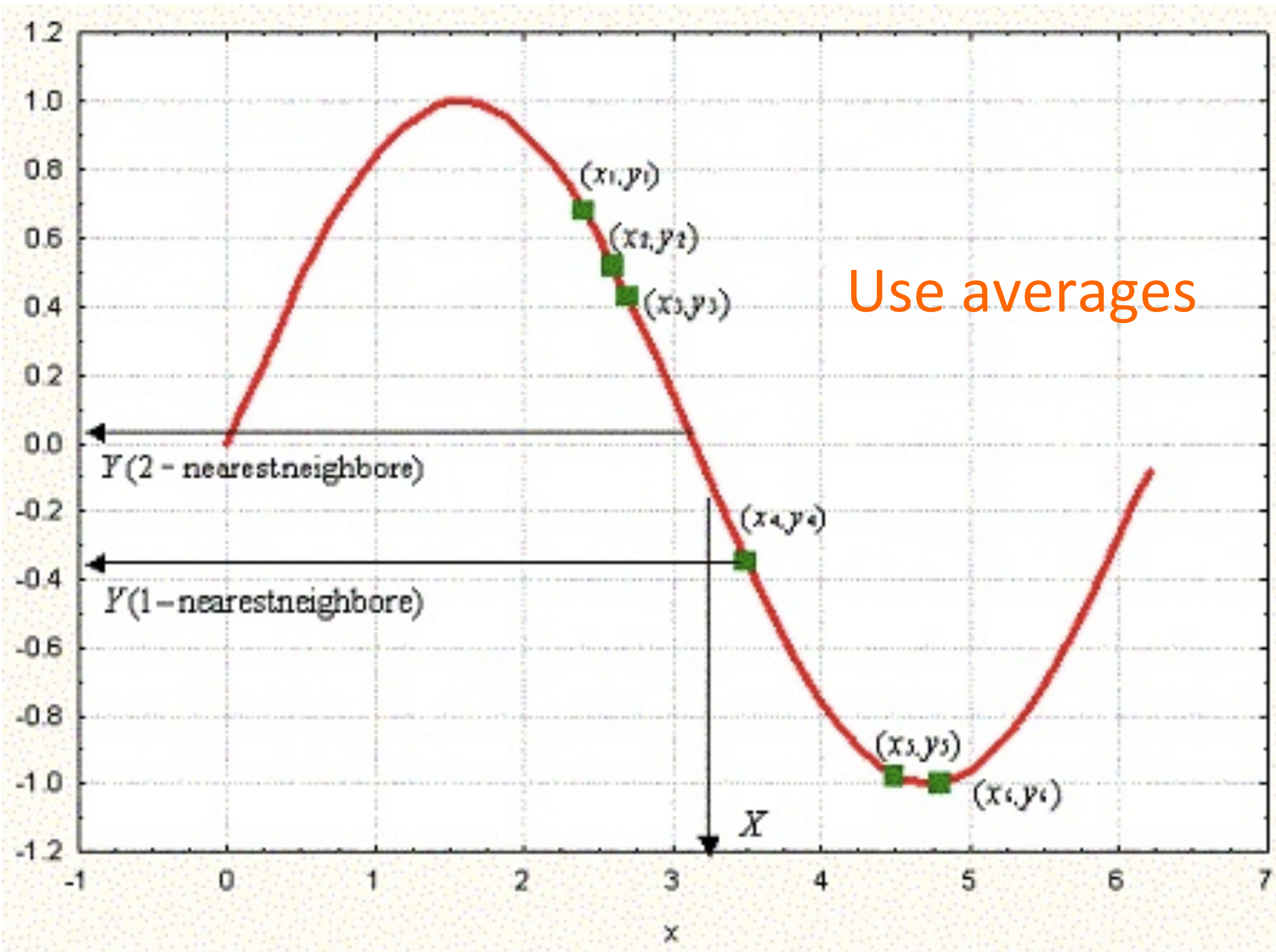
Should we scale the feature space?
What about categorical features?
Which distance metric should we pick?

Choosing k

- Cross – Validation
- As a rule of thumb: Start with $k = \sqrt{N}$

What happens when $k = N$?
How about we pick $k = 1$?

kNN for regression



http://www.statsoft.com/textbook/k-nearest-neighbors

kNN variants

- Weighted kNN
- Edited kNN

Both somewhat logical leaps from kNN

kNN Summarized

Use kNN when you have a large training set spanning a small feature space and your classes are balanced.

