11.08.2023

Statistical Methods in AI (CS7.403)

Lecture-4: Performance Measures, Benchmarking ; k-Nearest Neighbours

Ravi Kiran (ravi.kiran@iiit.ac.in)

https://ravika.github.io

@vikataravi

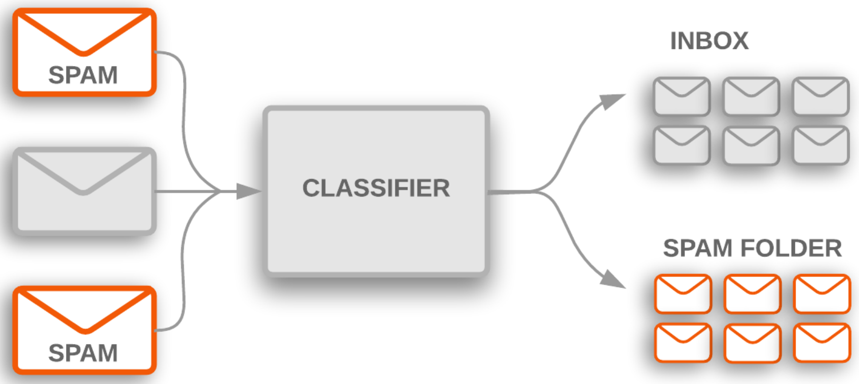
Center for Visual Information Technology (CVIT)

IIIT Hyderabad

Announcements

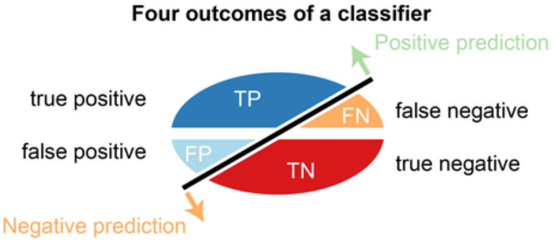
- For those who participated on Wednesday: Thanks !
- A1
  - Will be out today
  - Explanation session tomorrow in Tutorial
- Tutorial: Proof techniques

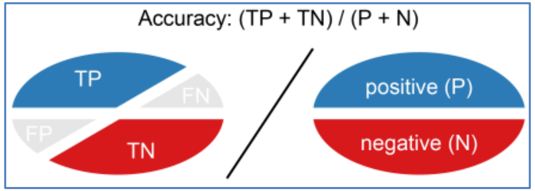
Binary Classification



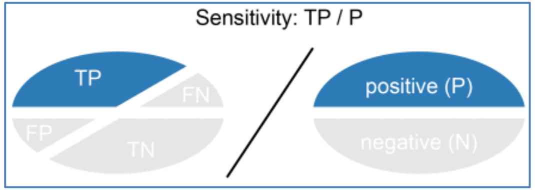
Accuracy vs Precision vs Recall

- Accuracy : Performance w.r.t both classes
- Recall : Performance w.r.t '+' class
- Precision : Reliability of predictions w.r.t '+' class

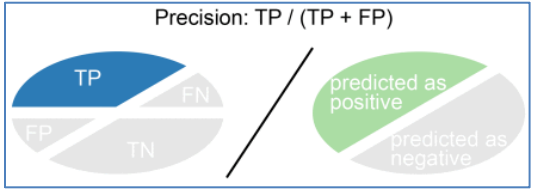




% of correct predictions



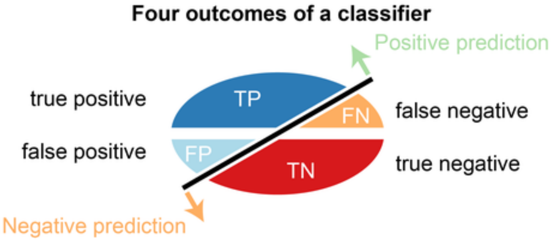
% of + class correctly predicted
[aka Recall / TPR]

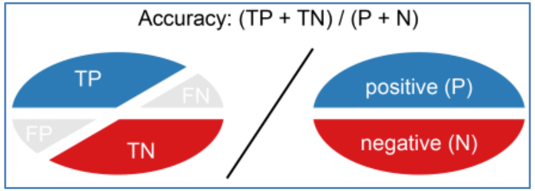


correct prediction of + class
[aka Precision]

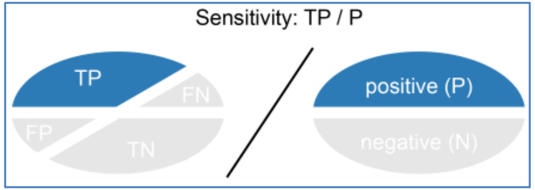
Accuracy vs Precision vs Recall

- Monitor **Precision** if a false positive carries higher cost.
- Monitor **Recall** if a false negative carries higher cost.

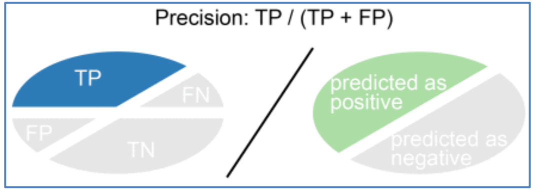




% of correct predictions

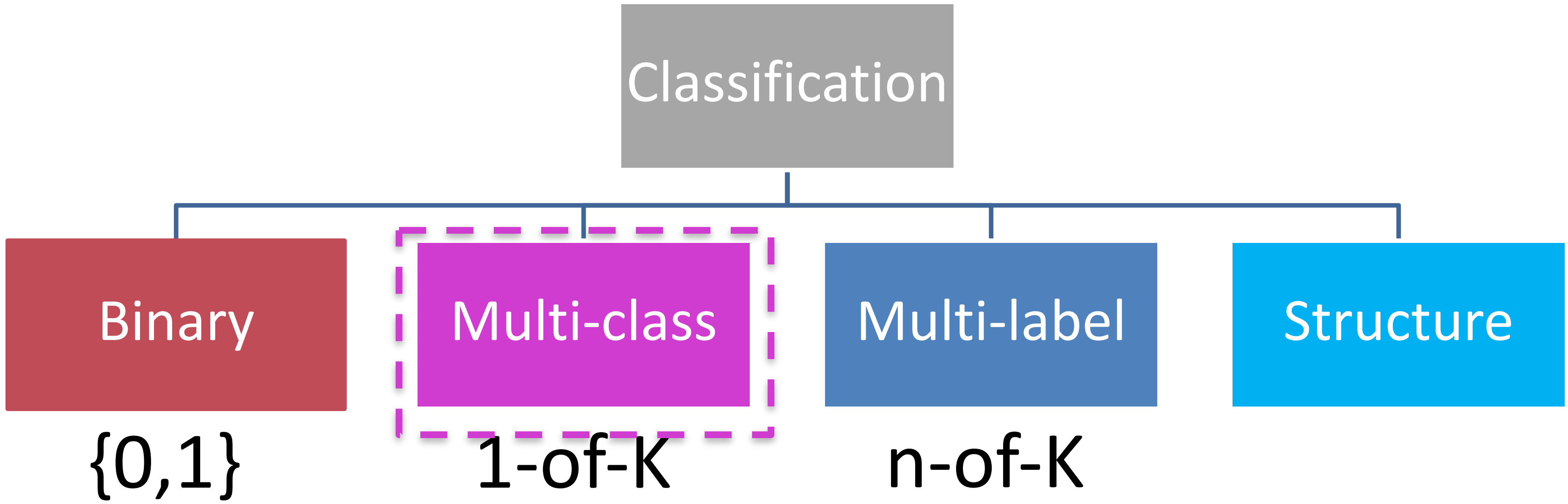


% of + class correctly predicted
[aka Recall / TPR]



correct prediction of + class
[aka Precision]

Precision, Recall, F1 score


Classification
Binary
Multi-class
Multi-label
Structure
{0,1}
1-of-K
n-of-K

Multi-class classification

https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf#page=76

Multi-class classification

Multi-class Classification - Confusion matrix

activity recognition from video

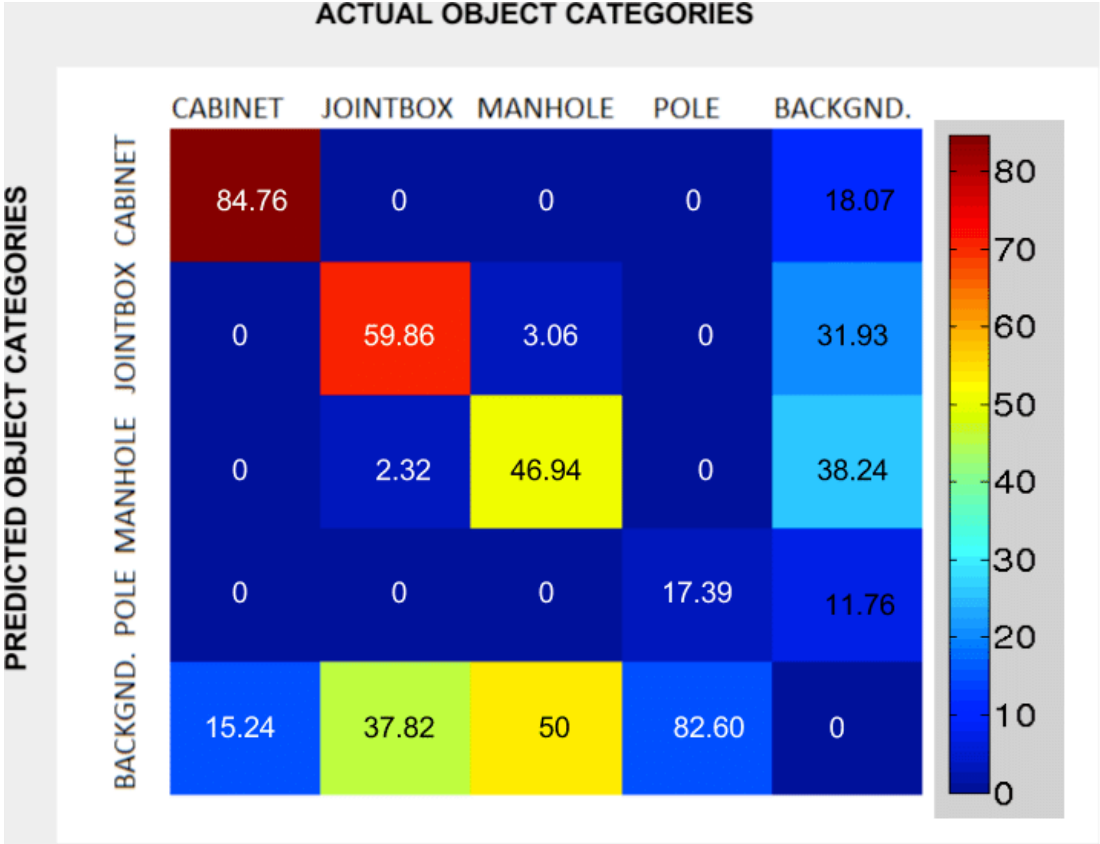
actual class / predicted class

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	100	0	0	0	0	0	0	0	0	0
jack	0	100	0	0	0	0	0	0	0	0
jump	0	0	89	0	0	0	11	0	0	0
pjump	0	0	0	100	0	0	0	0	0	0
run	0	0	0	0	89	0	11	0	0	0
side	0	0	0	0	0	100	0	0	0	0
skip	0	0	0	0	0	0	100	0	0	0
walk	0	0	0	0	0	0	0	100	0	0
wave1	0	0	0	0	0	0	0	0	67	33
wave2	0	0	0	0	0	0	0	0	0	100

