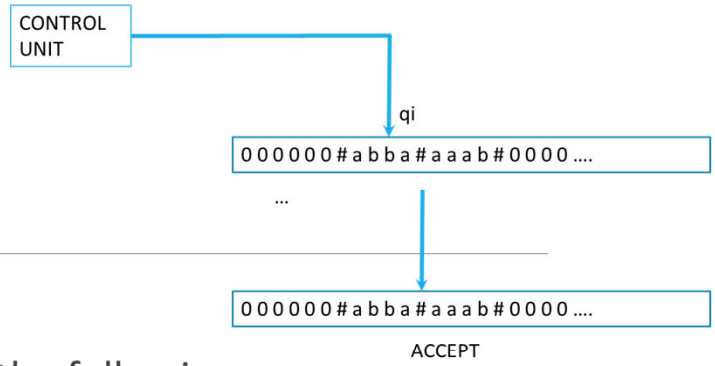
Machine Learning Examples in Research

PROF. MIKE TEODORESCU (TEODORES@BC.EDU)

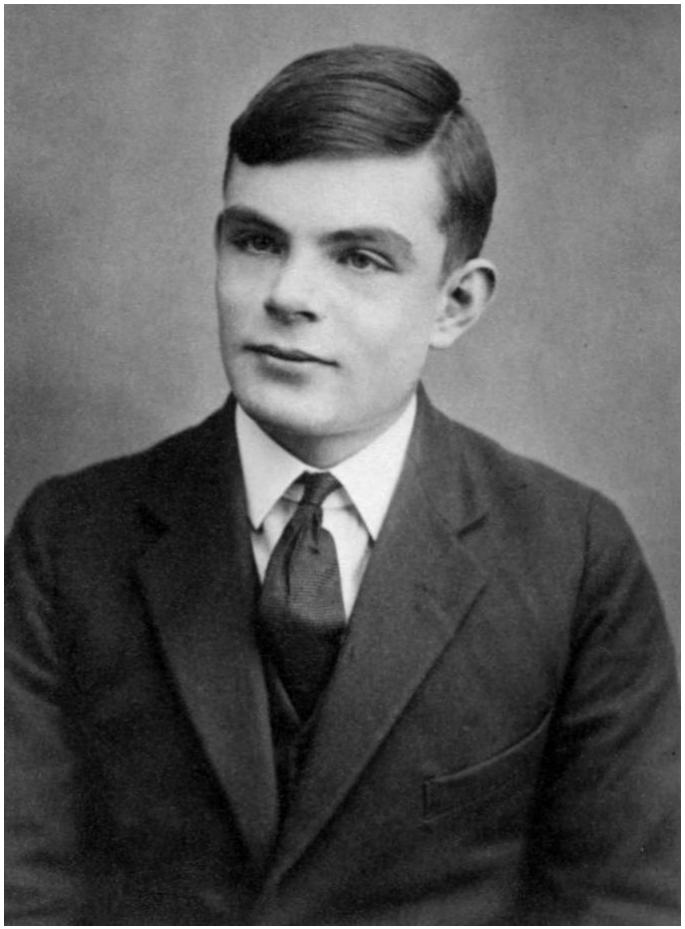
What is “computation”?



http://en.wikipedia.org/wiki/Typewriter#mediaviewer/File:1920s_Underwood_SE_layout.JPG

The computational model featuring the following:

- An infinite tape (i.e. unlimited memory) to be used for read/write
- A tape head that can read, write, and move around freely either LEFT or RIGHT on the tape
- A set of states that must include an ACCEPT and a REJECT state
- An input alphabet
- A special symbol (blank space, ^, # etc) **not** included in input alphabet
- A transition function (think of it as a lookup table of allowable instructions)



Alan Turing, 1912-1954

Machine Learning – What is it?

- A field at the intersection of computer science, statistics, and neuroscience that enables computer systems to perform tasks without instructions, draw inferences from data and generalize, and translate human queries into machine instructions.
- The study of methods that enable computers to find patterns in data and use those patterns to construct predictions (aka "data science")
- Alan Turing defined the intelligent machine as one that could respond to a human in a way indistinguishable from a human.
- While we do not yet have **an intelligent machine**, we have pretty good **prediction capabilities.**

What is “Learning”?

