k-Nearest Neighbors

Krishnakant Saboo

20th June 2020

Leaves

Maple

Cactus

Mango

Which plant does this leaf belong to?

Concepts from the previous example

- Training data
- Features
- Dissimilarity score
- Decision rule

Maple

Cactus

Mango

Training data

AI orange seller

It is orange season and Ms. Orange Seller is getting hundreds of oranges everyday from her suppliers. The suppliers give her a mixed bag of two types of oranges – clementine and navel. She wants to segregate the oranges using a machine and decides to employ techniques from the new ML course that she did.

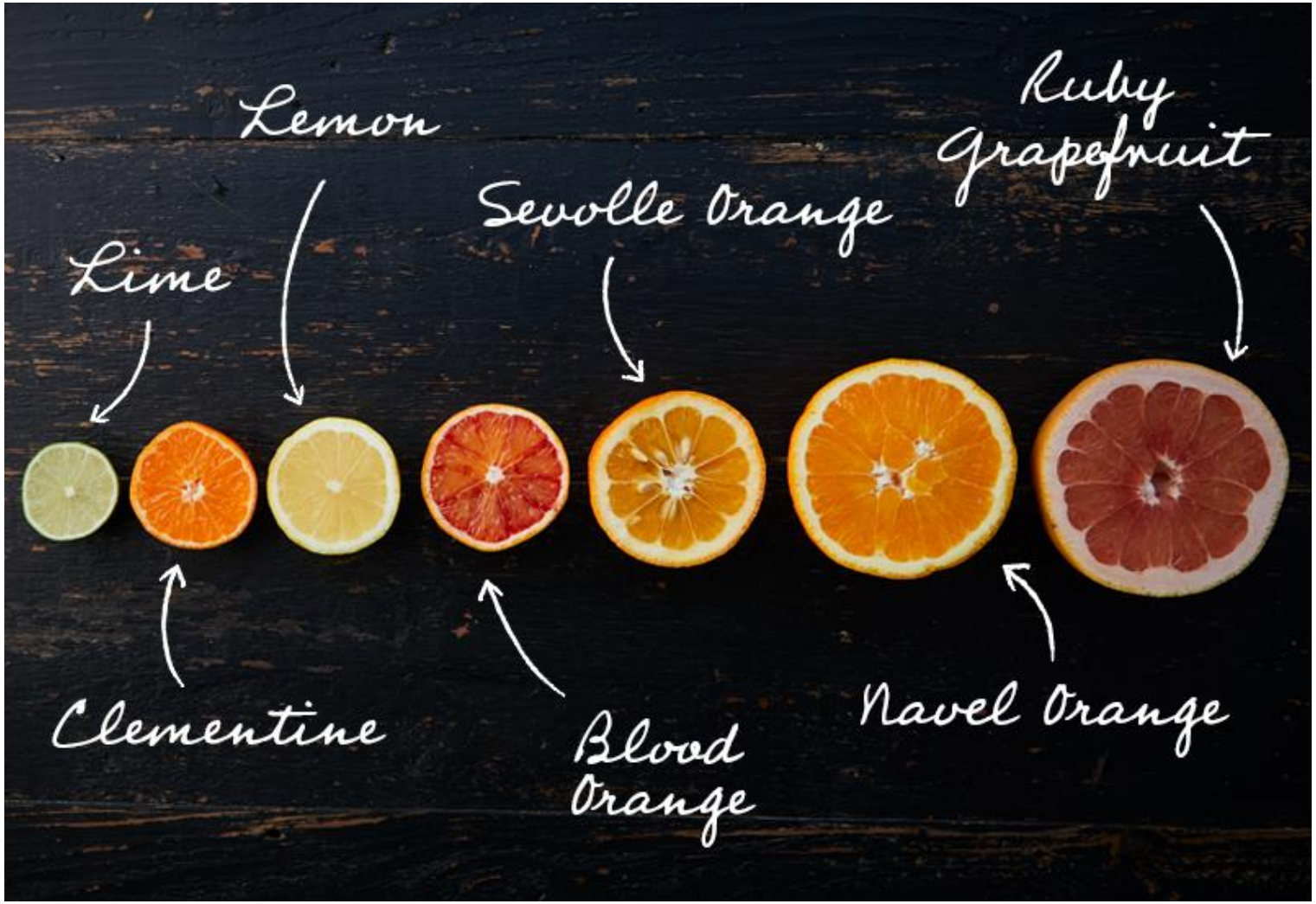
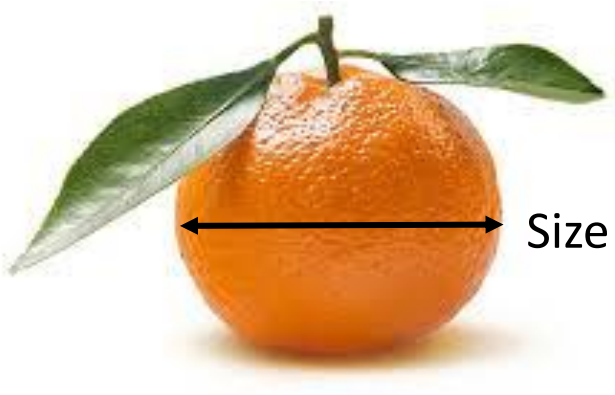
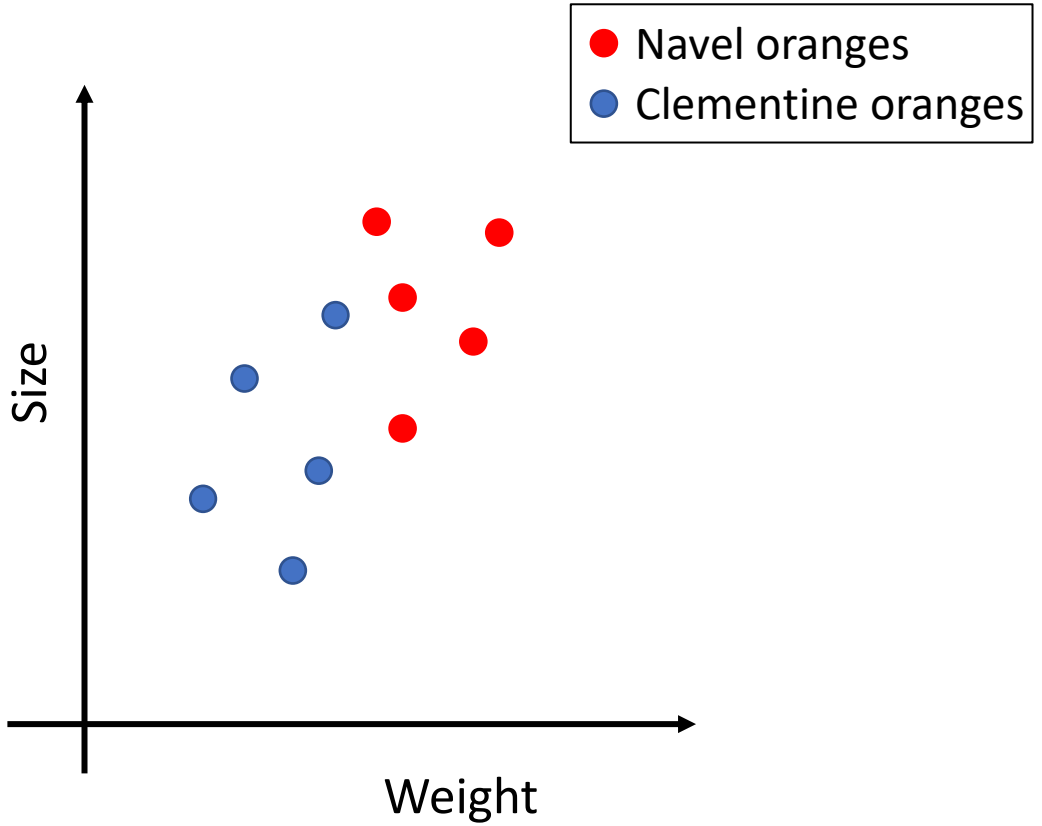


Image source: https://www.farmdrop.com/blog/your-guide-to-the-best-winter-citrus/