Courtesy: vision.jhu.edu

- Reveals several biases of the classifier:
  - Most confusing pairs
  - Least / Most likely classes

Multi-class Classification - Confusion matrix



- Reveals several biases of the classifier:
  - Most confusing pairs
  - Least / Most likely classes

https://www.researchgate.net/figure/Confusion-Matrix-Heat-Map-This-figure-depicts-that-Cabinets-jointboxes-and-manholes-have_fig5_315619273

Multi-class Classification: Measures

- Mean <measure> +- standard deviation
- Median <measure> +- median absolute deviation

Descriptor	Spectral bands	
	RGB	PCA RGB
Gist	74.14 ± 1.93	77.76 ± 2.62
MSIFT	88.92 ± 1.39	90.97 ± 1.81
MBoW	88.60 ± 1.70	88.31 ± 1.38
cSIFT	88.17 ± 1.17	88.76 ± 1.74
rgSIFT	88.24 ± 1.89	87.71 ± 1.33
BoWV [8]	71.86	N/A
SPMK [12]	74.00	N/A
SPCK++ [8]	76.05	N/A
Dense SIFT [2]	81.67 ± 1.23	N/A

(Sanity Check) Baselines

- 0 cost-to-build classifiers
- Binary: Assume equal # of samples / class
  - What is random guessing accuracy ?

(Sanity Check) Baselines

- 0 cost-to-build classifiers
- Binary: Assume unequal # of samples / class [n=#samples, x= % of + class (!= 50%) , - class is majority in training set]
  - Classifier: Predict majority class for all samples.
  - Accuracy ?

(Sanity Check) Baselines

- Multi-class: Assume equal # of samples / class [k classes, N total samples]
  - What is random guessing accuracy ?

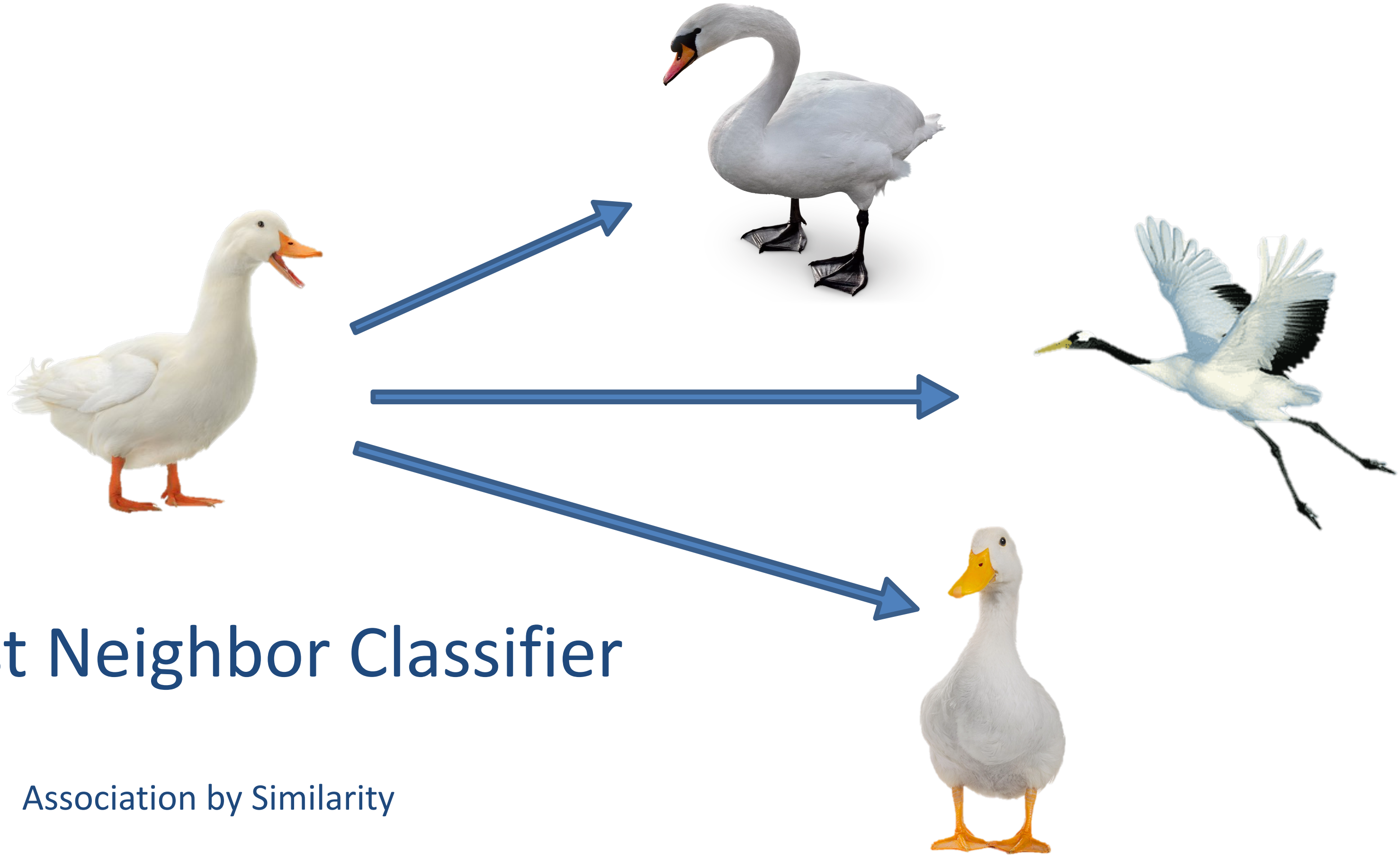
Summary

- Many metrics:
  - Accuracy, TP, FP, Precision, Recall
  - Class imbalance and decision-cost imbalance must be taken into account

- Confusion Matrix: Important to analyze and refine solution.

References and Reading

- Code
  - https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics



Nearest Neighbor Classifier

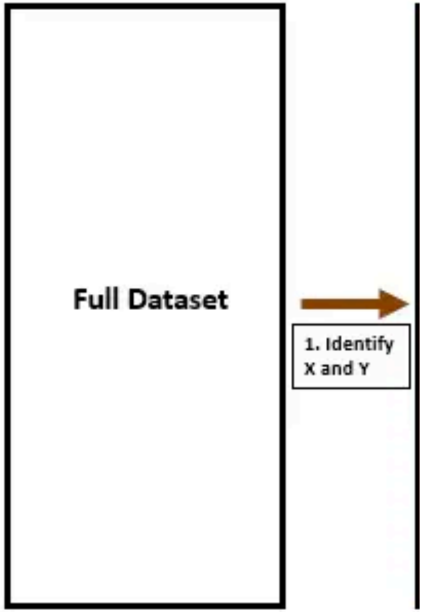
Association by Similarity

Goal of ML Classification

- Given:
  - A set of training $\boldsymbol{n}$ training samples: $(\mathbf{x}_i, y_i)$
- Correctly predict
  - Labels of **unseen** test samples: $\mathbf{x}_t$
- Goal:
  - Maximize the **accuracy** on unseen test samples

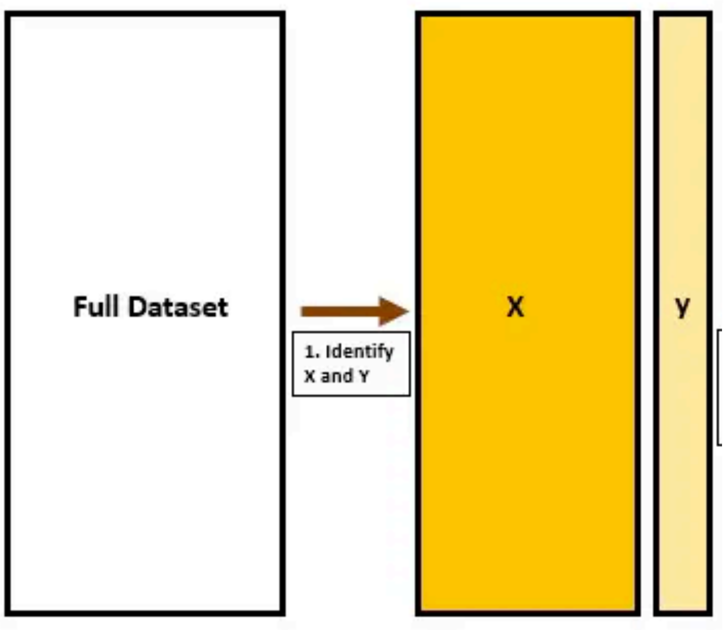
- How do we find the **accuracy** on unseen samples?

ML Train-Test Setup



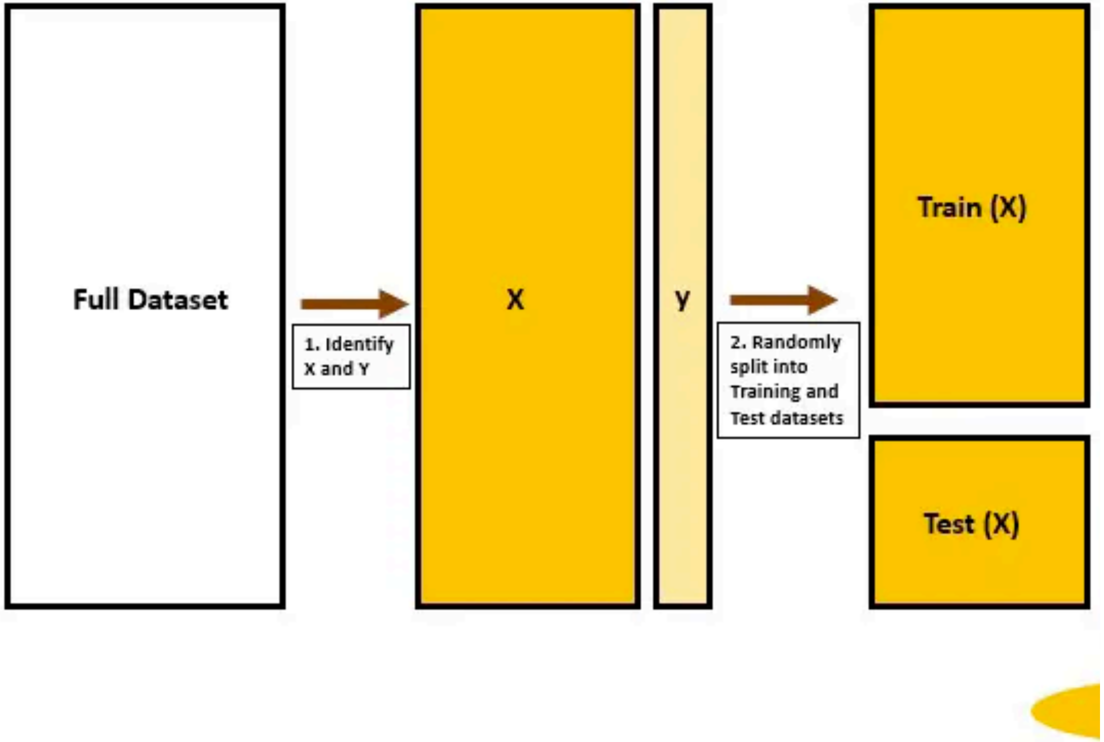
https://www.machinelearningplus.com/machine-learning/train-test-split/