The machine learns to behave in accordance with the statistics of the population

The machine utilizes algorithms to draw inferences based on data which improve with more data

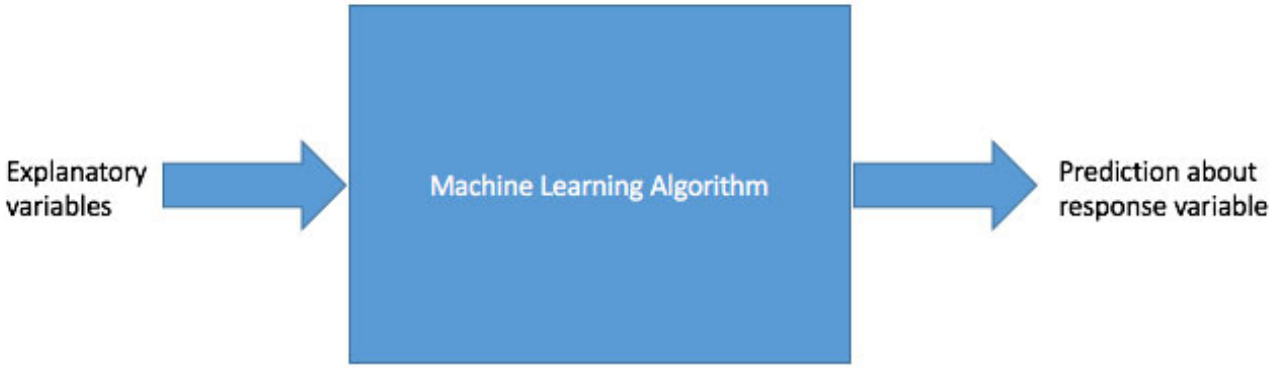
Data component:

- Training set
- Test set

Supervision:

- Supervised learning – human checks each prediction; algorithm adjusts based on corrections
- Unsupervised learning – algorithm runs according to a preprogrammed optimization criterion

What is “Learning”? (a visual)



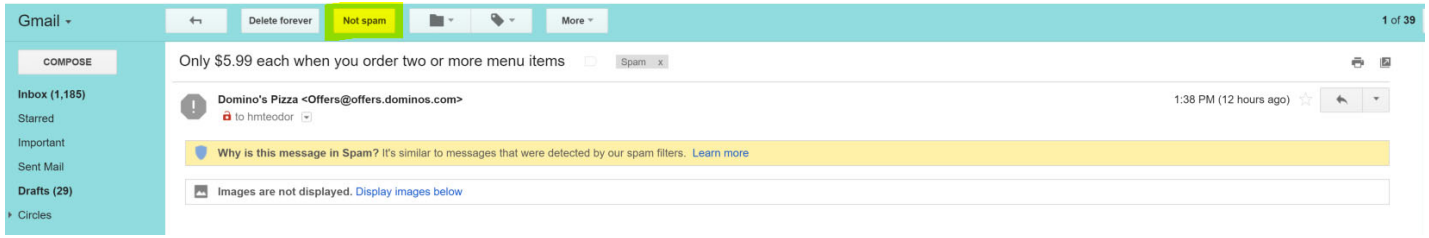
https://lelandkrych.wordpress.com/2016/05/10/machine-learning-for-data-analysis-decision-trees/

Supervised vs unsupervised learning

Learning = the machine learns to behave in accordance with the statistics of the population

Implicitly, makes a representation of the population statistics and uses it for deciding in new cases

Supervised learning: a set of examples are given and the machine “learns from mistakes and successes” in solving those examples; is taught when the answer is correct or erroneous.



Unsupervised: no examples, no corrections from a teacher / user

‘Labels’

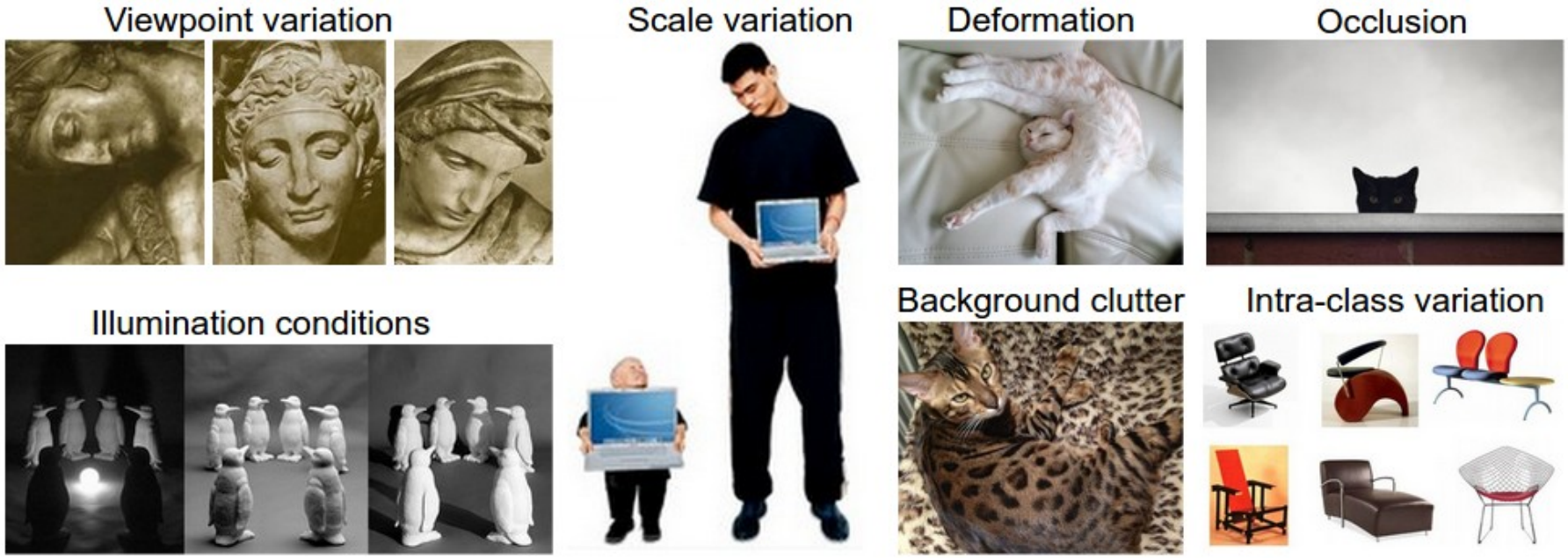
Supervised learning expects the predictors (X’s) to be paired with their correct outputs (Y’s).