AI orange seller

She decides to use size and weight of the orange to represent each orange

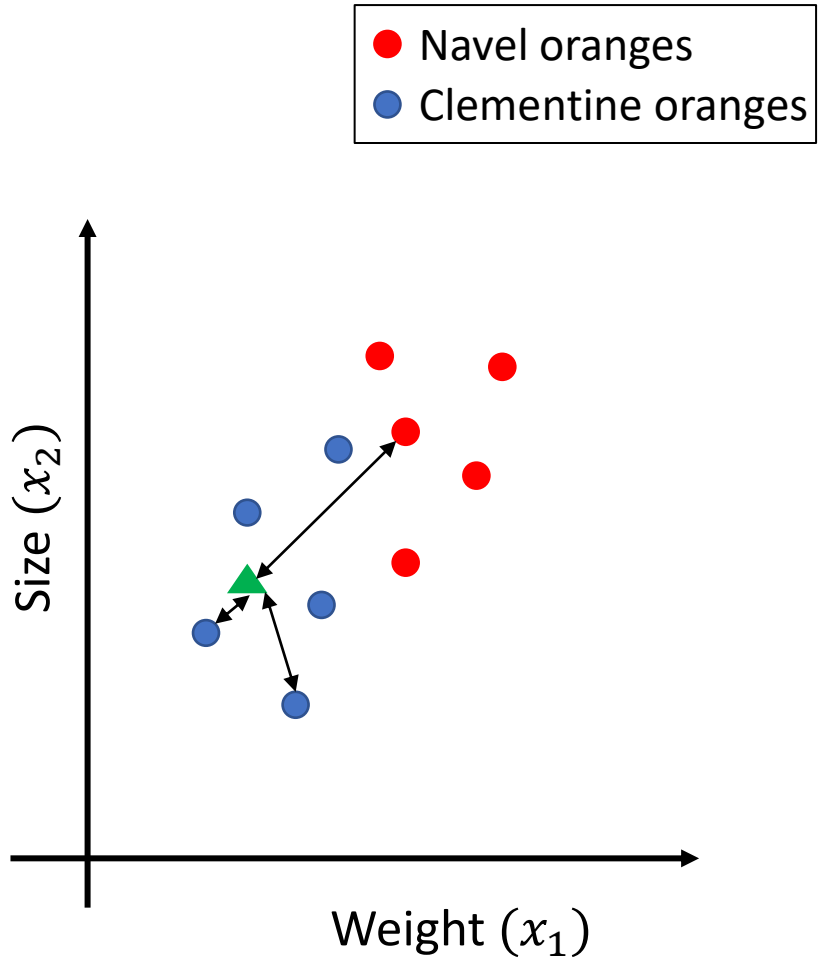




Nearest Neighbor

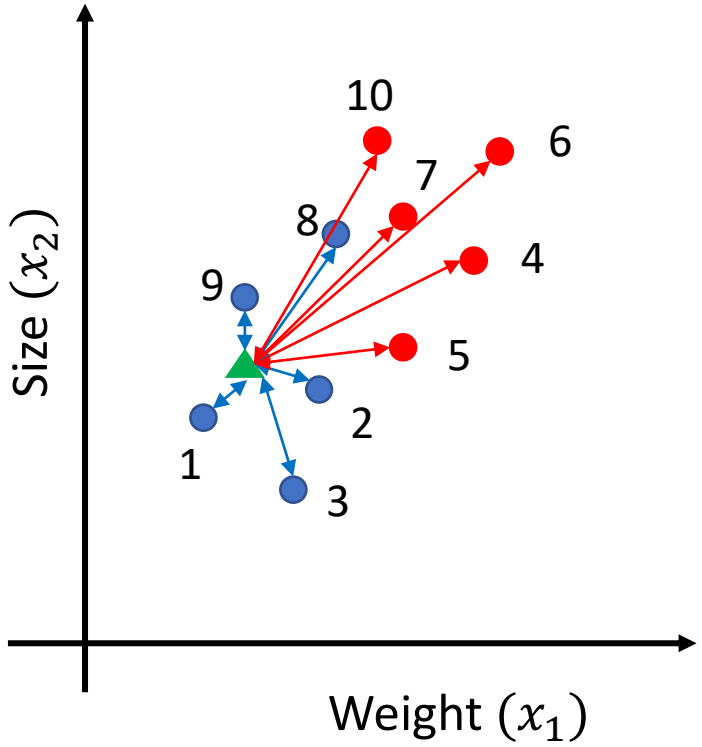
- Idea: If something is similar in one aspect, it is likely to be similar in the other aspect too
- For a new orange (sample), assign it to the type (class) of the orange (sample) that it is most similar to
- Similarity is measured with distance:

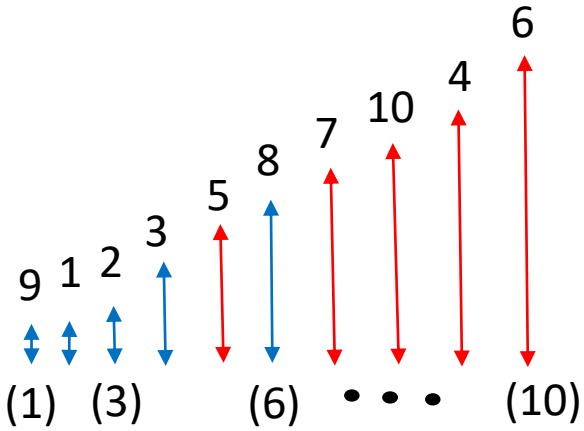
$$d\left(x^{(i)}, x^{(j)}\right) = \sqrt{\left(x_1^{(i)} - x_1^{(j)}\right)^2 + \left(x_2^{(i)} - x_2^{(j)}\right)^2}$$



Nearest Neighbor algorithm

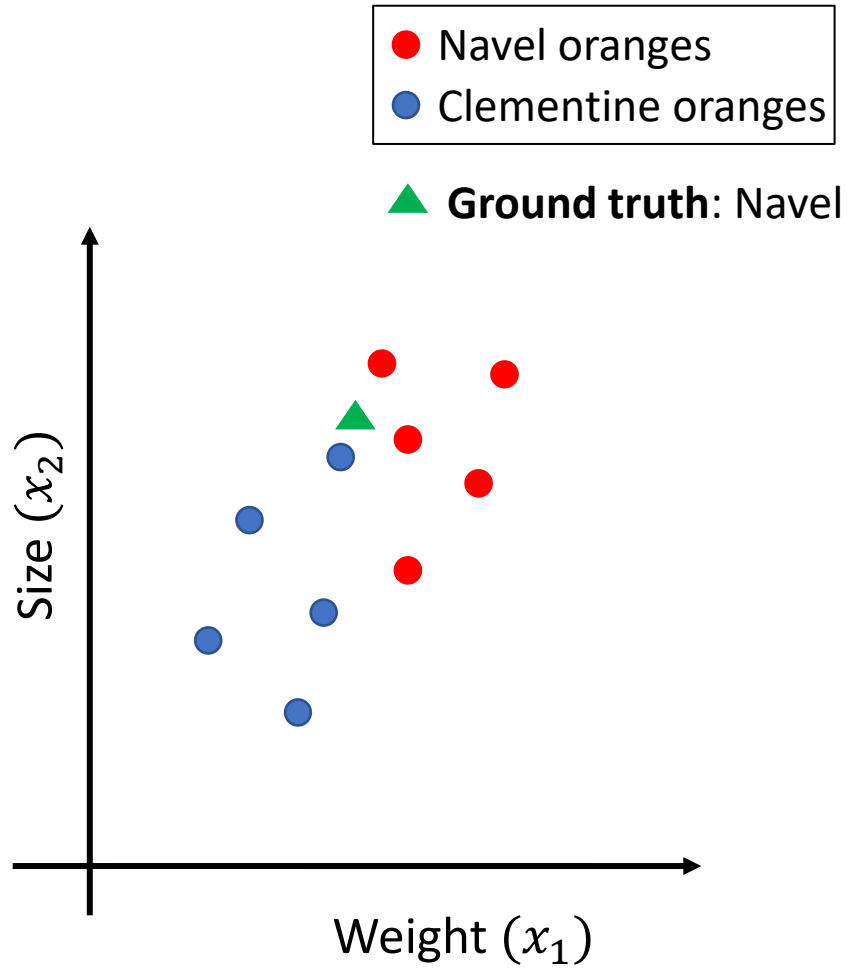
- Training data $(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)$
  - X_i are features and Y_i is label of sample i
- For a new sample with feature X_{new}, compute dissimilarity score $d(X_{new}, X_i)$ for every sample
- Sort the training data samples based on $d(X_{new}, X_i)$ as $(X_{(1)}, Y_{(1)}), (X_{(2)}, Y_{(2)}), \ldots, (X_{(N)}, Y_{(N)})$ such that $d(X_{new}, X_{(1)}) \leq d(X_{new}, X_{(2)}) \leq \cdots \leq d(X_{new}, X_{(n)})$
- Assign label for new sample $Y_{new} = Y_{(1)}$