ML Train-Test Setup



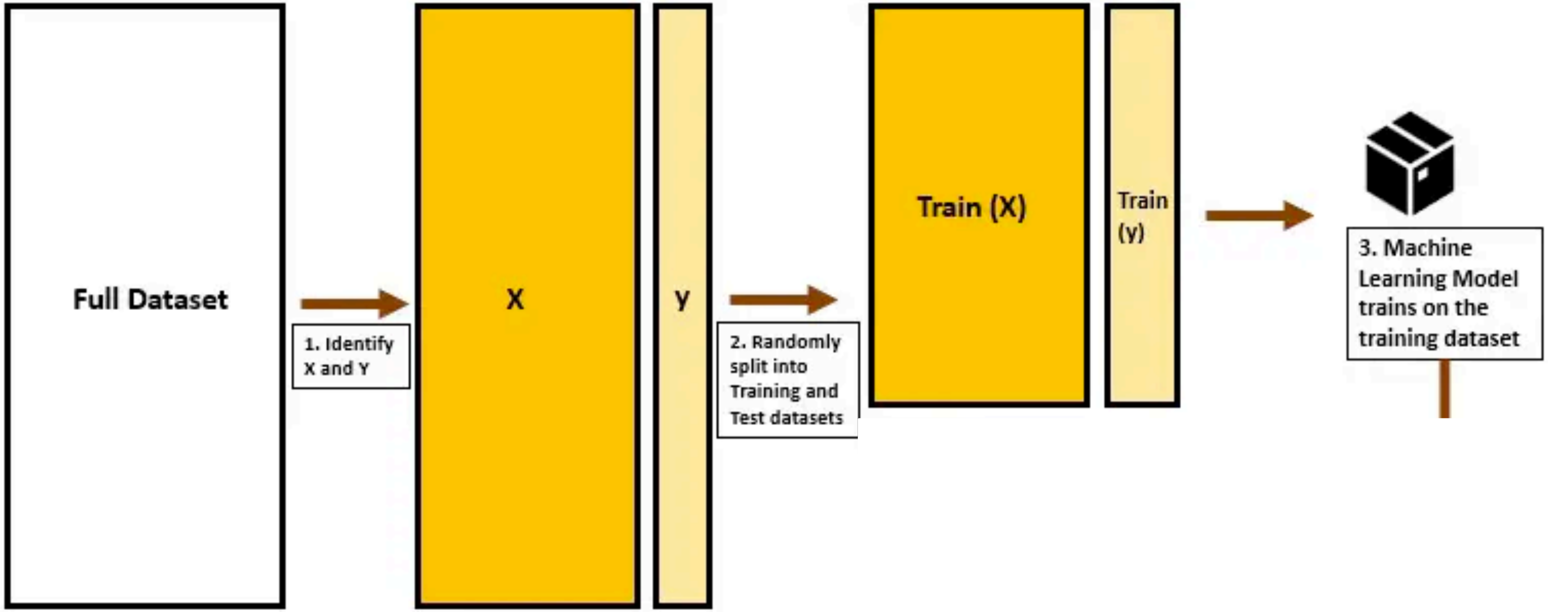
https://www.machinelearningplus.com/machine-learning/train-test-split/

ML Train-Test Setup



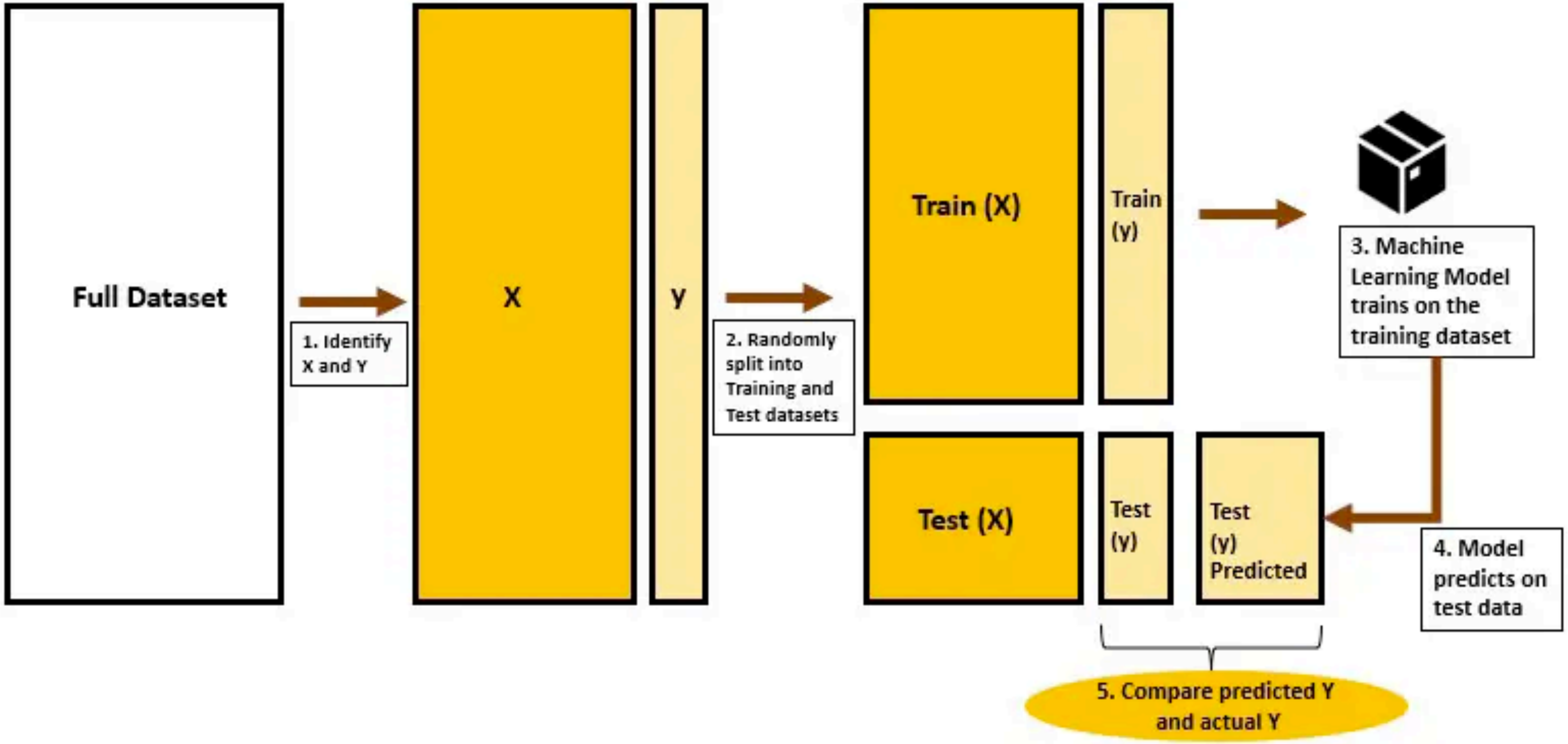
https://www.machinelearningplus.com/machine-learning/train-test-split/

ML Train-Test Setup



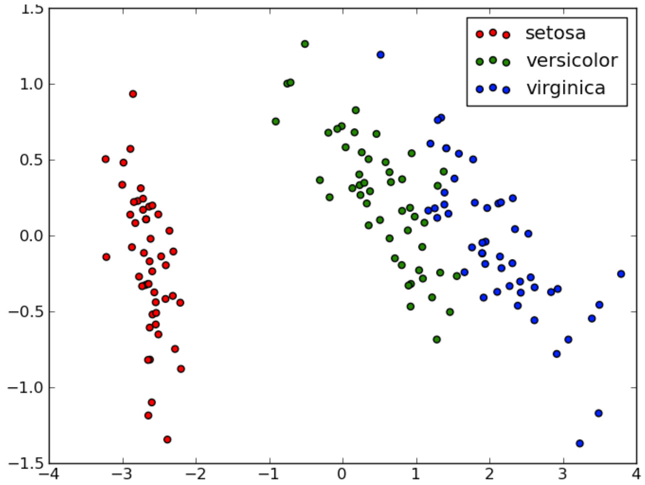
https://www.machinelearningplus.com/machine-learning/train-test-split/

ML Train-Test Setup



https://www.machinelearningplus.com/machine-learning/train-test-split/

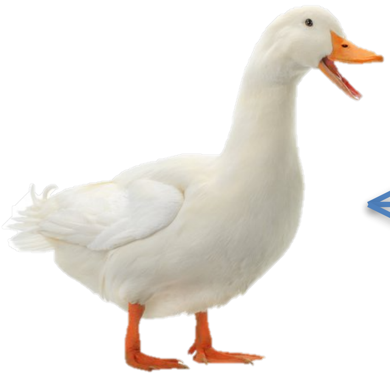
Data Samples → Vectors



https%3A%2F%2Fscikit-learn.sourceforge.net%2F0.5%2Fauto_examples%2Fplot_pca.html

How do we compare?

If it looks like a duck, walks like a duck and quacks like a duck,



Look (Appearance)
Walk (Motion)
Quack (Voice)

Like a Duck?