This is just another way of saying we know what the values of the dependent variable are (“target” variable)

In machine learning speak, these values are called “labels”

Logistic regression is the simplest classifier

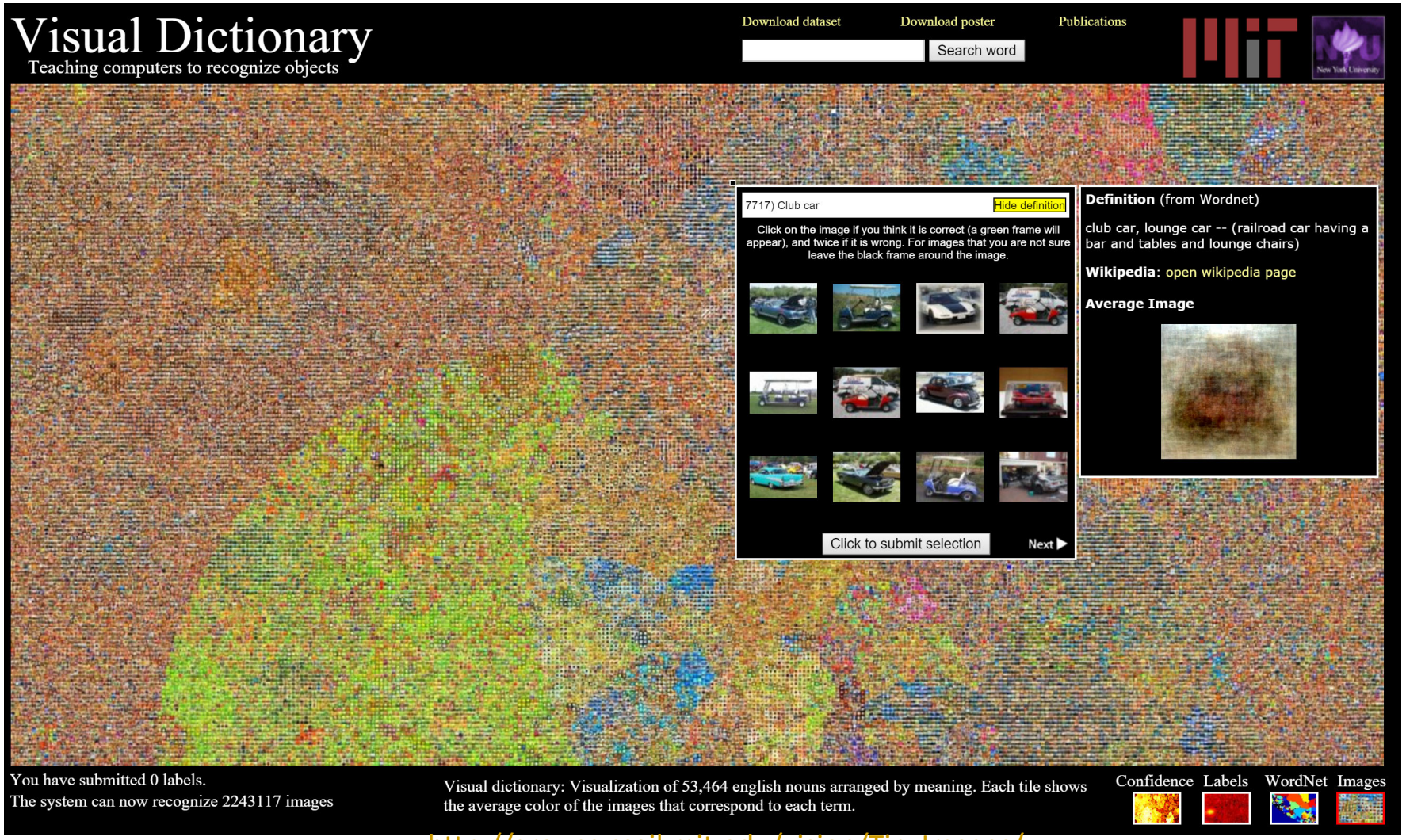
Classification Challenges



http://cs231n.github.io/assets/challenges.jpeg

Image Classification is Very Hard

https://www.freecodecamp.org/news/chihuahua-or-muffin-my-search-for-the-best-computer-vision-api-cbda4d6b425d/

