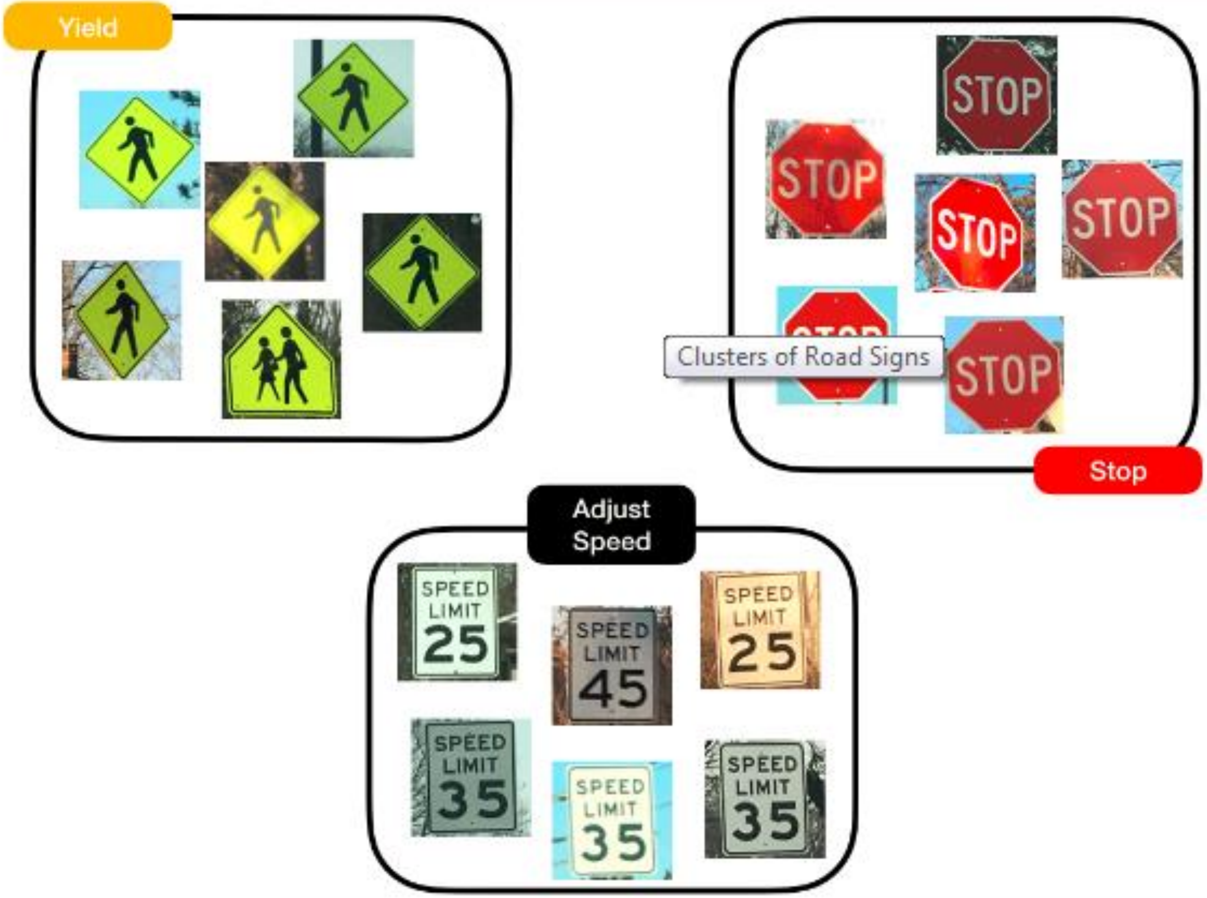
K-Nearest Neighbors

Classification tasks for driverless cars

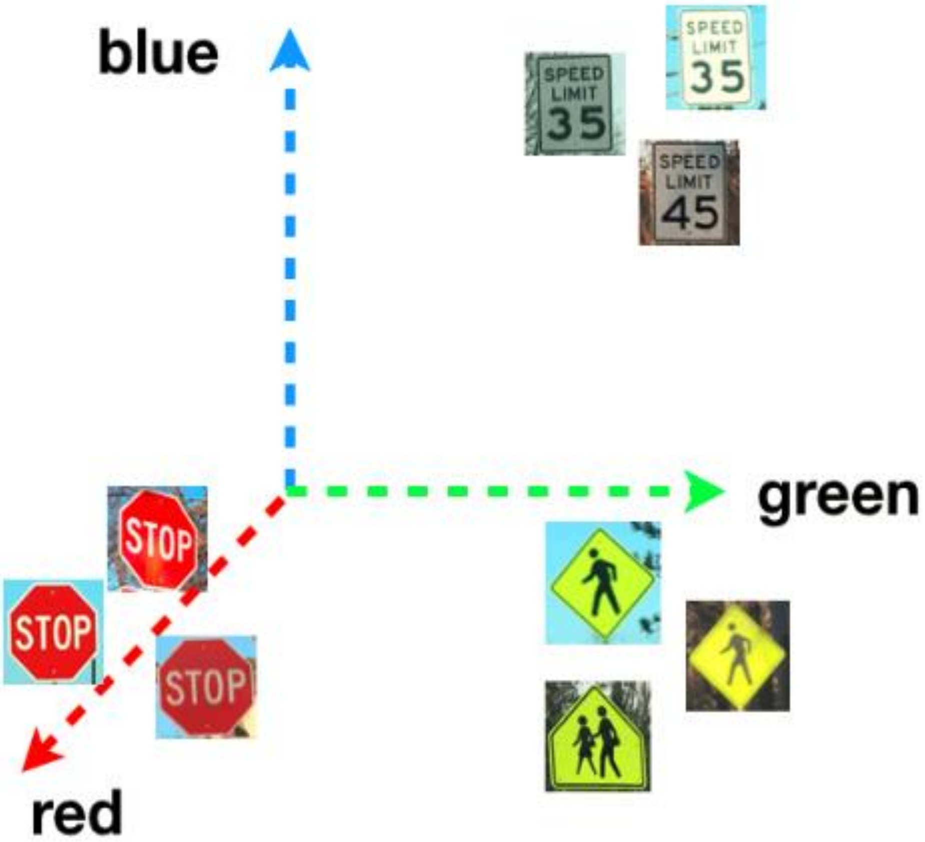
Understanding Nearest Neighbors



Basic Idea

- For a given record to be classified, identify nearby records