Duck
Look-D
Walk-D
Quack-D

- Find distance to feature vectors of known classes

Nearest Neighbor Classifier

- Assign label of that sample which is nearest to the test sample

X_{test} :

[30.9, 15.1, 1.32]

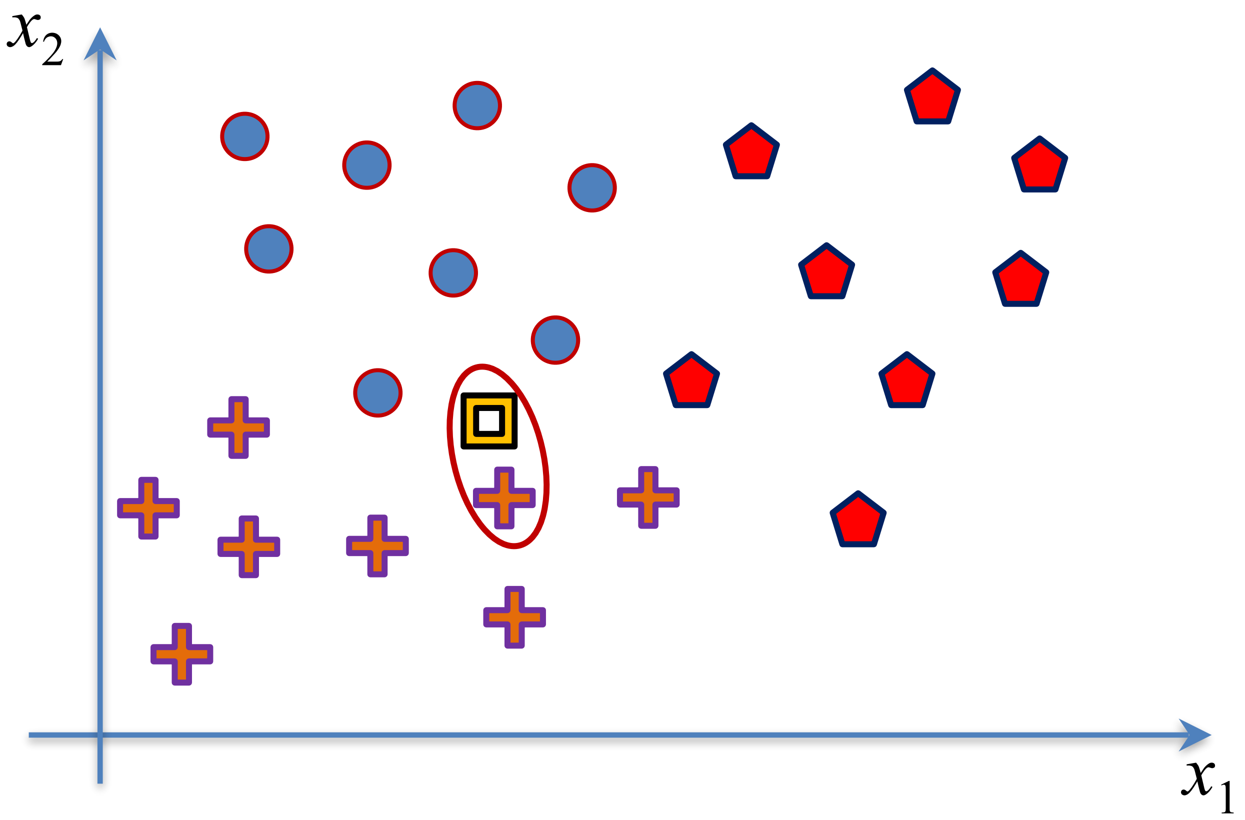
The test sample is a Duck

Training Samples

Distance	Feature Vector	Label
1.30	X_1 [32.1, 14.6, 1.42]	Duck
5.78	X_2 [25.3, 16.3, 2.11]	Swan
13.54	X_3 [42.2, 7.7, 0.38]	Crane
4.71	X_4 [26.7, 17.1, 2.04]	Swan
15.21	X_5 [44.1, 7.6, 0.32]	Crane
3.04	X_6 [31.4, 12.1, 1.29]	Duck
13.58	X_7 [41.9, 7.2, 0.35]	Crane

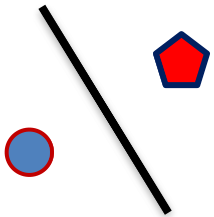
Visualization in in Feature Space

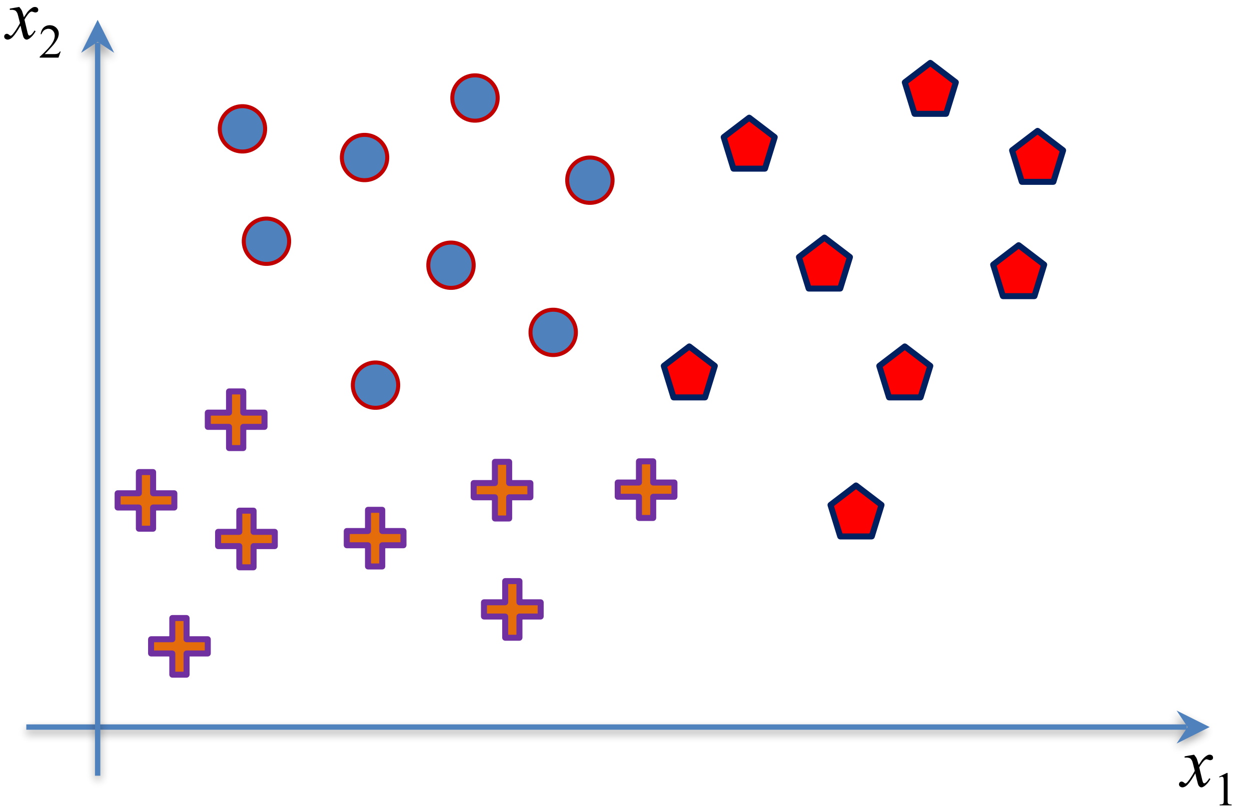
- The nearest train sample is a +.
- We assign the test sample to the + class.



Class Boundaries

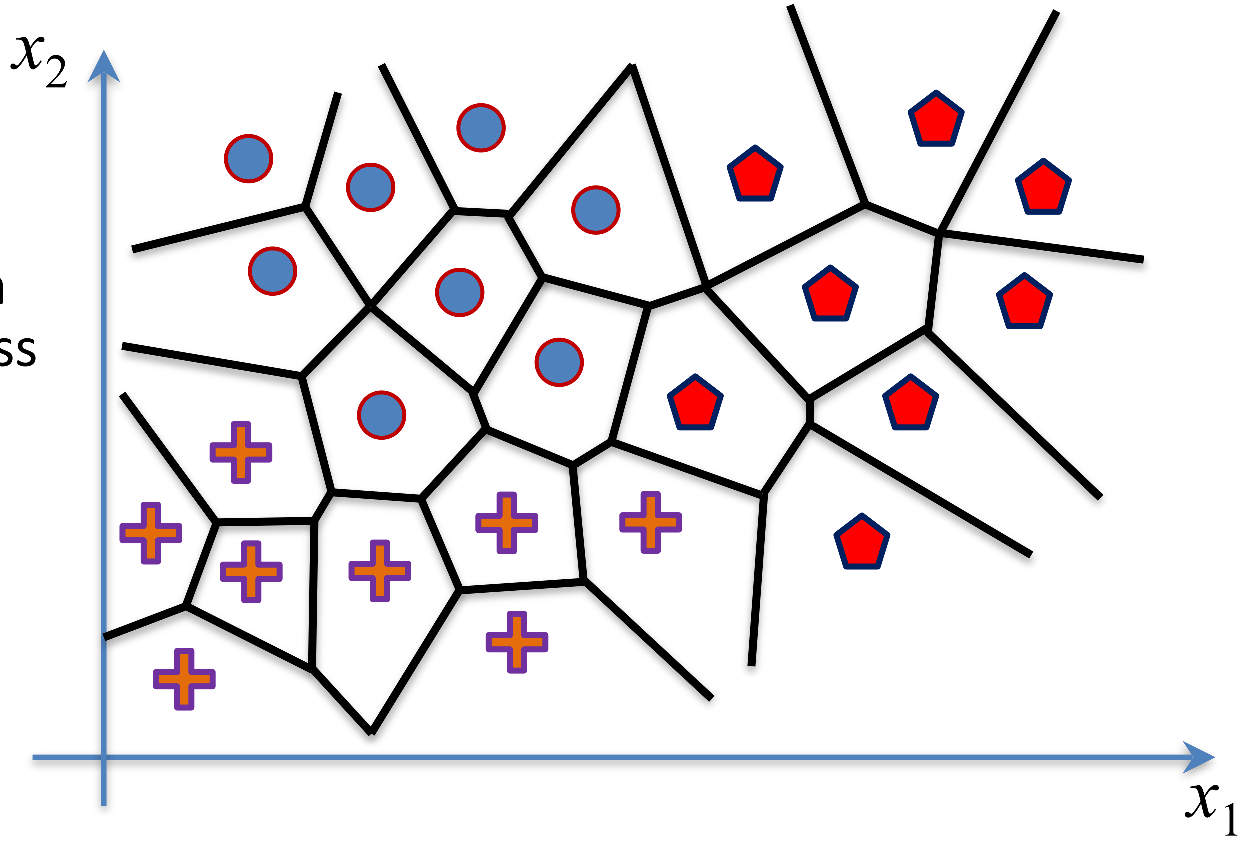
- Partition the feature space
- Consider 2 samples





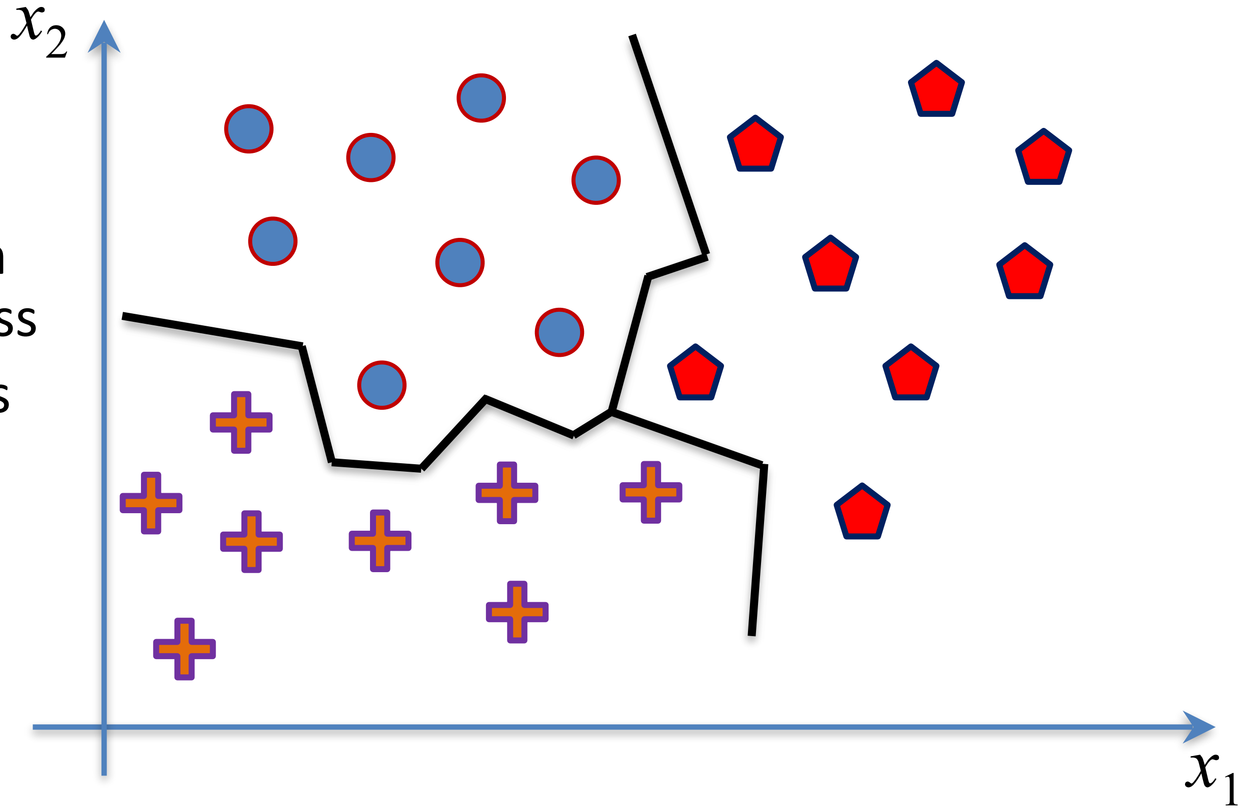
Boundaries between every pair of samples