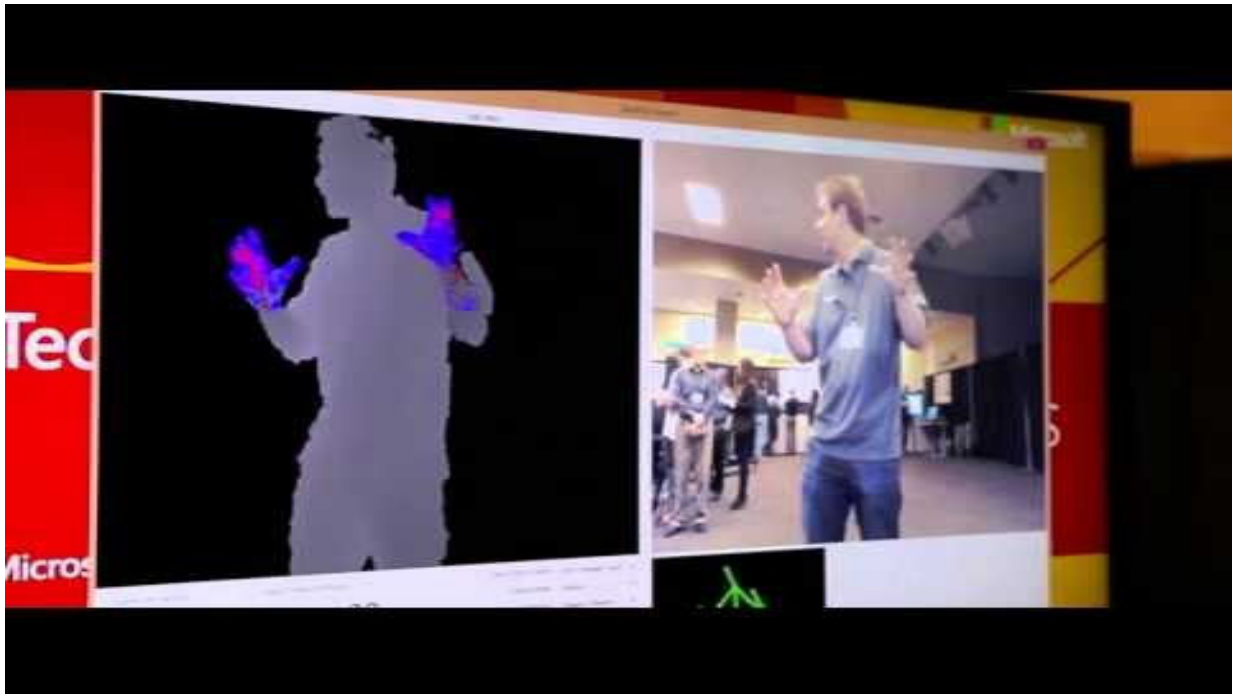


http://groups.csail.mit.edu/vision/TinyImages/

Classification Examples – Music

http://thesis.flyingpudding.com/videos/demo/index.html

Interaction via Kinect Gestures – Also Classification



https://www.youtube.com/watch?v=MwZMNMmODJA

Classification Examples - Text

Sentiment Analysis

Author determination

Spam filtering (context filtering)