- “Near” means records with similar predictor values $X_1, X_2, \ldots X_p$
- Classify the record as whatever the predominant class is among the nearby records (the “neighbors”)

Measuring similarity with distance



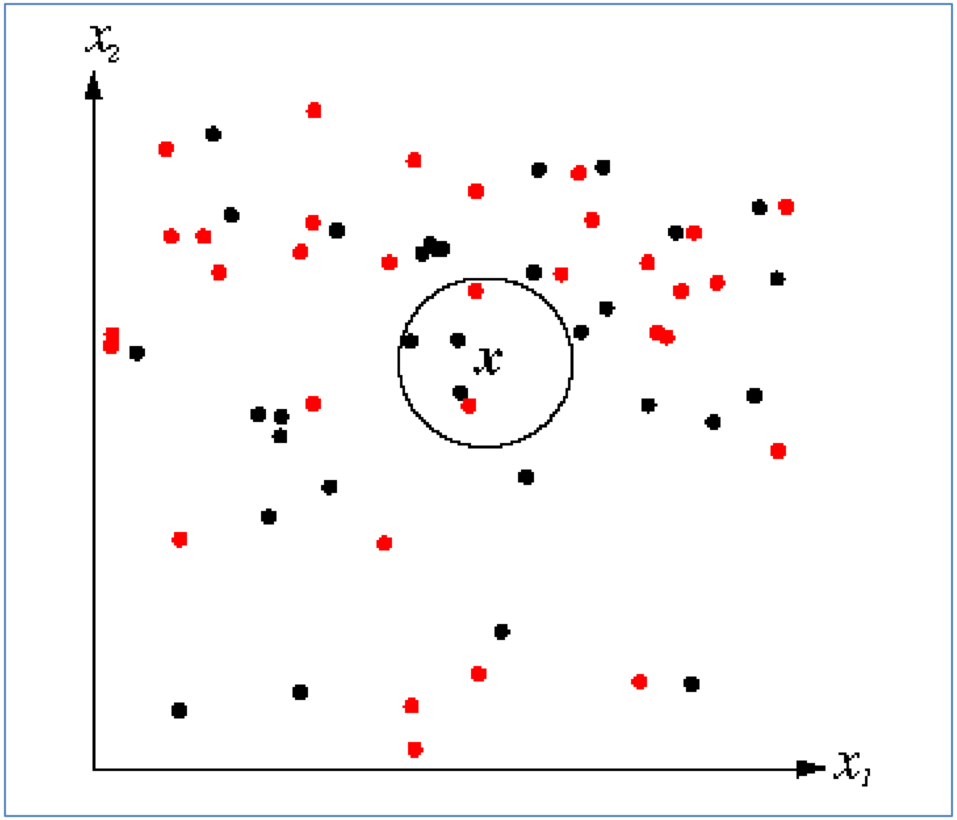
How to Measure “nearby”?