- Voronoi Tessellation
- We can ignore boundaries between samples of same class



Boundaries between every pair of samples

- Voronoi Tessellation
- We can ignore boundaries between samples of same class
- Decision Boundary is piece-wise Linear



The 1-NN Classification Algorithm

Problem

- Given:
  - A set of training ***n*** training samples: $(\mathbf{x}_i, y_i)$
  - A test sample: $\mathbf{x}_t$
- Find:
  - label($\mathbf{x}_t$)

Algorithm

```
for xi in {Xtrain}:
        dist = Dist(xi, xt)
        if (dist < minDist):
            minDist = dist
            nearest = xi
return (yi)
```

1-NN

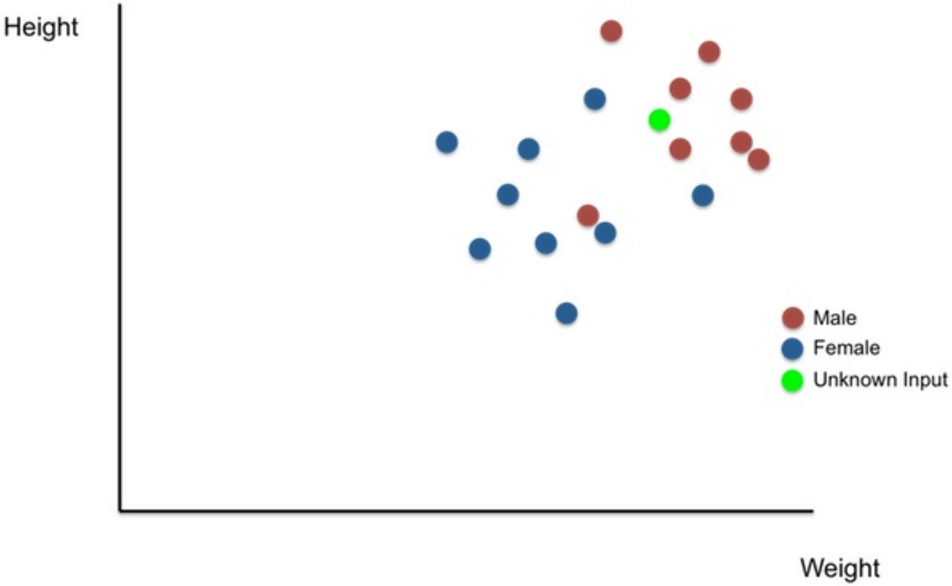
Algorithm:

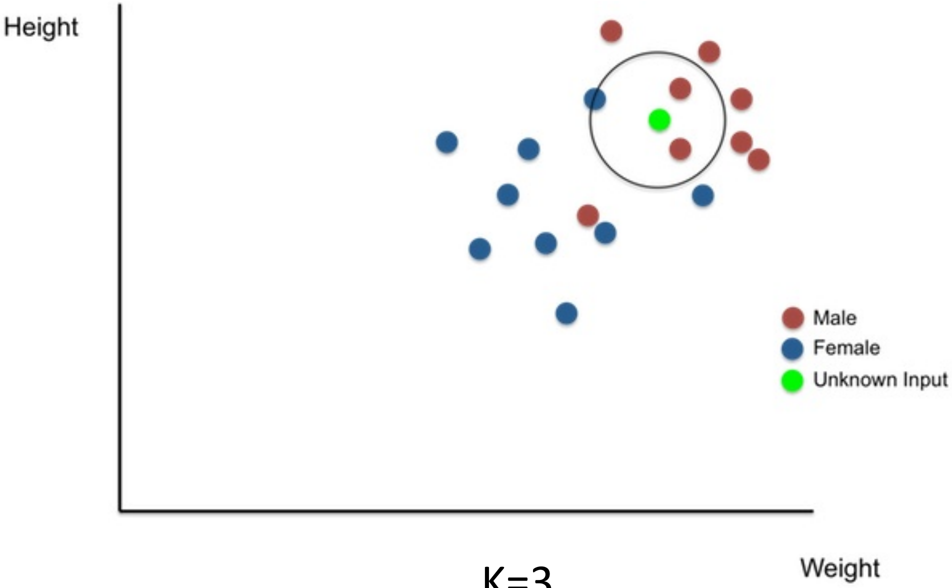
1. Find example $(\mathbf{x}^*, t^*)$ (from the stored training set) closest to the test instance $\mathbf{x}$. That is:

$$\mathbf{x}^* = \underset{\mathbf{x}^{(i)} \in \text{train. set}}{\text{argmin}} \; \text{distance}(\mathbf{x}^{(i)}, \mathbf{x})$$

2. Output $y = t^*$

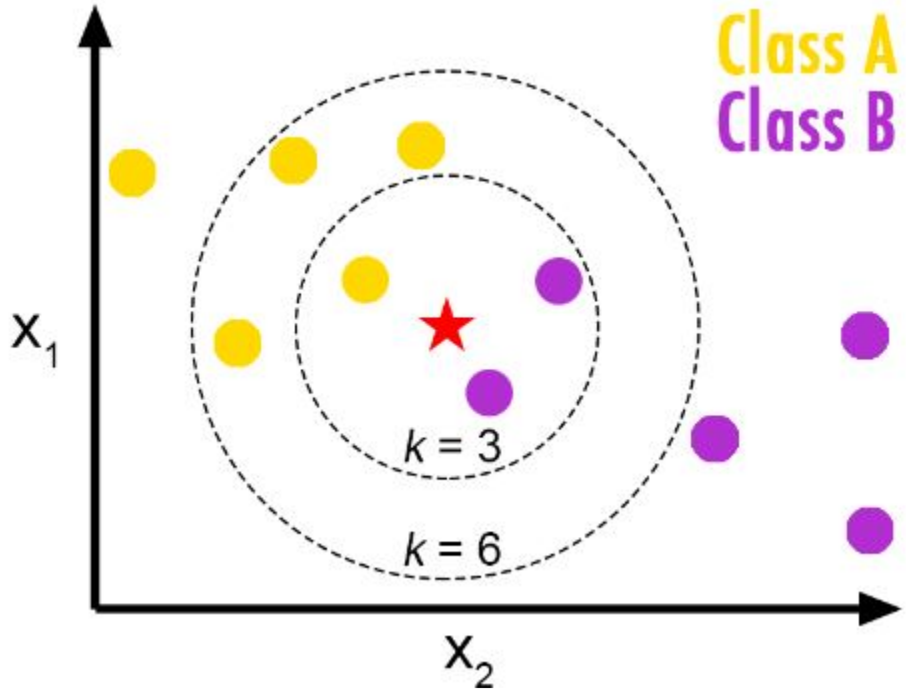
k-nearest neighbor classifier





K=3

k-nearest neighbor classifier



K is usually an odd number

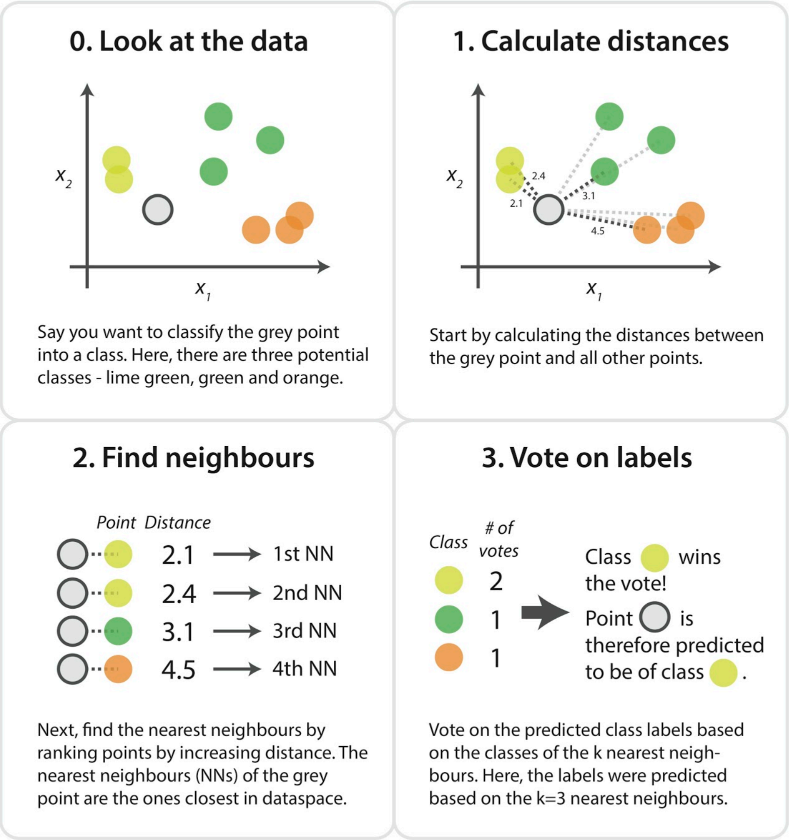
k-NN

Algorithm (kNN):

1. Find k examples $\{\mathbf{x}^{(i)}, t^{(i)}\}$ closest to the test instance $\mathbf{x}$
2. Classification output is majority class