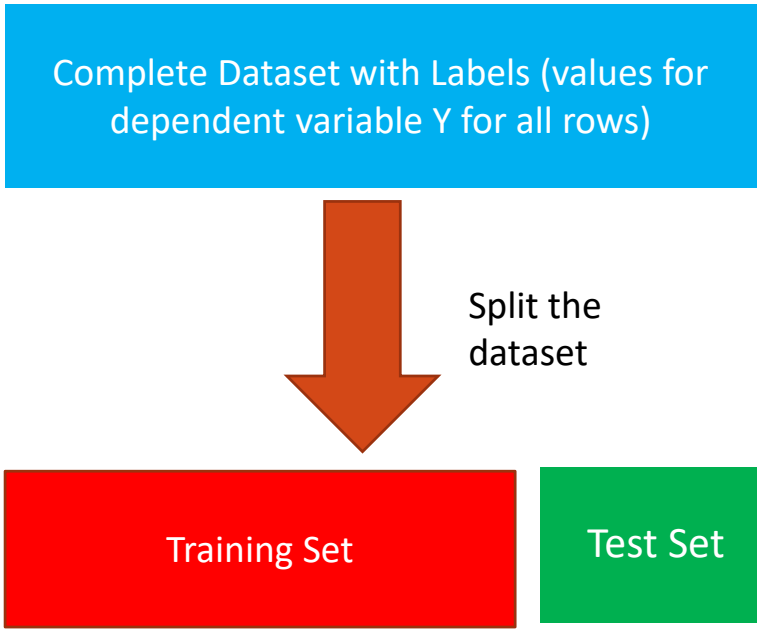
Integrity of recommendations (Yelp, Amazon)

Text belonging to a particular type of document

Basics: Train, Evaluate, Use

- Train the algorithm on the training set
- Test the algorithm on the test set [Evaluate]
- Use the algorithm elsewhere

How does the model behave out of sample?



Key point: you should randomly sample the training set and the test set

Accuracy

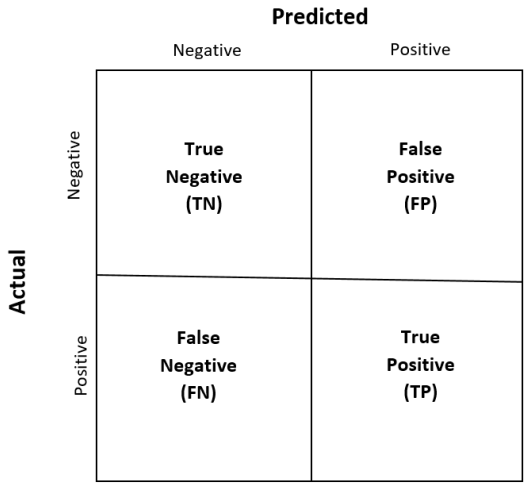
TP = true positive (Correctly classified as Positive)

TN = true negative (Correctly classified as Negative)

FP = false positive (Incorrectly classified as Positive)

FN = false negative (Incorrectly classified as Negative)

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$



An accuracy of 0.5 is no better than random.

Problem?

Doesn't tell us anything about prediction of negatives; can mislead if the two classes are imbalanced (i.e. 90% of the test sample is positive, 10% negative)

Machines Can Outperform Humans in Object Recognition – e.g. Facebook's Tag (but 'good' accuracy depends on existing benchmark)