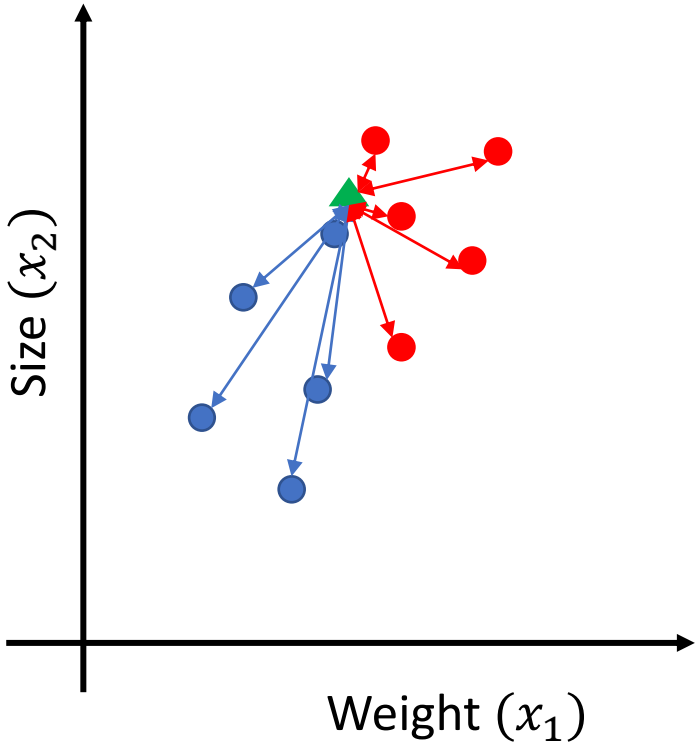


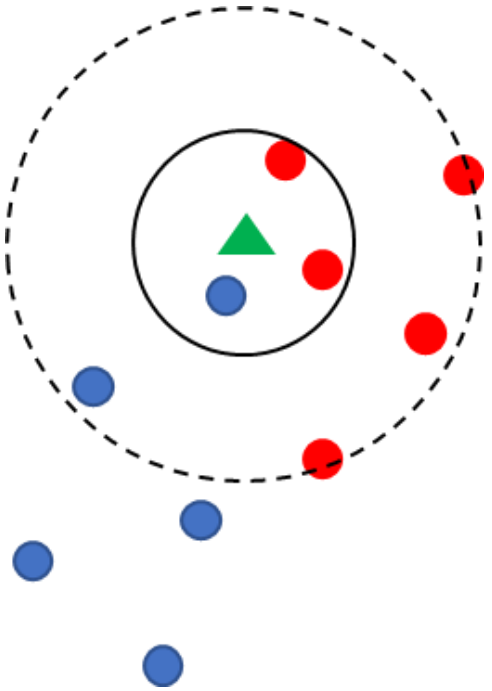
Affects of noise

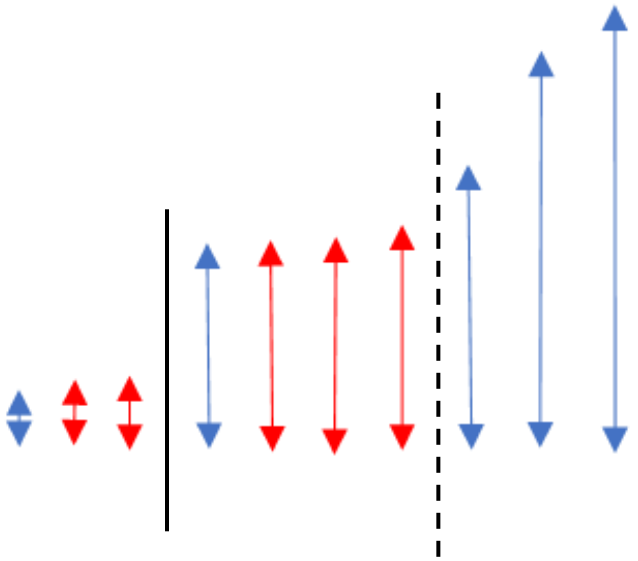
- Which type is the new orange?
- Nearest neighbor method clementine (closest point is blue) but Ms. Seller knows oranges well and is sure that the new one is a navel orange
- All though the size and weight are high and orange is mostly surrounded by navel oranges (red), the clementine orange (blue) sample is the closest
- Nearest neighbor method is not robust
  - Easily affected by noisy samples, outliers



What if we considered multiple neighbors?







- Assign label as the majority class of the top k neighbors

K-Nearest Neighbor algorithm

- Training data $(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)$
  - X_i are features and Y_i is label of sample i
- For a new sample with feature X_{new}, compute dissimilarity score $d(X_{new}, X_i)$ for every sample
- Sort the training data samples based on $d(X_{new}, X_i)$ as $\left(X_{(1)}, Y_{(1)}\right), \left(X_{(2)}, Y_{(2)}\right), \ldots, (X_{(N)}, Y_{(N)})$ such that $d\left(X_{new}, X_{(1)}\right) \leq d\left(X_{new}, X_{(2)}\right) \leq \cdots \leq d(X_{new}, X_{(n)})$
- Assign label for new sample $Y_{new} = majority\{Y_{(1)}, \ldots, Y_{(k)}\}$

How to choose k?

- k should be an odd number
- The boundaries become smoother as k increases
  - Better able to handle noise
- How much to increase k?
  - What happens when k=N?