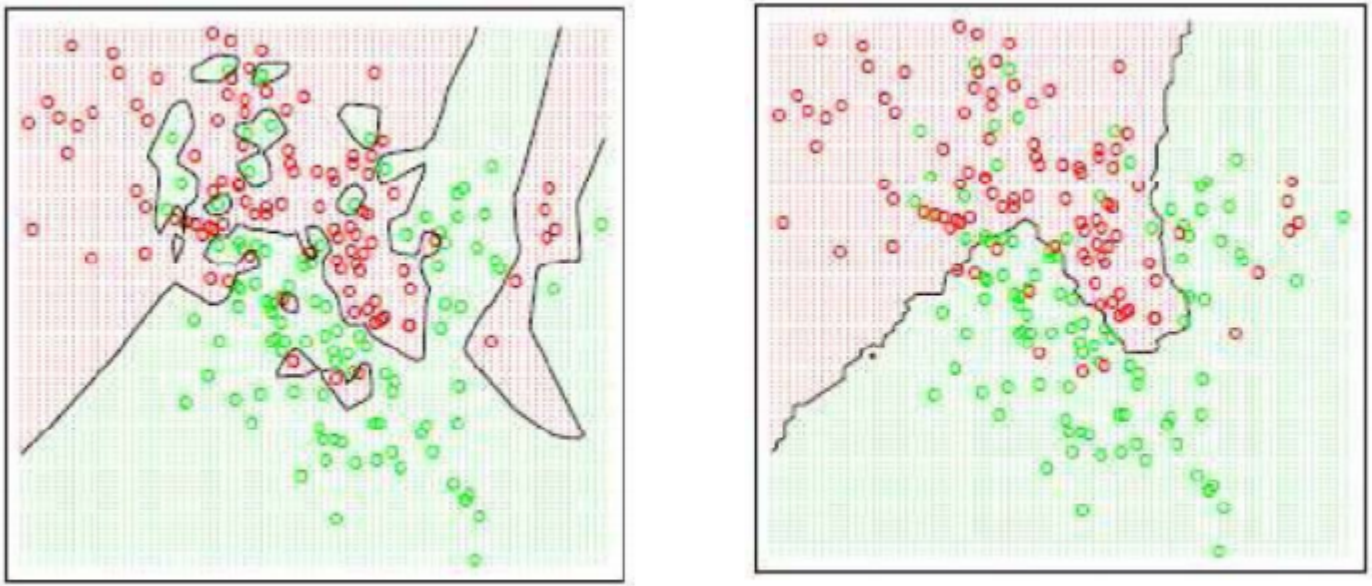
$$y = arg \max_{t^{(z)}} \sum_{r=1}^{k} \delta(t^{(z)}, t^{(r)})$$

k-NN algorithm in pictures



Effect of K

K=1 K=15



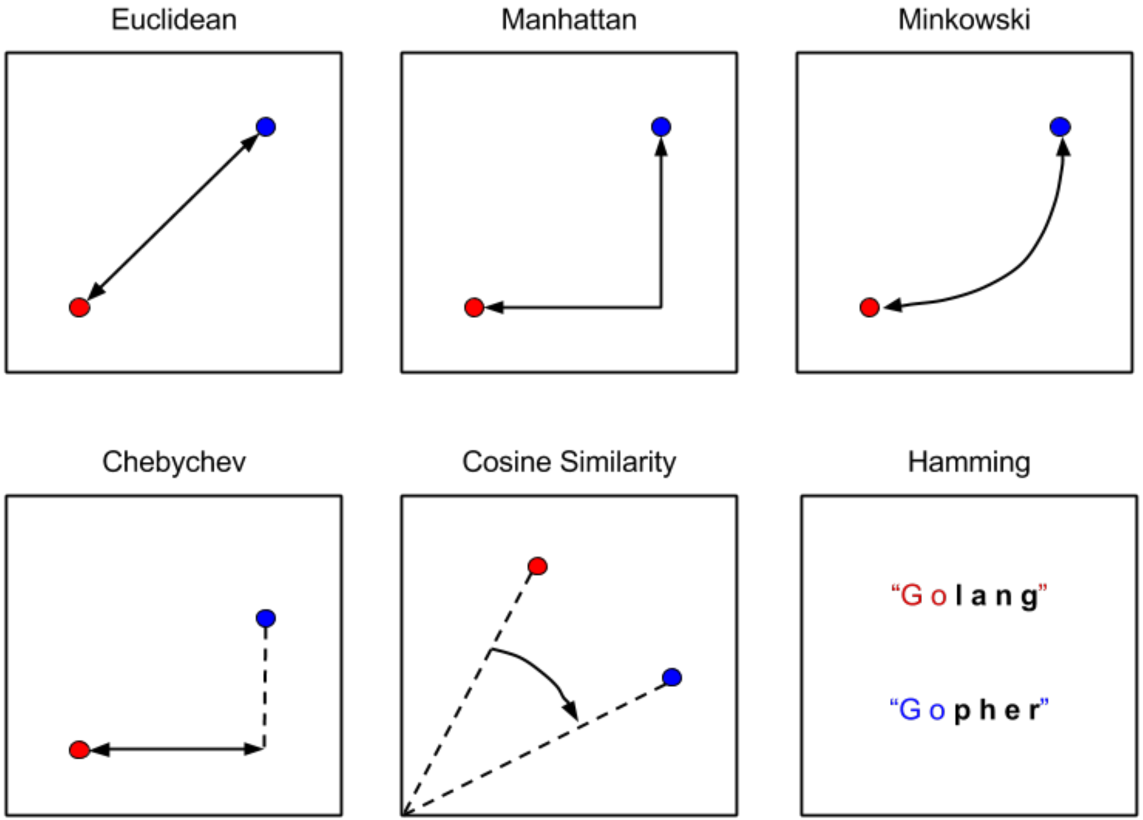
Figures from Hastie, Tibshirani and Friedman (Elements of Statistical Learning)

Larger k produces smoother boundary effect and can reduce the impact of class label noise.

Complexity of k-NN

- Training
  - Time:
  - Space:
- Testing
  - Time:
  - Space:

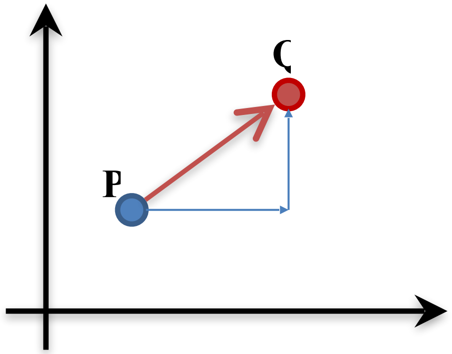
Distance measures



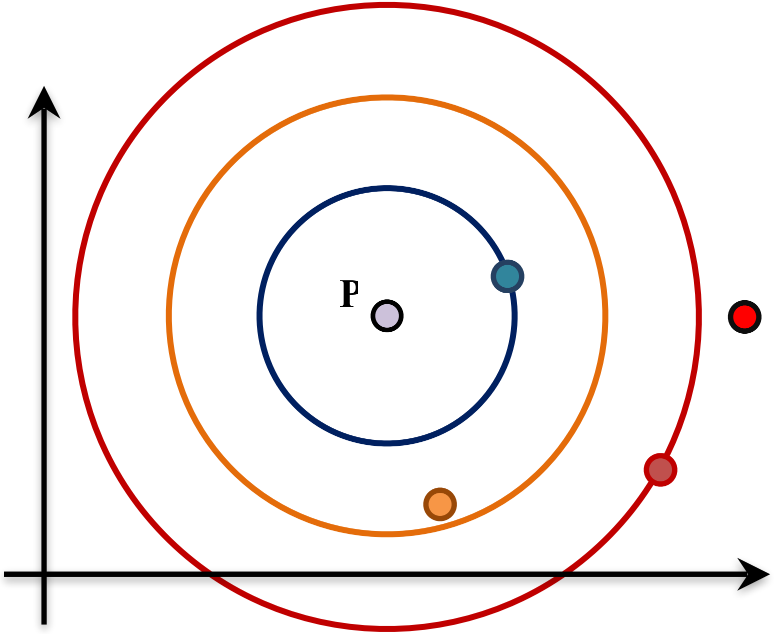
Euclidean Distance [L2]

$$d(P,Q)^2 = \sum_{i=1}^{d} \left(p_i - q_i\right)^2$$

$$d(P,Q)^2 = (P - Q)^T (P - Q)$$

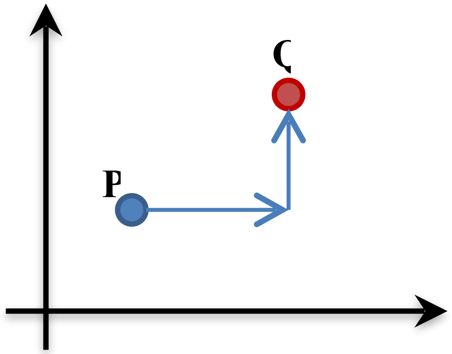


Iso-surfaces

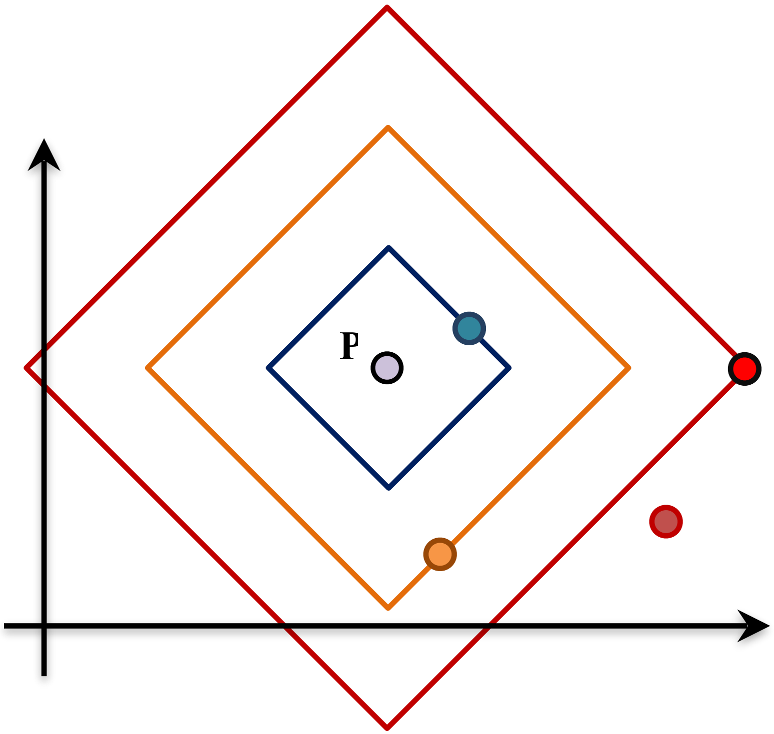


Manhattan Distance [L1]