The most popular distance measure is
Euclidean distance

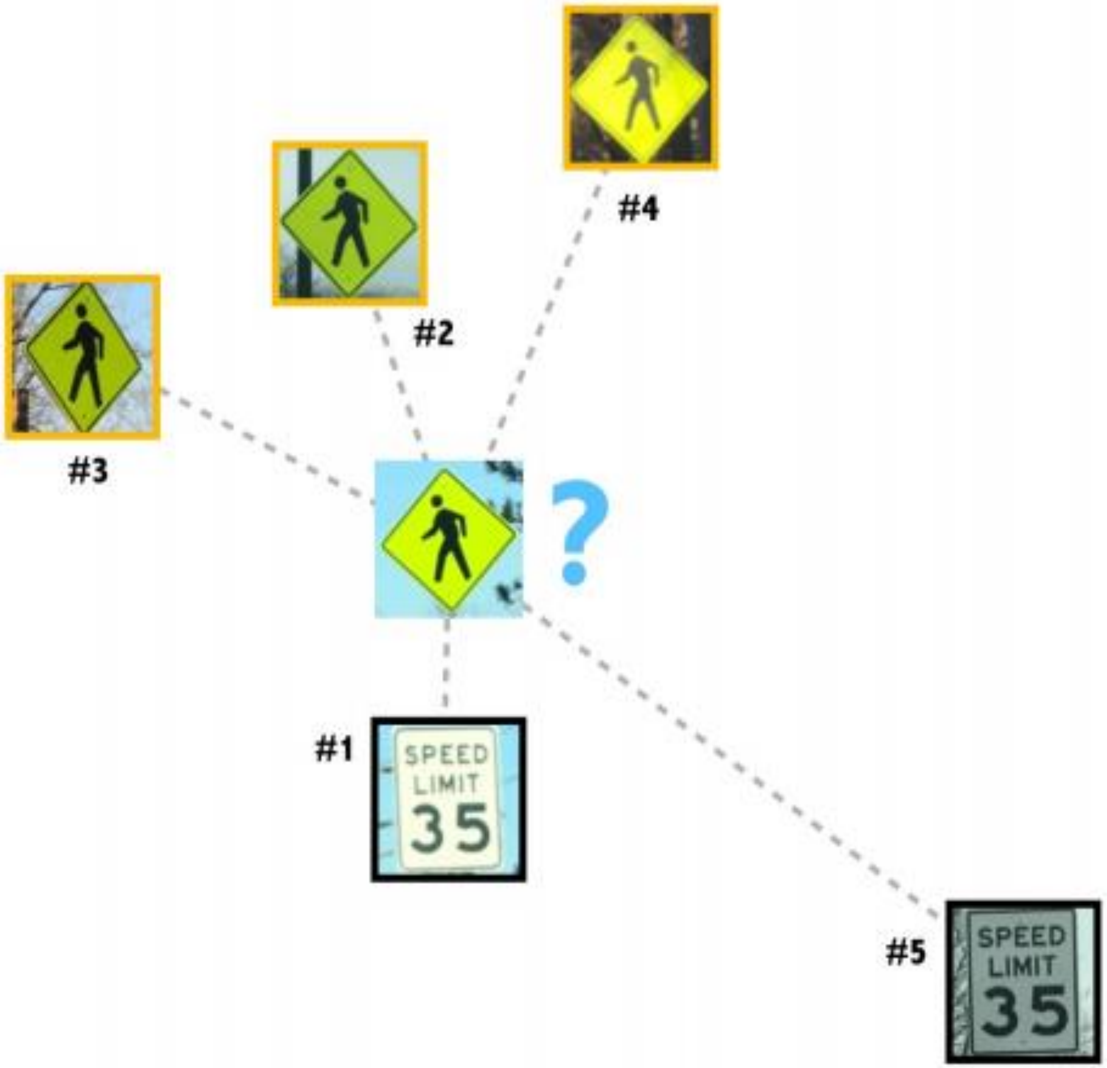
$$\sqrt{(x_1 - u_1)^2 + (x_2 - u_2)^2 + \cdots + \left(x_p - u_p\right)^2}$$