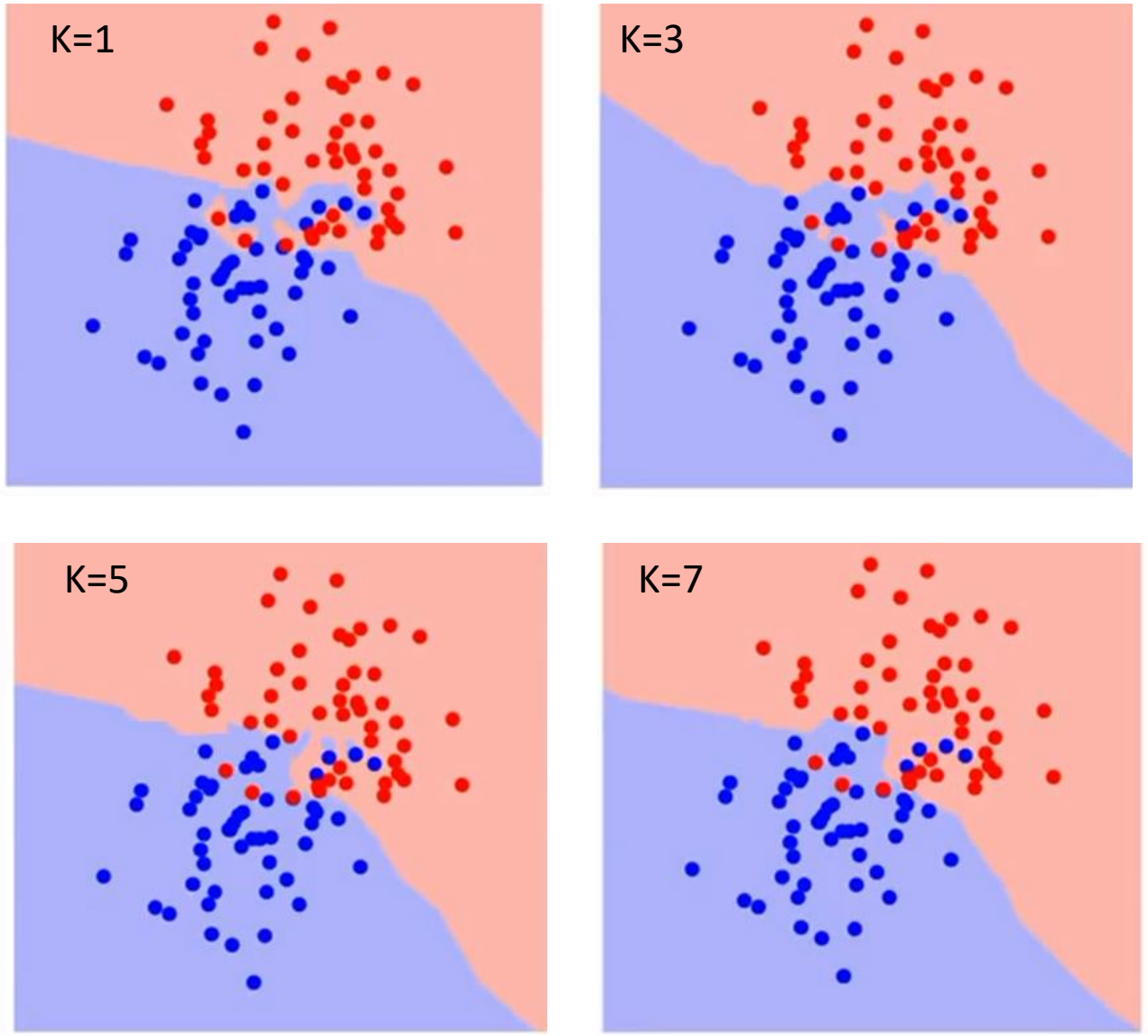
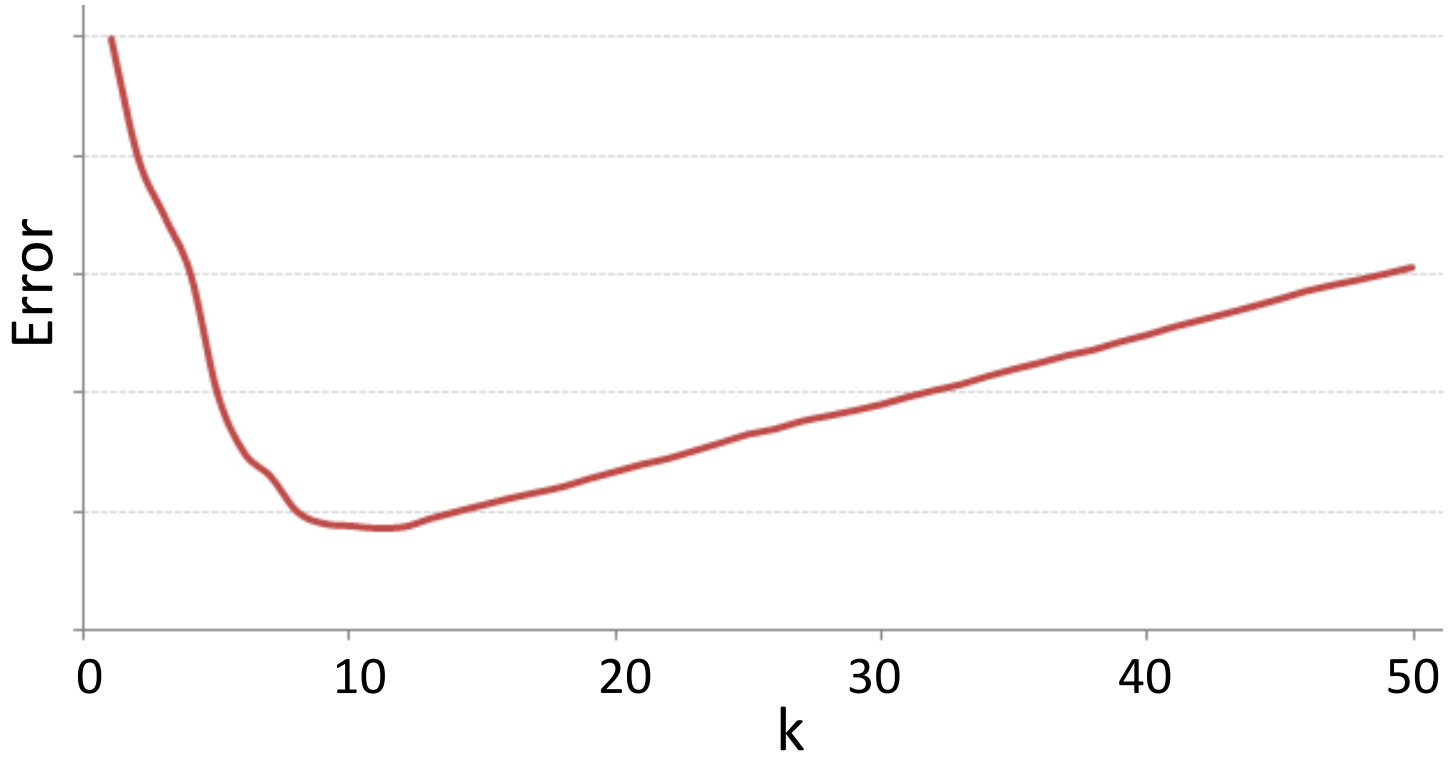


Image source: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

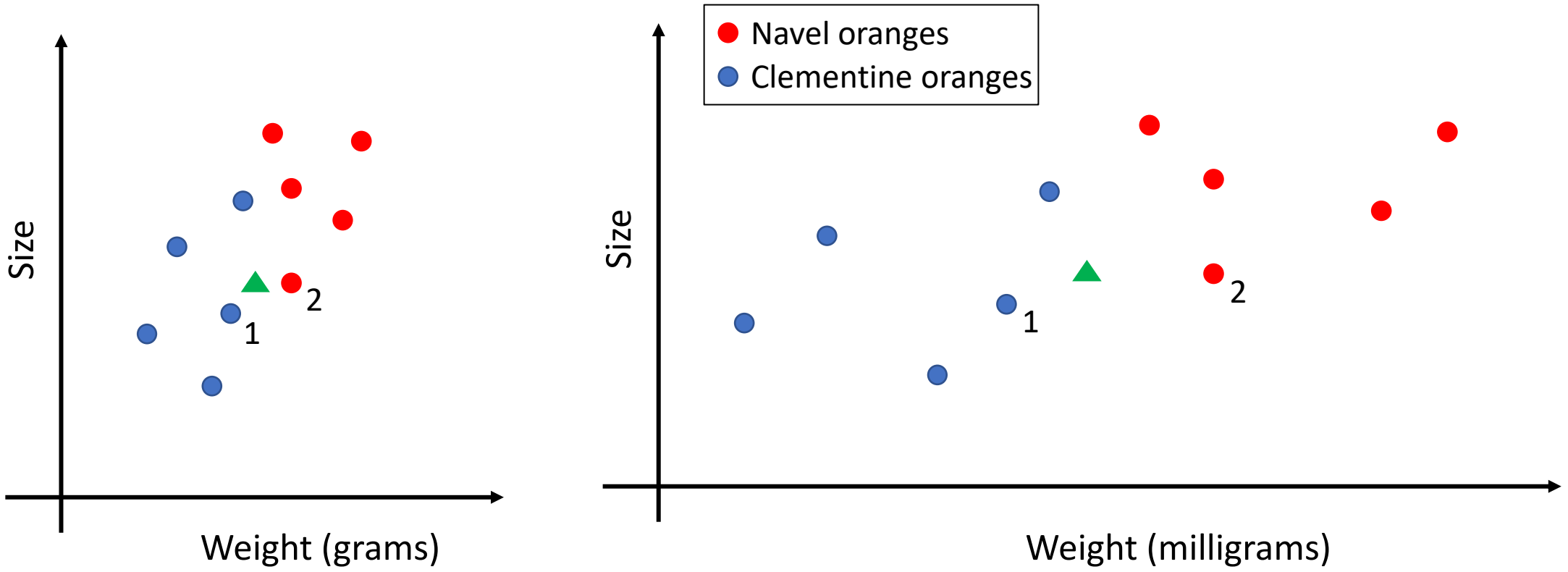
Performance error as k increases



- As k increases, performance improves until a certain point after which it starts degrading
- Optimal choice for k in this case seems to be 9 or 11

Image source: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Should the weight be measured in g or mg?