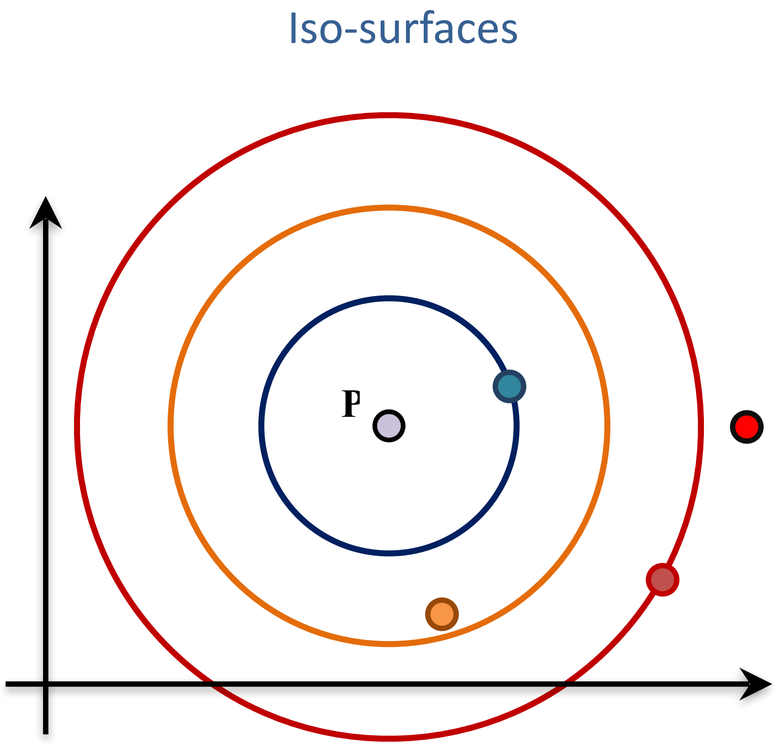
$$d(P,Q) = \sum_{i=1}^{d} \left| \left(p_i - q_i\right) \right|$$



Iso-surfaces

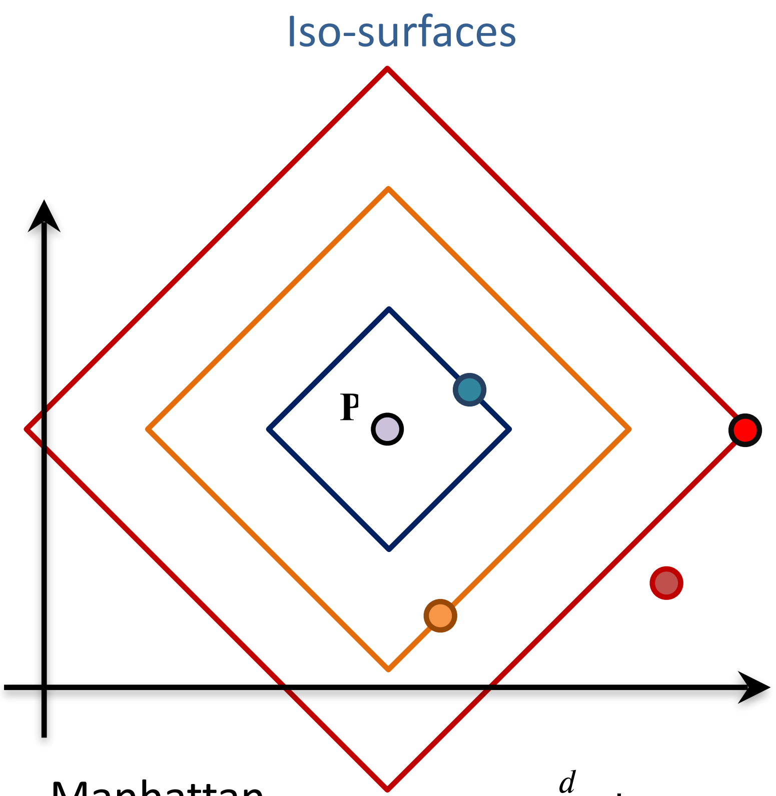


Distance measure can affect k-NN classification



Euclidean

$$d(P,Q)^2 = \sum_{i=1}^{d} \left(p_i - q_i\right)^2$$

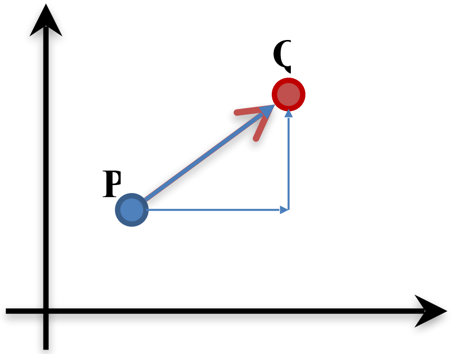


Manhattan

$$d(P,Q) = \sum_{i=1}^{d} \left| \left(p_i - q_i\right) \right|$$

Minkowski Dist.

$$d(P,Q)^r = \sum_{i=1}^{d} \left| p_i - q_i \right|^r$$



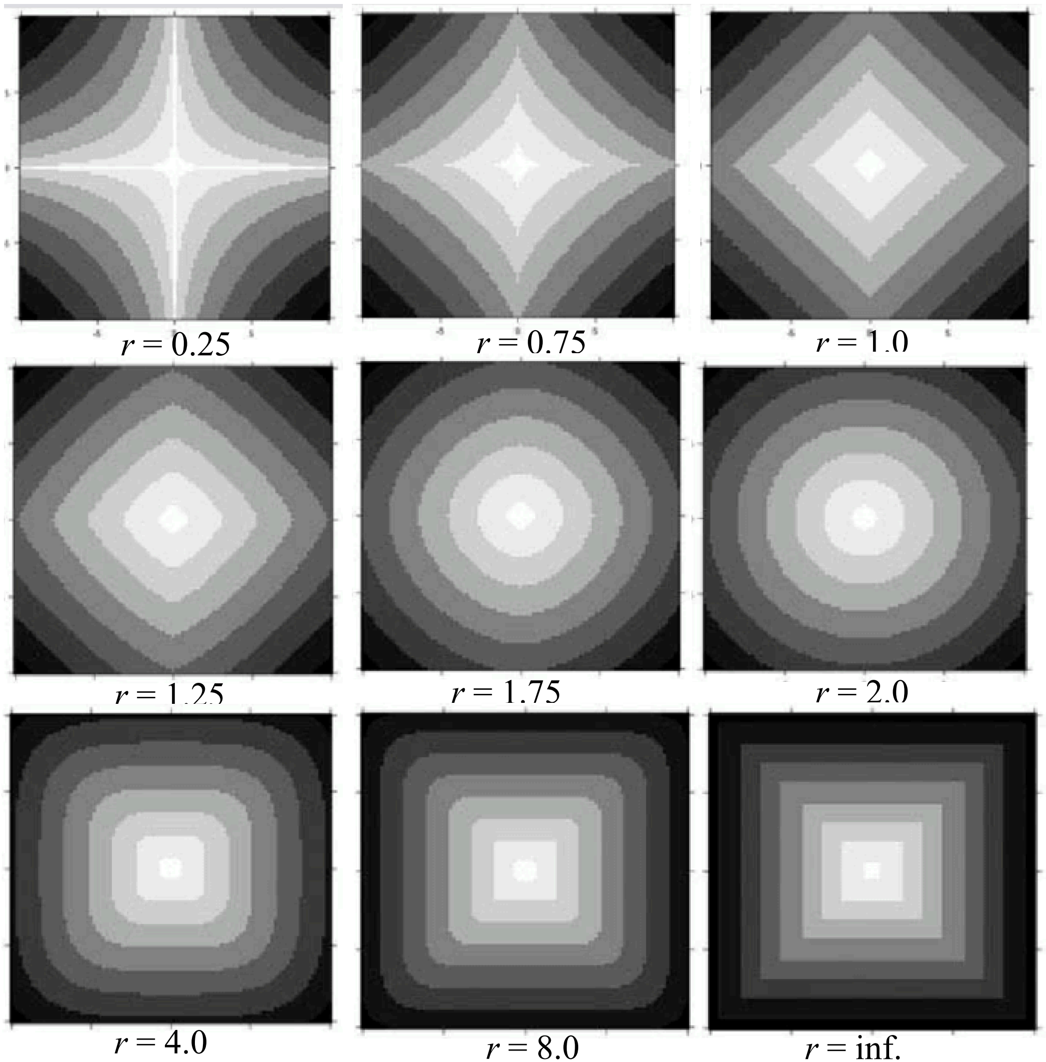
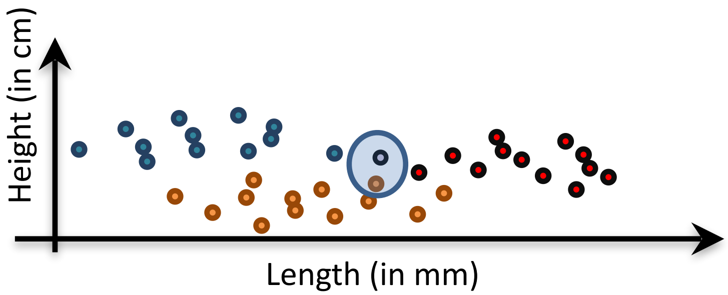
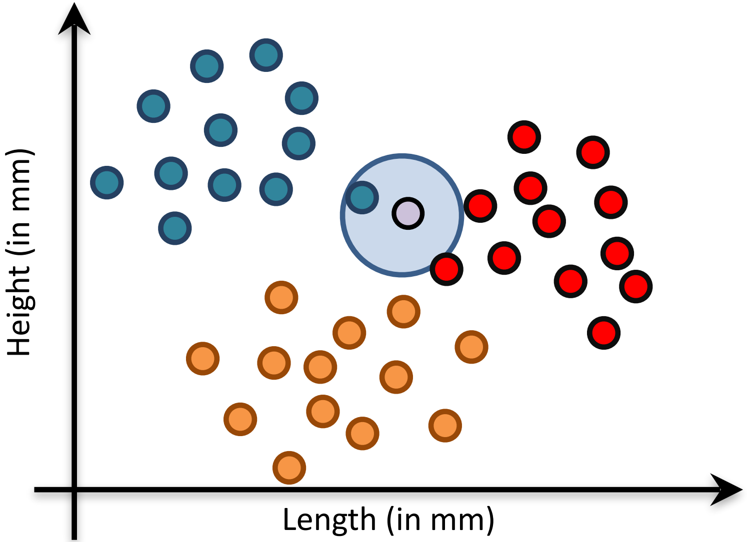


Fig. by Lu et al., "The Minkowski Approach for Choosing the Distance Metric in Geographically Weighted Regression"

Note: Feature Normalization

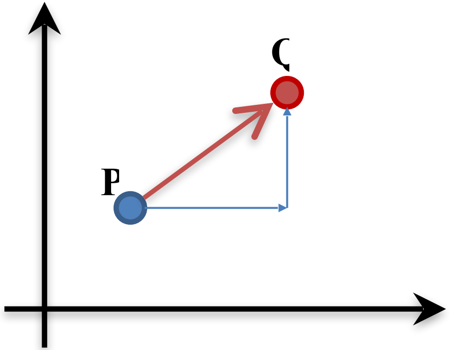
- If different features have different variances
  - Some features will dominate distance computation
  - Normalization can reduce this feature bias