Illustration of the kNN Rule

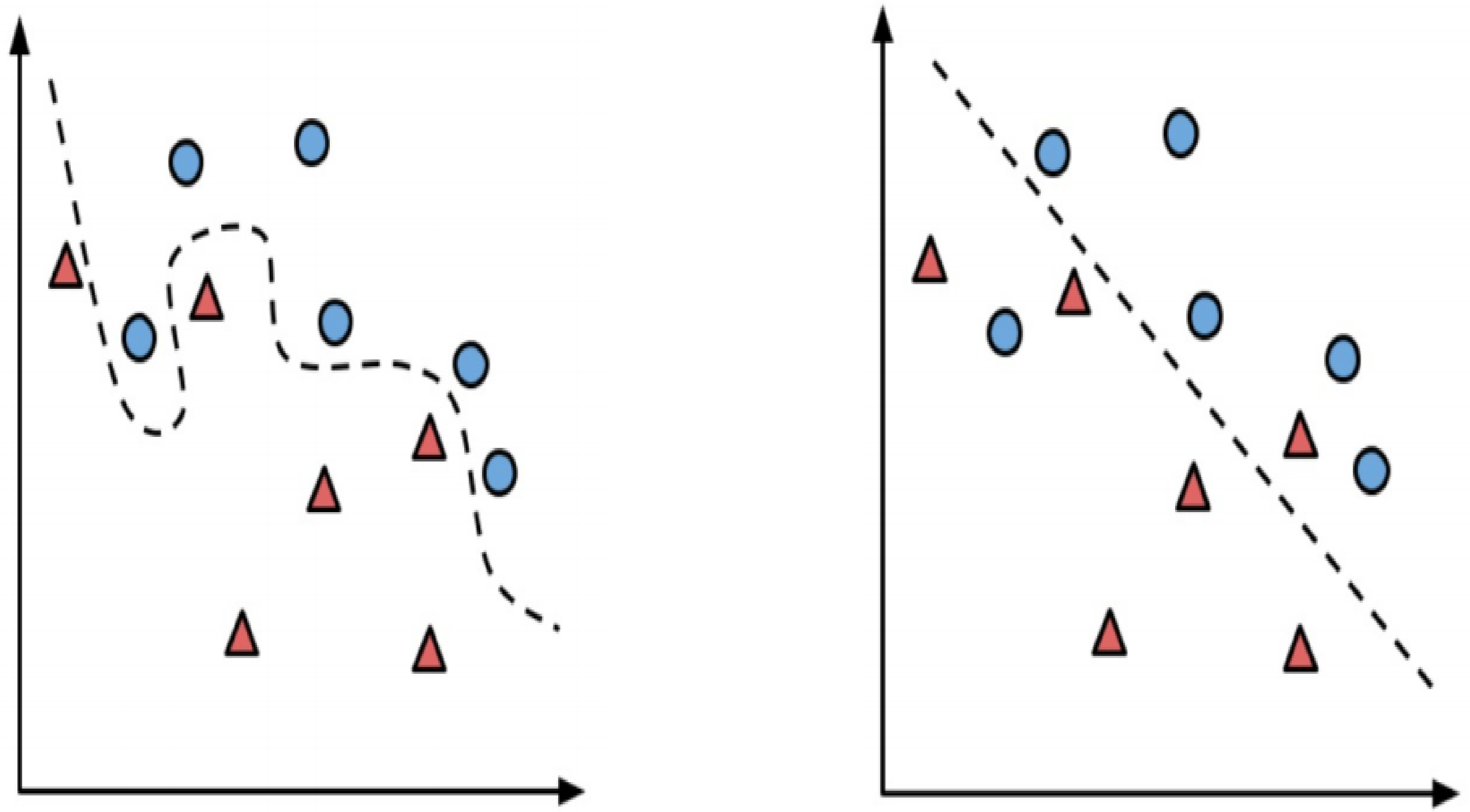
- k=5



Choosing 'k' neighbors



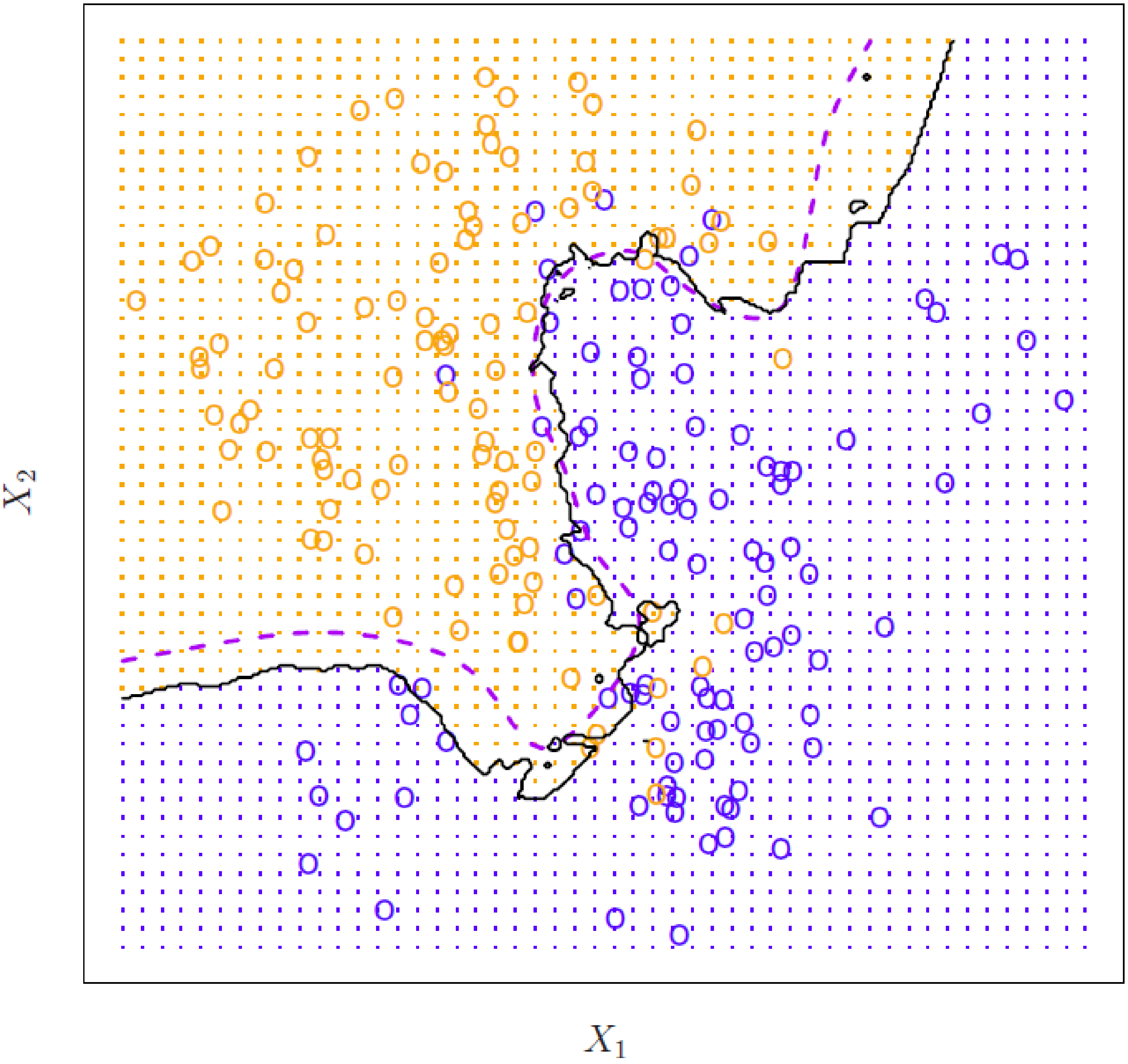
Bigger 'k' is not always better



Smaller k

Larger k

KNN: K=10



Choosing *k*

- *K* is the number of nearby neighbors to be used to classify the new record
  - *k*=1 means use the single nearest record
  - *k*=5 means use the 5 nearest records