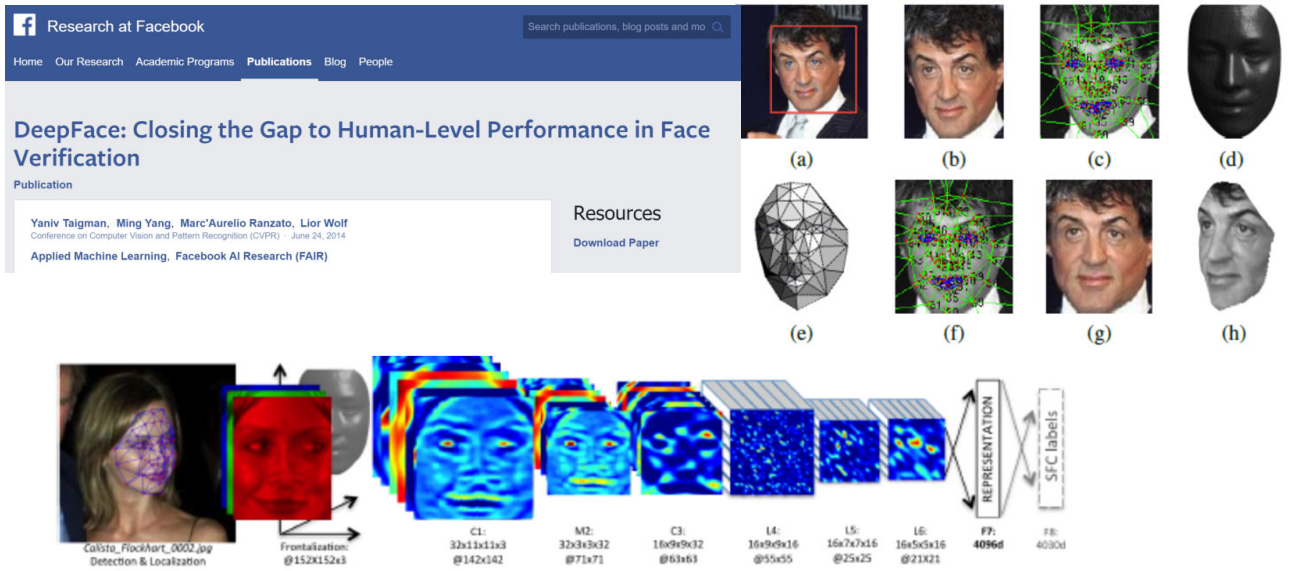
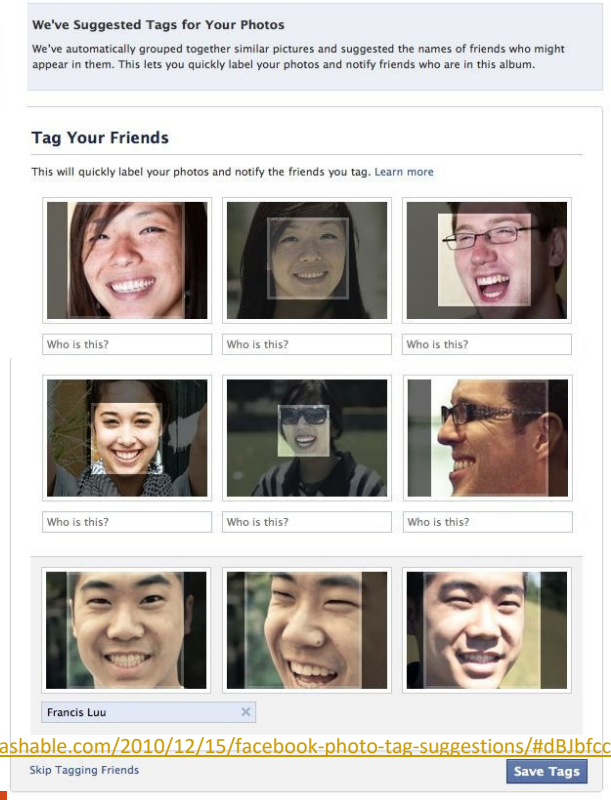


Figure 2. **Outline of the *DeepFace* architecture.** A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/

Facebook's DeepFace facial recognition software has over 97% accuracy



http://mashable.com/2010/12/15/facebook-photo-tag-suggestions/#dBJbfcc3LEqW

What are some Unsupervised learning examples?

Topic Modeling

Assumes words a probabilistic generative process for text (LDA):

1. For every document, draw number of words of document *d* from a Poisson distribution

2. For every document, draw the proportions of the topics for document *d* from a Dirichlet distribution

3. For every word in document d:
    a. Draw the topic the word is assigned to
    b. Draw the word itself

Topic Modeling

Instead of representing documents as vectors of words as in the Vector Space Model research in strategy (Younge and Kuhn 2016, Arts *et al.* 2017), one can determine the topic distribution where each document is a mix of topics (the topics are determined from the corpus of all documents).

Topic Rank	LDA Green Technology Patents Top Topics	LDA All Patents 2009-2012 Top Topics
1	(wind, blade, turbine, fuel, surface)	(data, end, structure, group, level)
2	(gas, side, solar, current, fluid)	(image, data, set, signal, configured)
3	(light, wind, turbine, gas, flow)	(data, memory, circuit, configured, element)
4	(gas, power, wind, heat, turbine)	(data, group, control, signal, member)
5	(power, signal, canceled, cell, control)	(apparatus, member, end, image, configured)
6	(configured, wind, turbine, support, fluid)	(layer, control, group, light, apparatus)
7	(wind, turbine, fuel, configured, air)	(layer, data, user, configured, image)
8	(heat, liquid, stream, gas, wind)	(power, side, control, group, signal)
9	(voltage, energy, engine, fuel, light)	(body, material, value, apparatus, end)