- New orange (green) is equidistant from oranges 1 and 2 when weight is measured in g but closer to orange 1 when weight is measured in mg
- Scale of the features affects distance and therefore k-NN performance

K-Nearest Neighbors

Pros

- Algorithm is simple and intuitive
- Non-parametric and allows for non-linear decision boundaries

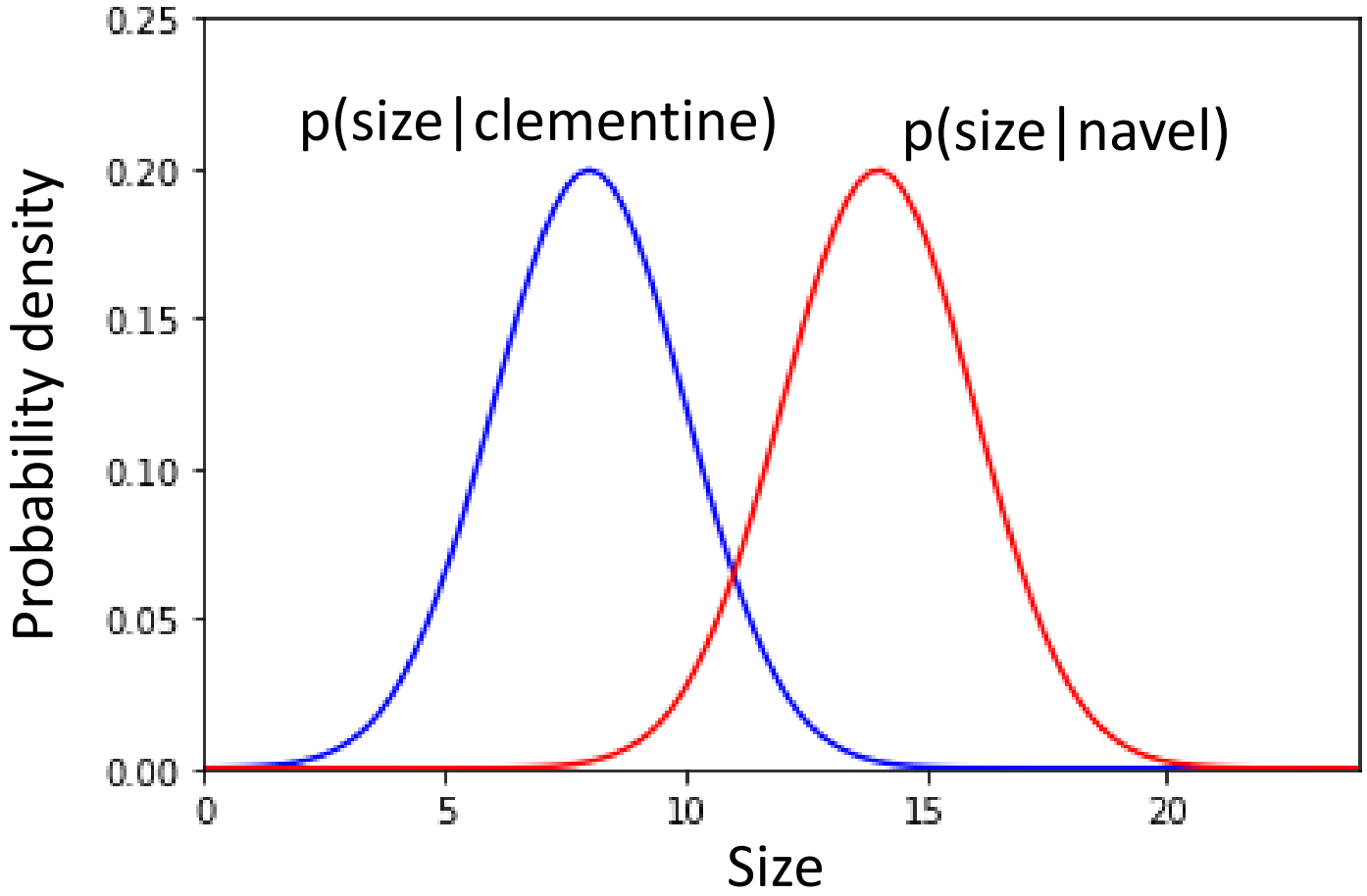
Cons

- Computationally expensive; doesn't scale to large datasets
- Performance severely degrades in the presence of noisy or irrelevant features
- Performance is affected by the scale of the features
- Majority voting can become a drawback if number of samples are larger in one class
- Distances are less meaningful in high-dimensions

Source: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

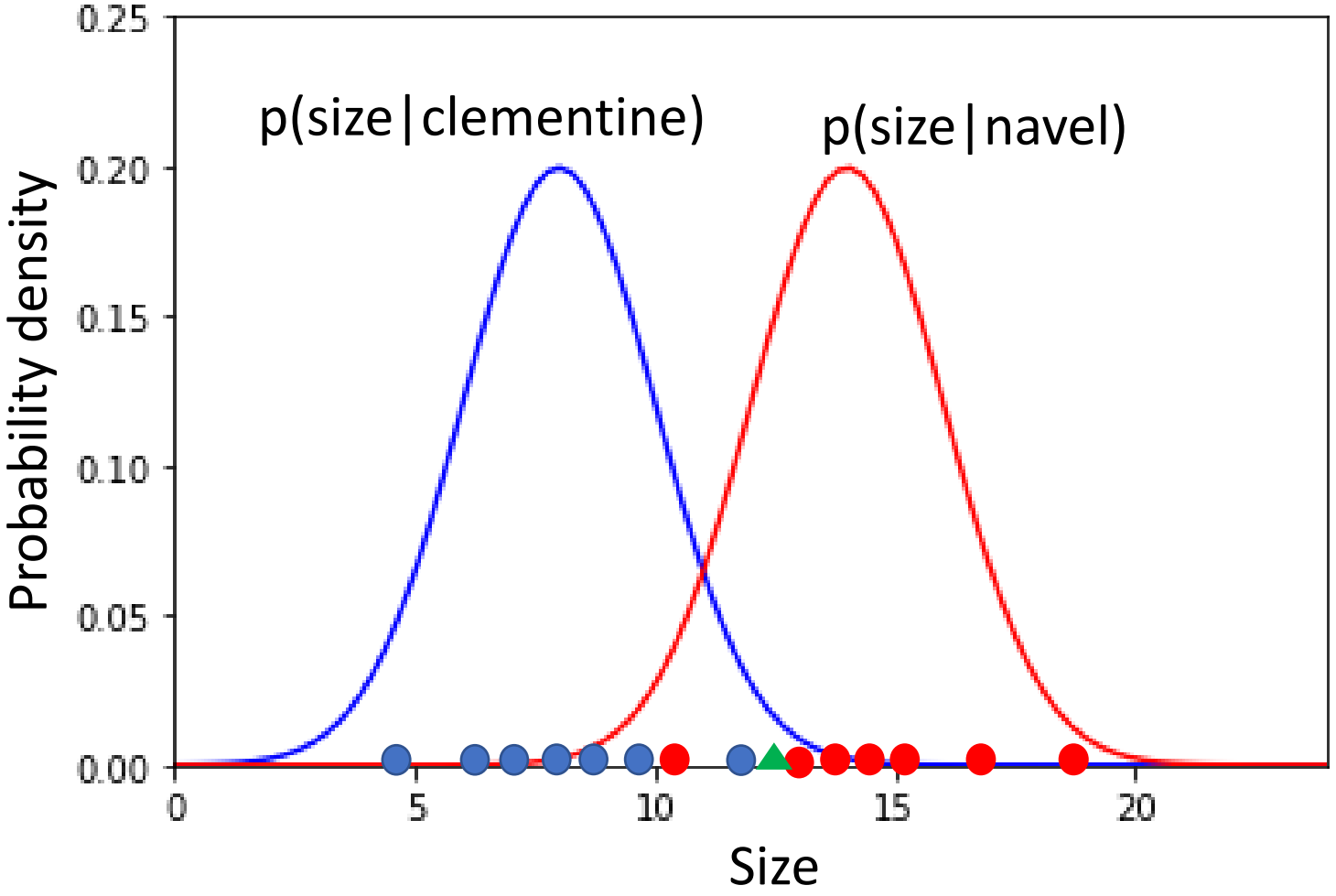
AI orange seller revisited…

Ms. Seller decides to determine the type of the orange just based on its size. She read some research papers on orange sizes and found that the size are Gaussian distributed as shown below.