- Typically choose that value of *k* which has lowest error rate in validation data

Low *k* vs. High *k*

- Low values of *k* (1, 3 …) capture local structure in data (but also noise)
- High values of *k* provide more smoothing, less noise, but may miss local structure

- Note: the extreme case of *k = n* (i.e. the entire data set) is the same thing as “naïve rule” (classify all records according to majority class)

Using K-NN for Prediction (for Numerical Outcome)

- Instead of “majority vote determines class” use average of response values

- May be a weighted average, weight decreasing with distance

kNN assumes numeric data

rectangle = 1

diamond = 0

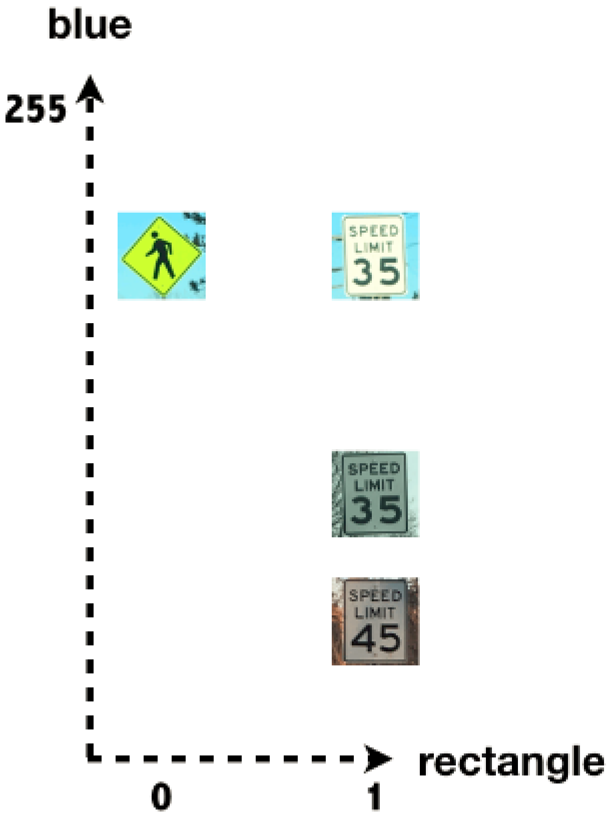
rectangle = 0

diamond = 1

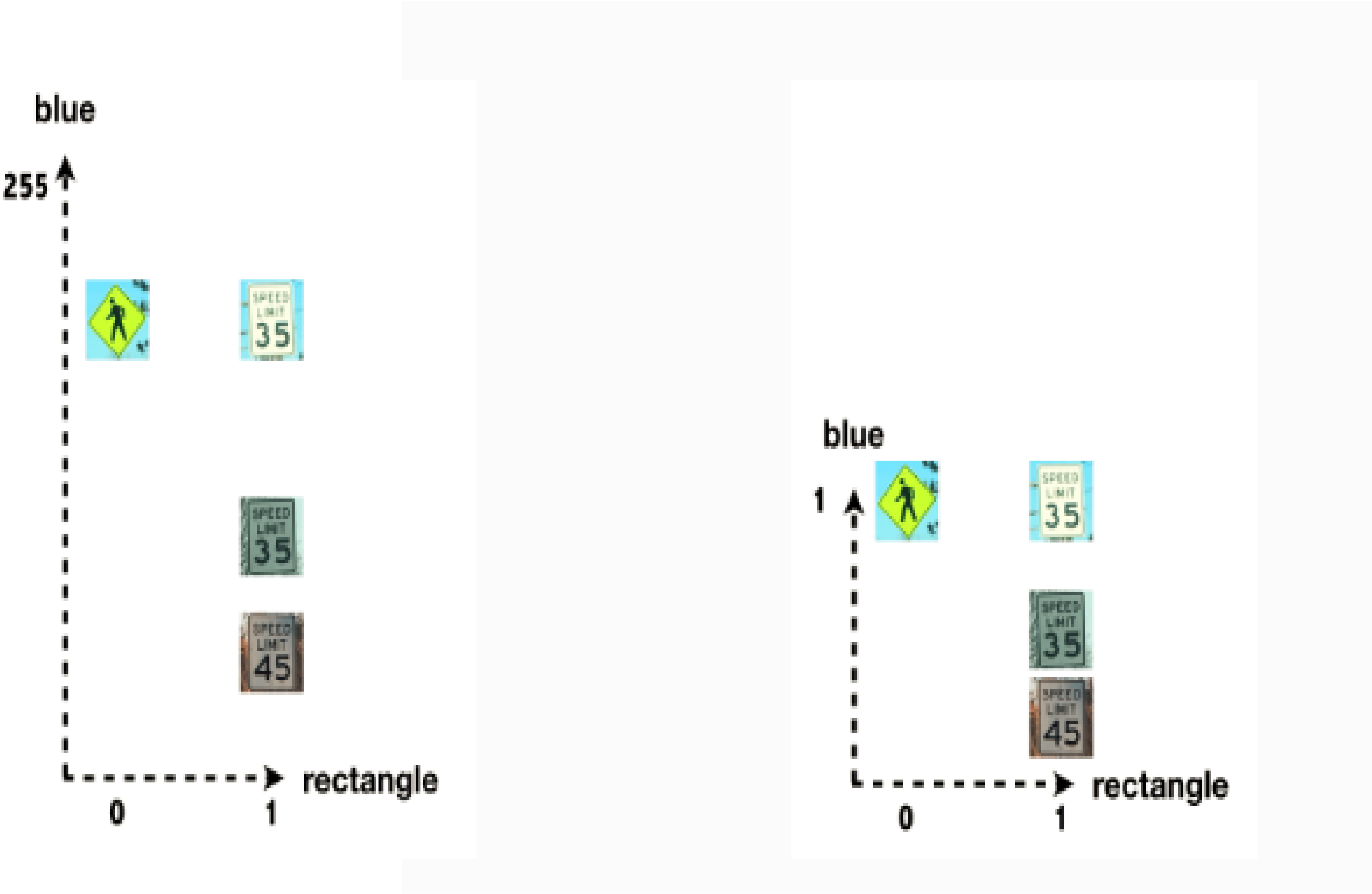
rectangle = 0

diamond = 0

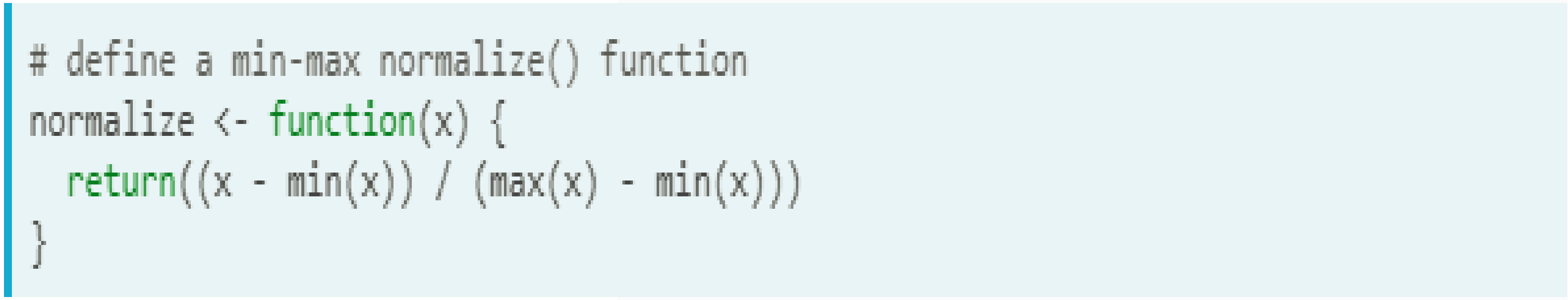
kNN benefits from normalized data



kNN benefits from normalized data



Normalizing data in R



```
# define a min-max normalize() function
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}
```

Applying nearest neighbors in R

```
library(class)
pred <- knn(training_data, testing_data, training_labels)
```

Advantages

- Simple
- No assumptions required about Normal distribution, etc.
- Effective at capturing complex interactions among variables without having to define a statistical model

Shortcomings

- Required size of training set increases exponentially with # of predictors, *p*
  - This is because expected distance to nearest neighbor increases with *p* (with large vector of predictors, all records end up “far away” from each other)
- In a large training set, it takes a long time to find distances to all the neighbors and then identify the nearest one(s)
- These constitute “curse of dimensionality”