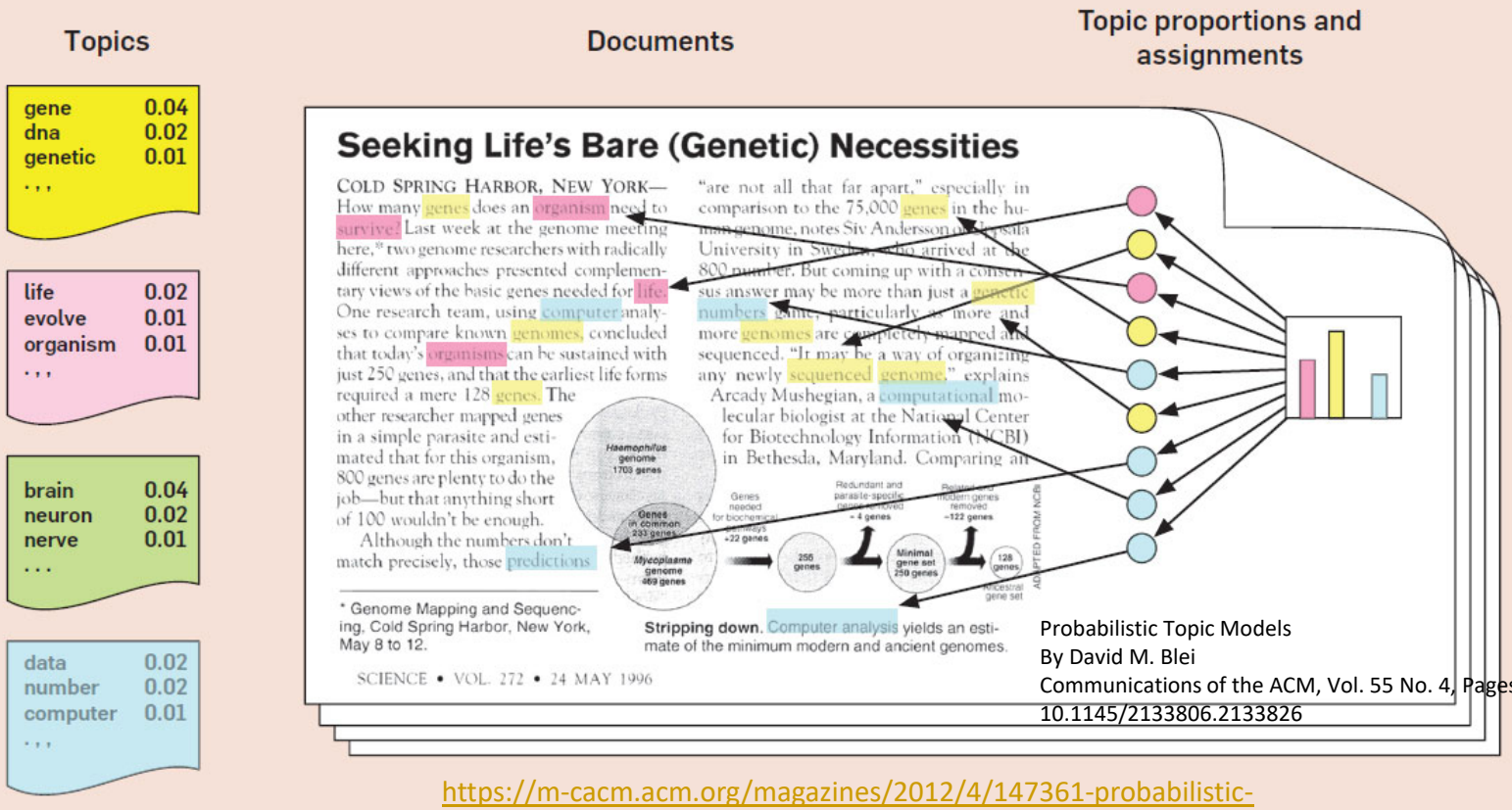
Topic Modeling (see Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), pp.993-1022)

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Fig 8 from Blei et al 2003 above)