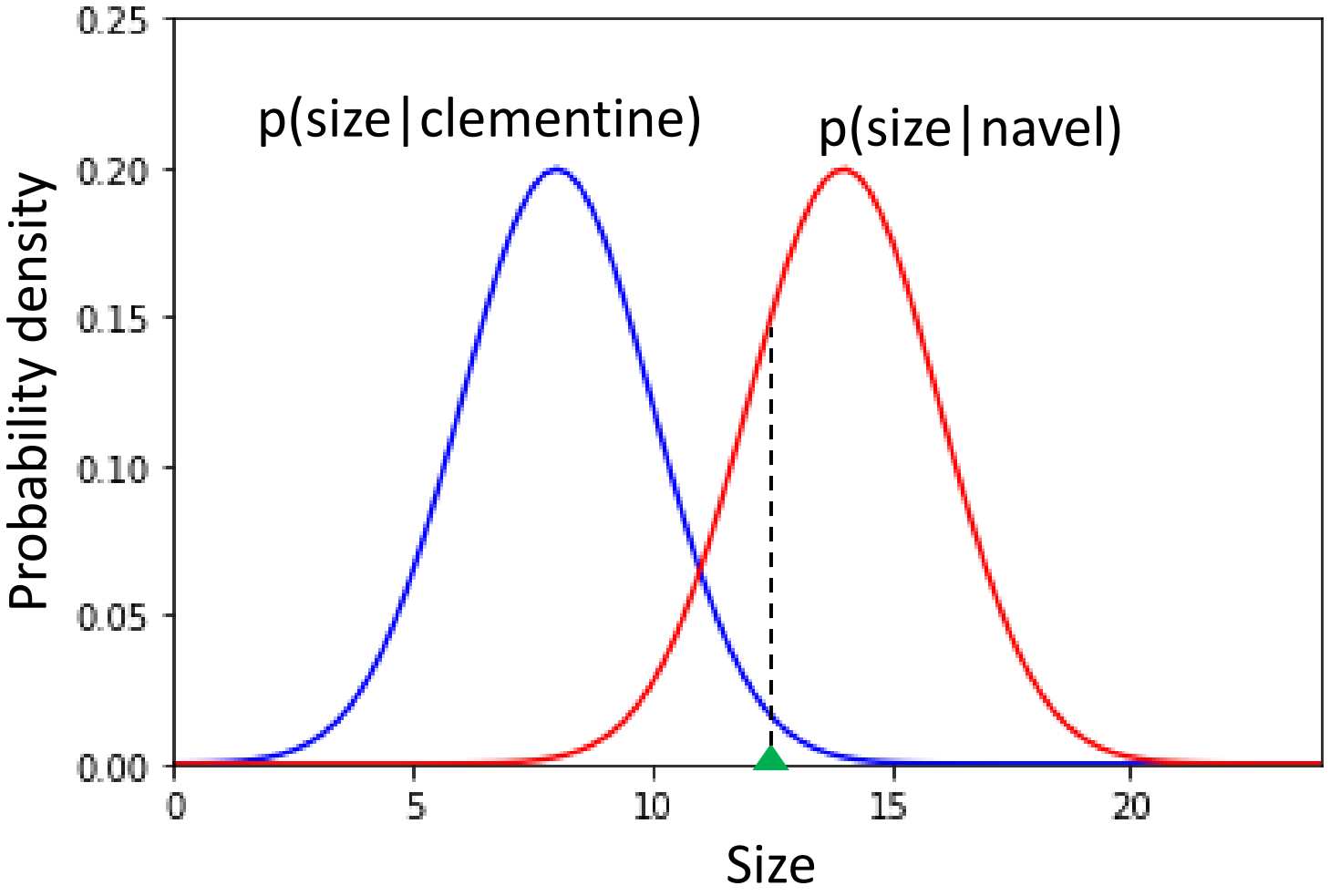


AI orange seller revisited…

What does kNN really mean in the context of these Gaussian distributions?



AI orange seller revisited…



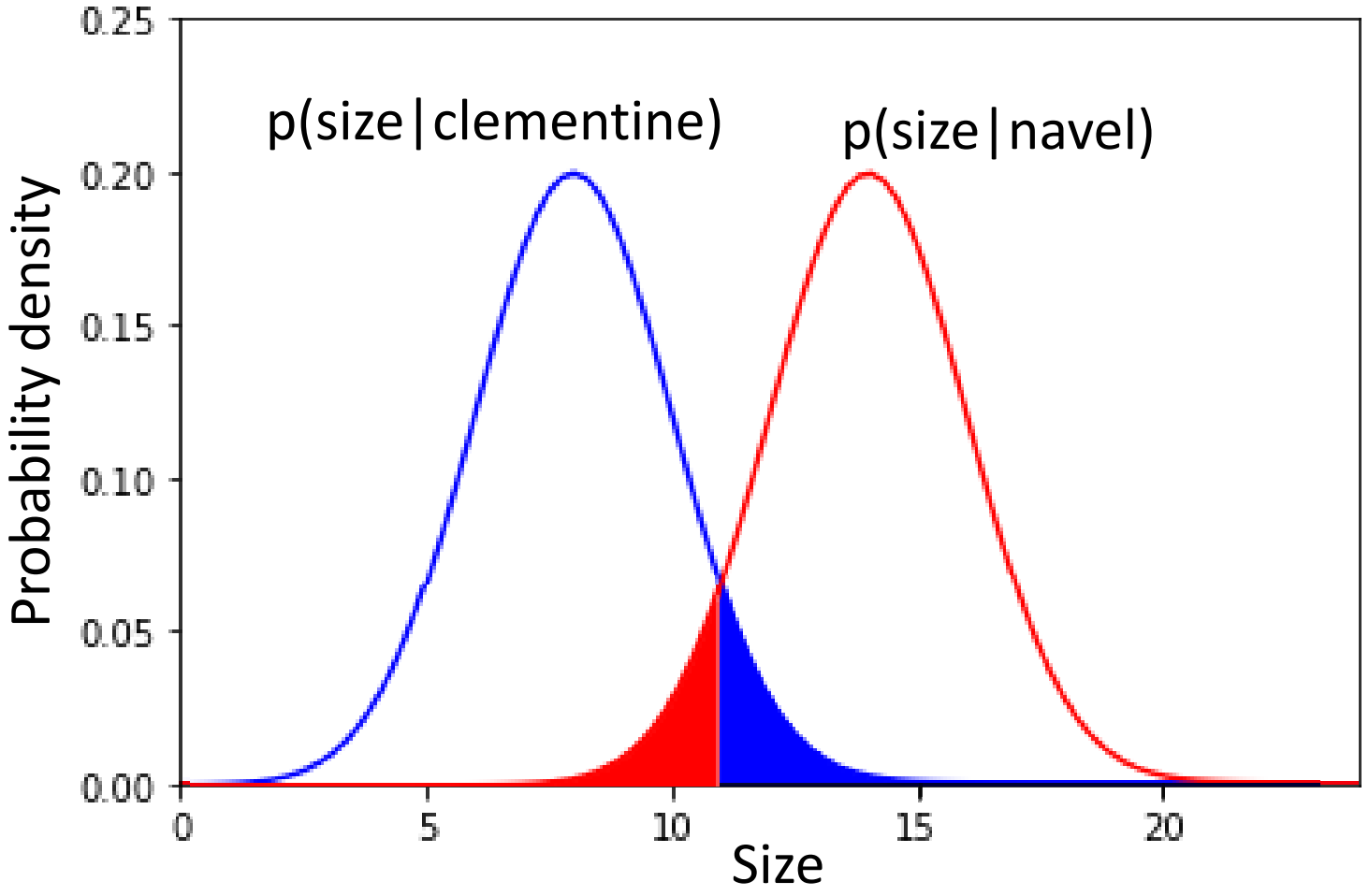
Size=13

p(size=13|clementine) < p(size=13|navel)

So the new orange is classified as a navel orange

An orange of any given size will be assigned to the class which has a higher probability density at that size

Limit on classification performance



An orange of any given size will be assigned to the class which has a higher probability density at that size

Area shaded in blue represents clementine oranges that will be misclassified as navel oranges

Area shaded in red represents navel oranges that will be misclassified as clementine oranges

Fundamental limit on the classification performance depends on the overlap between the distributions.