PROS

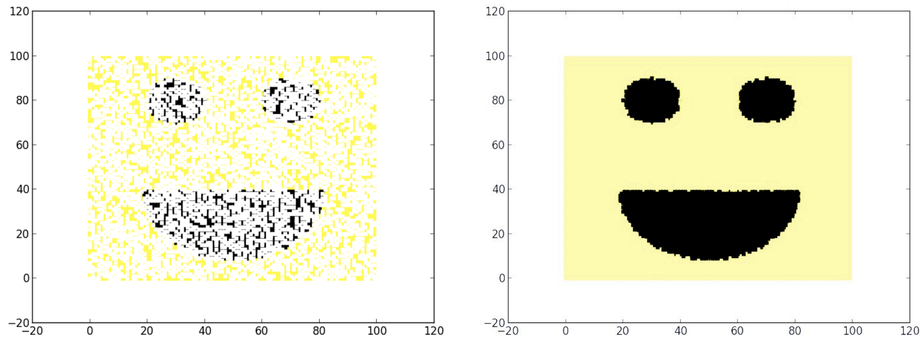
- Works with any number of classes
- Easy to store the model
- Can learn very complex functions

CONS

- Slow ☹
- Irrelevant attributes can affect results
- Watch out for the curse of dimensionality.

kNN use cases

- Classification
- Imputation



Read deal human face completion examples on scikit
http://scikit-learn.org/stable/modules/neighbors.html

- Anomaly detection

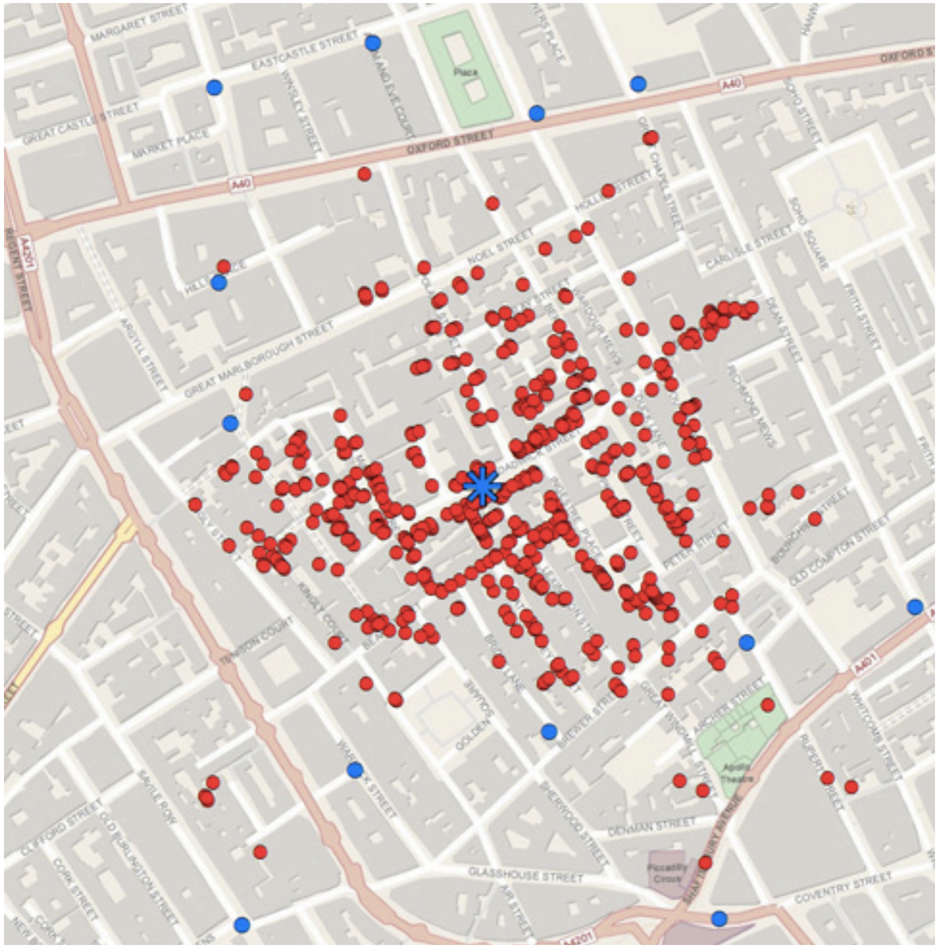
Distance to kth neighbour gives a metric on the outlier score

Down the rabbit hole...

Nearest Neighbors & John Snow

Famous successful example of nearest neighbors even before kNN was invented

https://plus.maths.org/content/uncovering-cause-cholera



kNN error bounds

Theoretical guarantees about the error bounds as N approches infinity.

Works out to be less than twice the optimal error (Bayesian Error). Pretty cool for a “lazy” technique.

Disclaimer: Poor pun usage

kNN, voronoi diagrams & tessellation

Voronoi diagraom on right, courtesy Mysid (SVG), Cyp (original) - Manually vectorized in Inkscape by Mysid, based on Image:Coloured Voronoi 2D.png., CC BY-SA 3.0, $3

Material on kNN

Quick but comprehensive read:
[Linked here]

Also comprehensive but longer:
http://www.scholarpedia.org/article/K-nearest_neighbor

The motherlode:
"Pattern Classification" by Duda and Hart