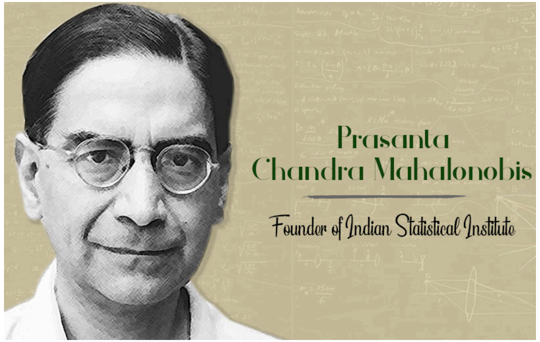




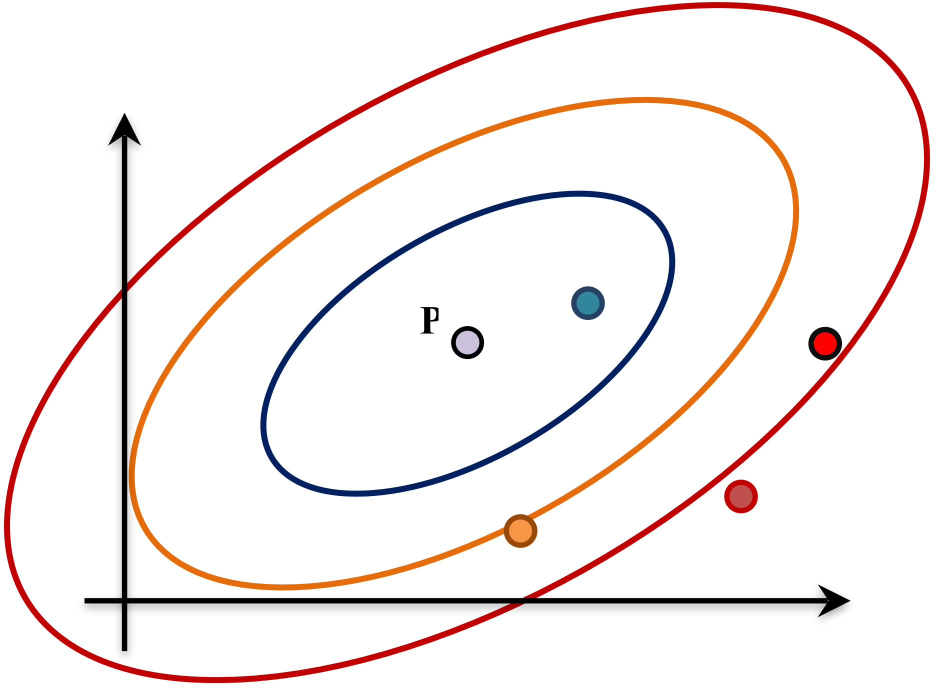
Mahalanobis Distance

$$d(P,Q)^2 = (P-Q)^T \mathbf{S}^{-1}(P-Q)$$





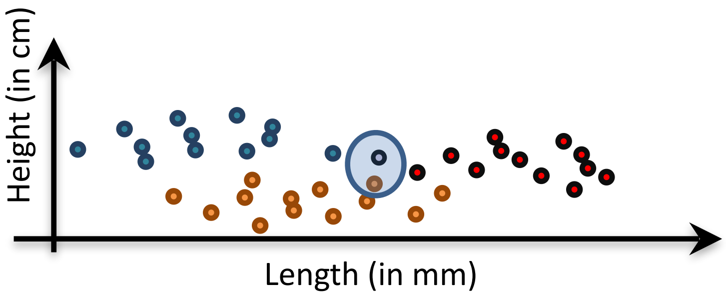
Iso-surfaces

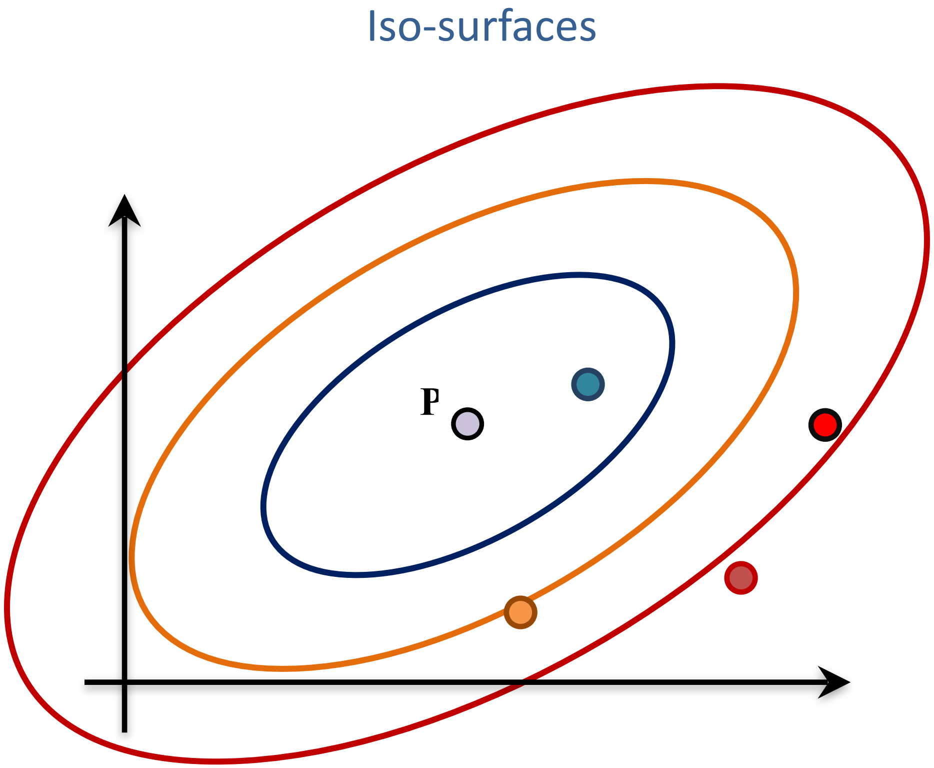


https://www.getbengal.com/details/the-indian-statistical-institute-in-kolkata-was-founded-by-the-father-of-indian-statistics-pc--mahalanobis-getbengal-story

Mahalanobis Distance

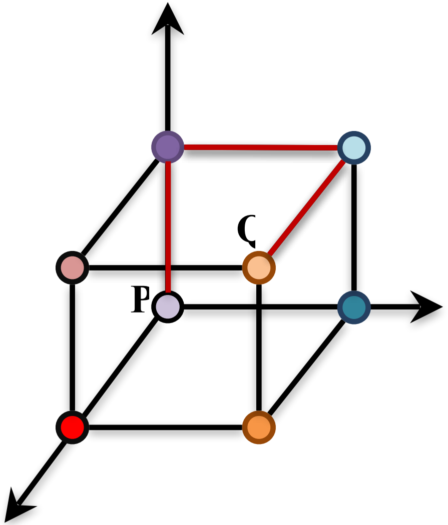
$$d(P,Q)^2 = (P-Q)^T \mathbf{S}^{-1}(P-Q)$$

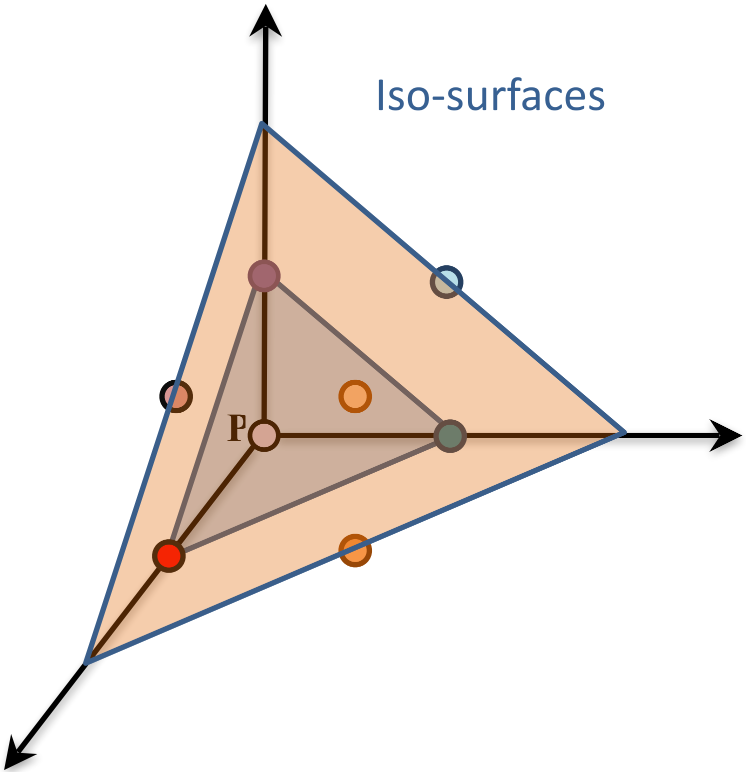




Hamming Distance

$$d(P,Q) = \sum_{i=1}^{d} \mathrm{I}\left(p_i \neq q_i\right)$$



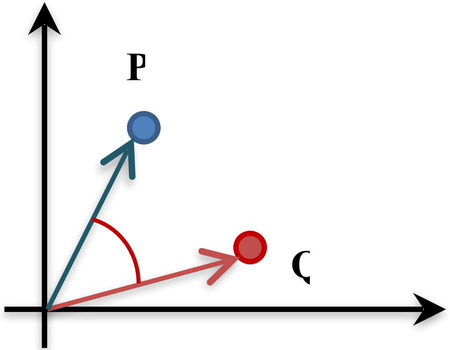


0	1	1	0	1	1	0
0	0	1	0	1	0	0

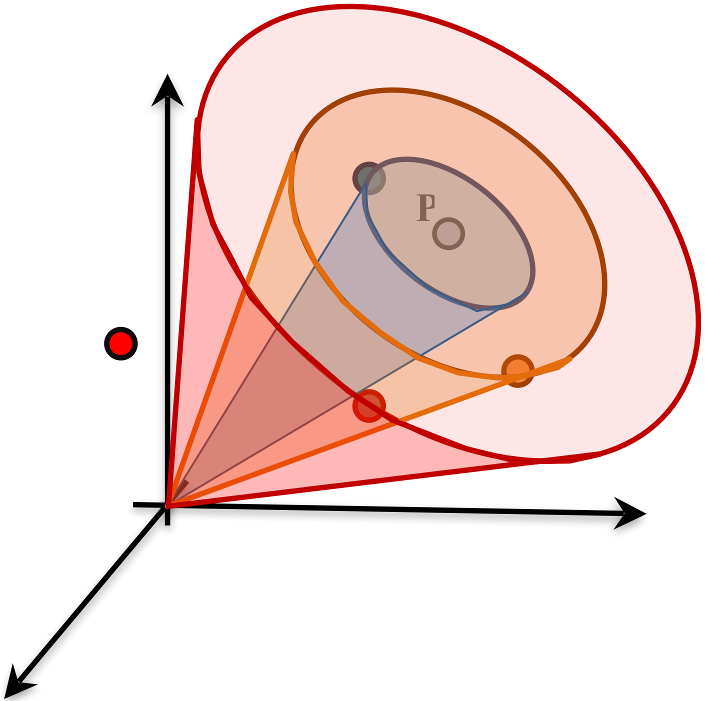
Cosine Distance

$$s(P, Q) = cos\theta = \frac{\mathbf{P} \cdot \mathbf{Q}}{\|\mathbf{P}\| \|\mathbf{Q}\|}$$

$$d(P, Q) = 1 - s(P, Q)$$



Iso-surfaces



Summary of Distances

- Distance Metrics decide Neighborhoods
- Need not be a ”metric”
  - Jaccard Distance
  - Edit Distance
- Feature vectors need not be of same length
- Selection of metric depends on the nature of feature vector