Evaluation of classifier performance

- Specify classifier output for each sample separately
  - Very cumbersome
- Need summary statistics!
- Denote
  - Clementine orng: Negative class
  - Navel orng: Positive class

Ground truth

Prediction	Positive (Navel)	Negative (Clementine)
Positive (Navel)	True positive (TP)	False Positive (FP)
Negative (Clementine)	False negative (FN)	True negative (TN)

Confusion matrix

Confusion matrix examples

Ground truth

Prediction

	Positive (Navel)	Negative (Clementine)
Positive (Navel)	25 (TP)	10 (FP)
Negative (Clementine)	0 (FN)	15 (TN)

Classifier 1

Ground truth

Prediction

	Positive (Navel)	Negative (Clementine)
Positive (Navel)	20 (TP)	0 (FP)
Negative (Clementine)	5 (FN)	25 (TN)

Classifier 2

Ms. Seller built two different classification models and tested it on 50 samples. Here is the confusion matrix for the two classifiers.

- Given an orange, what is classifier 1 more likely to call it?
- Given an orange, what is classifier 2 more likely to call it?
- Which method is better?

Accuracy, Sensitivity, Specificity

Accuracy: $\frac{TP+TN}{Total\ samples}$

$$\frac{25+15}{25+10+0+15}=\frac{40}{50}=0.8$$

Sensitivity: $\frac{TP}{TP+FN}$

$$\frac{25}{25+0}=1$$

Specificity: $\frac{TN}{TN+FP}$

$$\frac{15}{15+10}=\frac{15}{25}=0.6$$

Ground truth

Prediction	Positive (Navel)	Negative (Clementine)
Positive (Navel)	25 (TP)	10 (FP)
Negative (Clementine)	0 (FN)	15 (TN)

Classifier 1

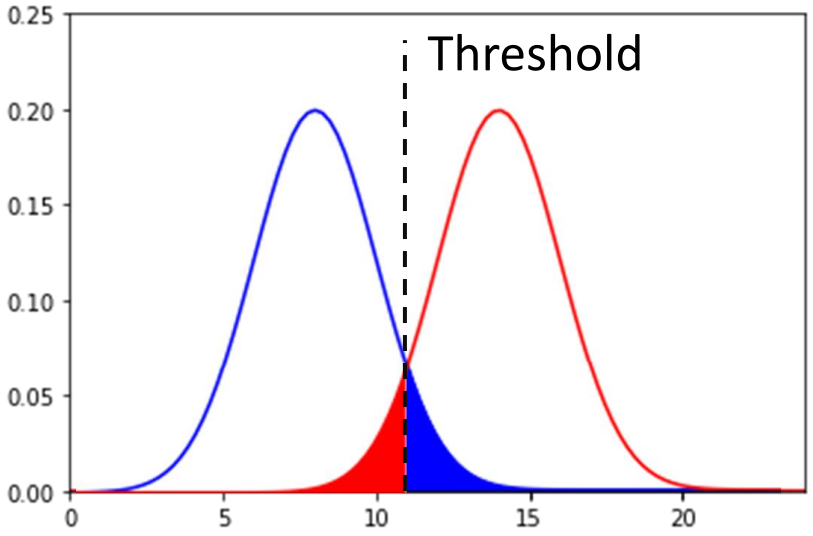
How many oranges were classified correctly?

How many navel oranges were correctly identified?

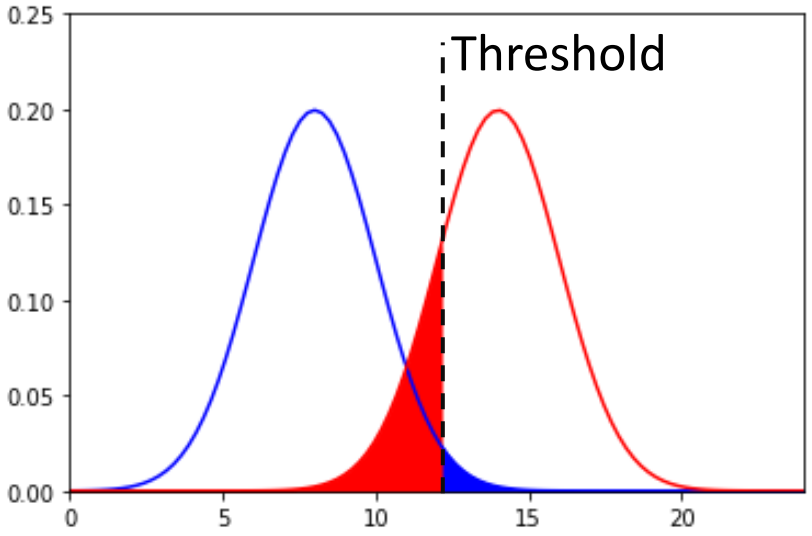
How many clementine oranges were correctly identified?

Tradeoff between sensitivity and specificity

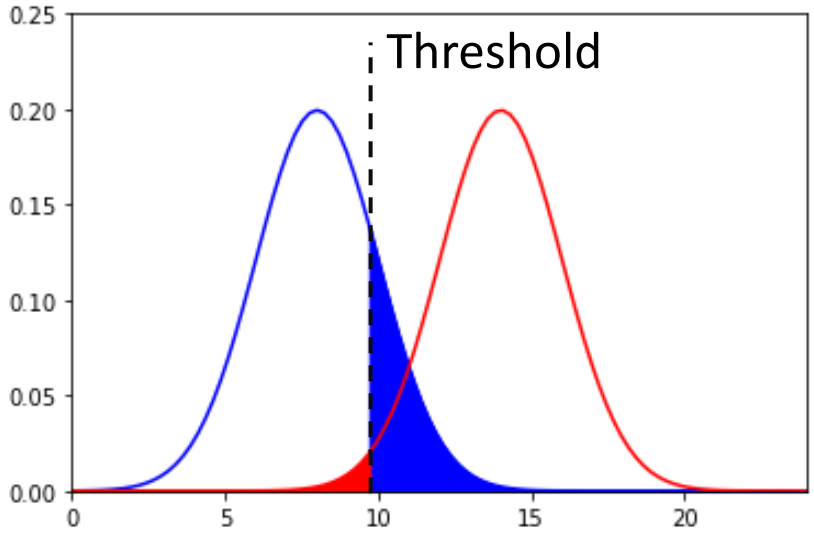
Red: Navel; Blue: Clementine



No preference



Clementines preferred

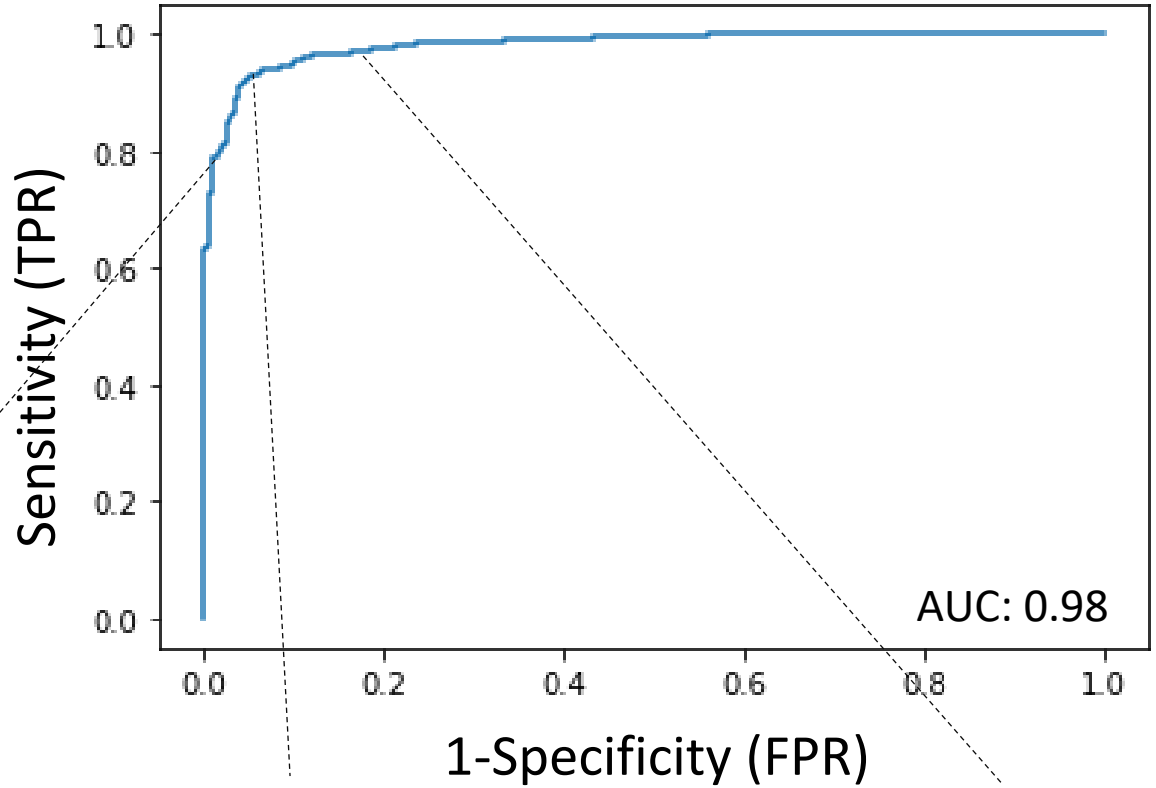


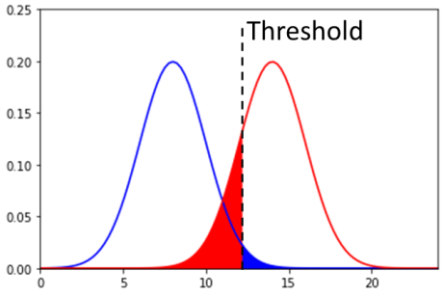
Navel preferred

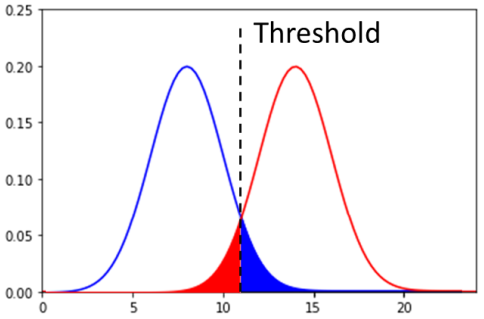
- Ms. Seller observes that if sensitivity increases, specificity decreases
- What threshold to choose? Depends on the customer demand (application specific)
- Is there a way to make classifier evaluation independent of the threshold?

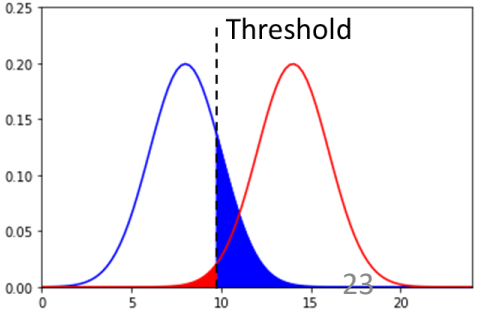
Receiver operating characteristics

- Receiver operating characteristics: Plot sensitivity and 1-specificity by varying the threshold
- Area under the curve
  - Between [0, 1]; higher the better
  - Chance level is 0.5; so typically larger than that

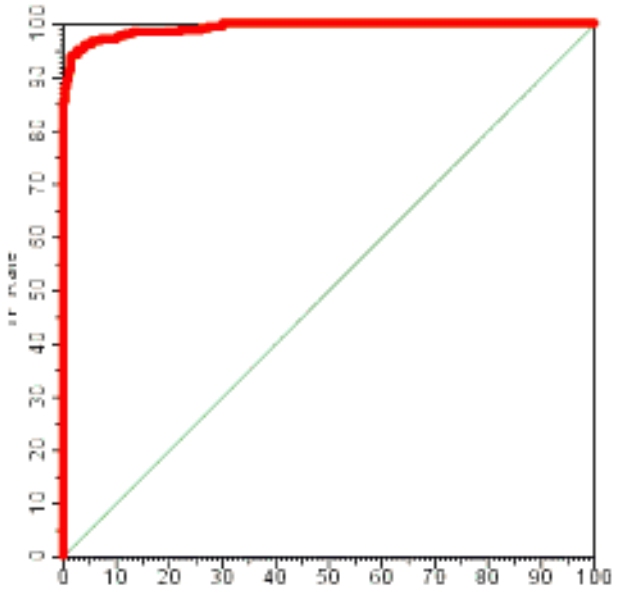




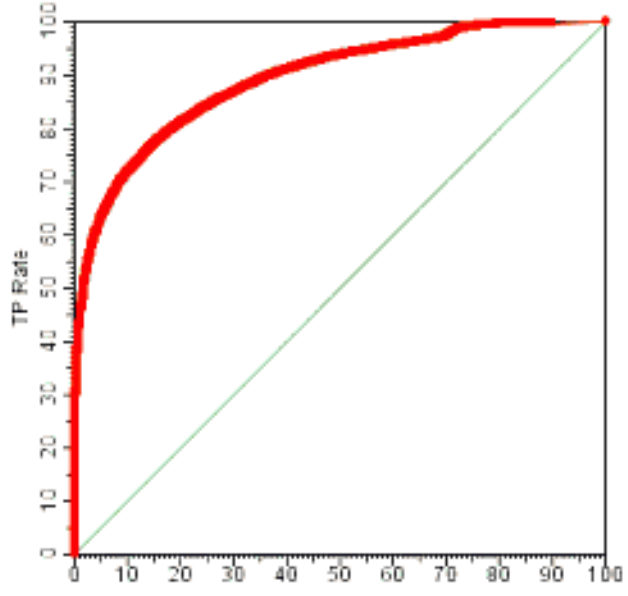




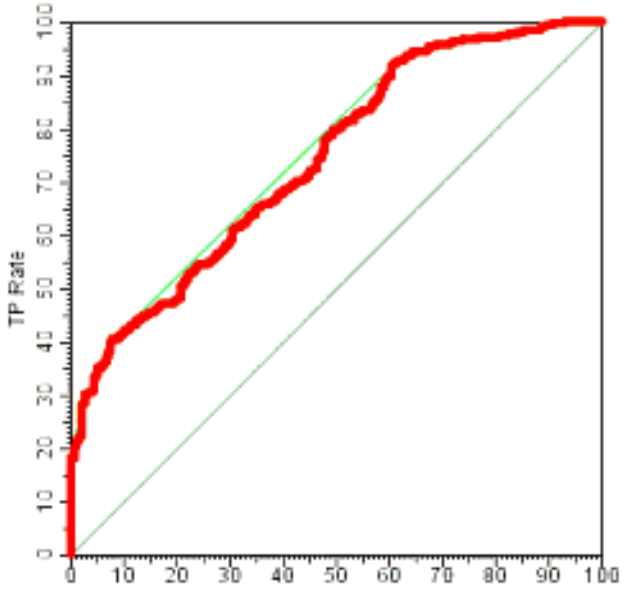
Receiver operating characteristics



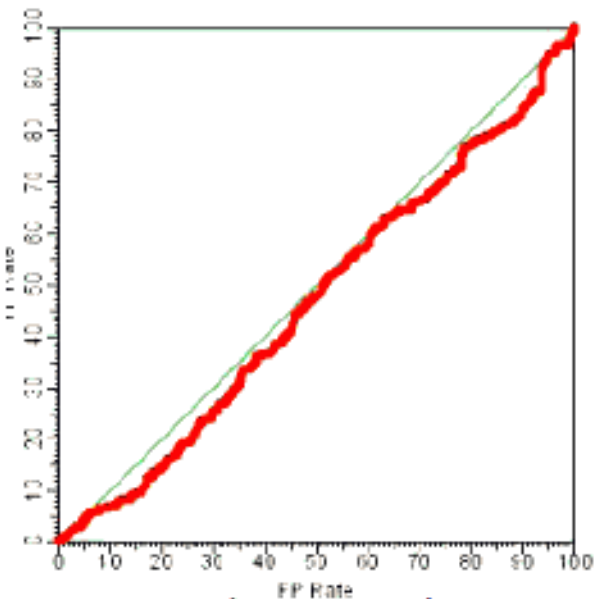
good separation



reasonable



poor separation



random separation

Image source: http://mlwiki.org/index.php/ROC_Analysis

Questions?