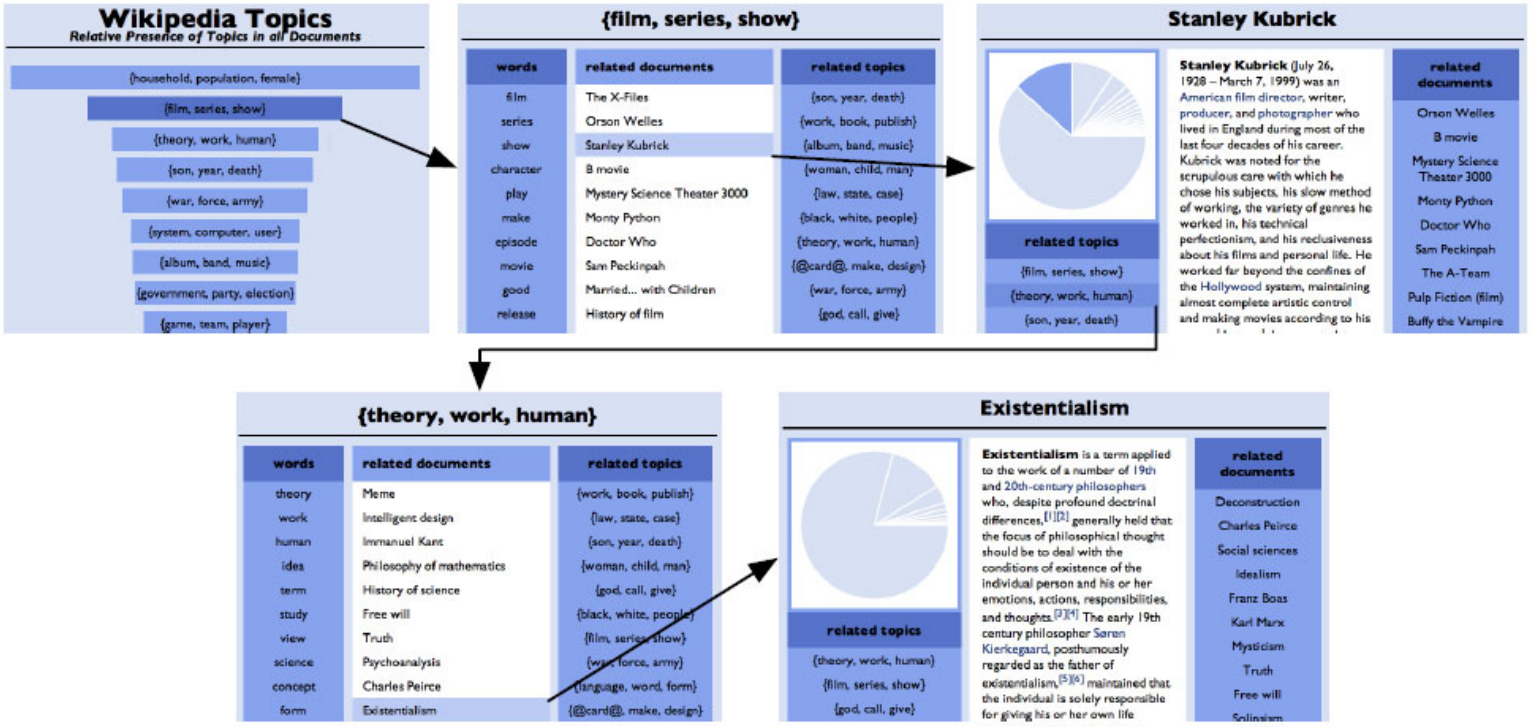
Probabilistic generation process (unsupervised learning)



Probabilistic Topic Models
By David M. Blei
Communications of the ACM, Vol. 55 No. 4, Pages 77-84
10.1145/2133806.2133826

https://m-cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext?mobile=true – Figure 1

Topic Modeling – Wikipedia (from Chaney, A.J.B. and Blei, D.M., 2012, May. Visualizing topic models. In Sixth international AAAI conference on weblogs and social media.)



Clustering v. classification

Picture: Star cluster (NASA, Credit & Copyright: Karel Teuwen)

Clusters are determined in the feature space

The resulted clusters depend on the features you are interested in (selected features) and on the method used

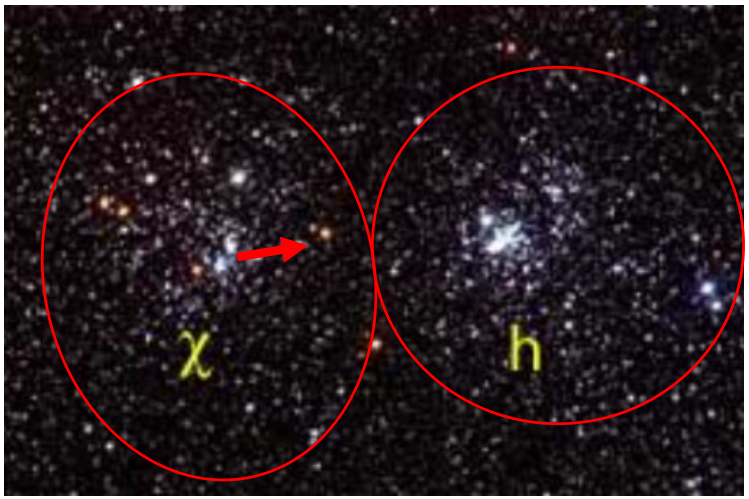
Different clustering methods may lead to very different results

There may be little semantic significance in the classes (partition) if the features are too low-level

The process is unsupervised, therefore no human intervention beyond features and method choice

Simply a mechanical application of an algorithm to find similar datapoints

Visual example: Clustering = grouping cases, objects based on distance or similarity



Credit: N.A.Sharp /NOAO /AURA/NSF (left), Nigel Sharp, Mark Hanna/ NOAO/ AURA/NSF (right)

http://www.atnf.csiro.au/outreach/education/senior/astrophysics/stellarevolution_clusters.html

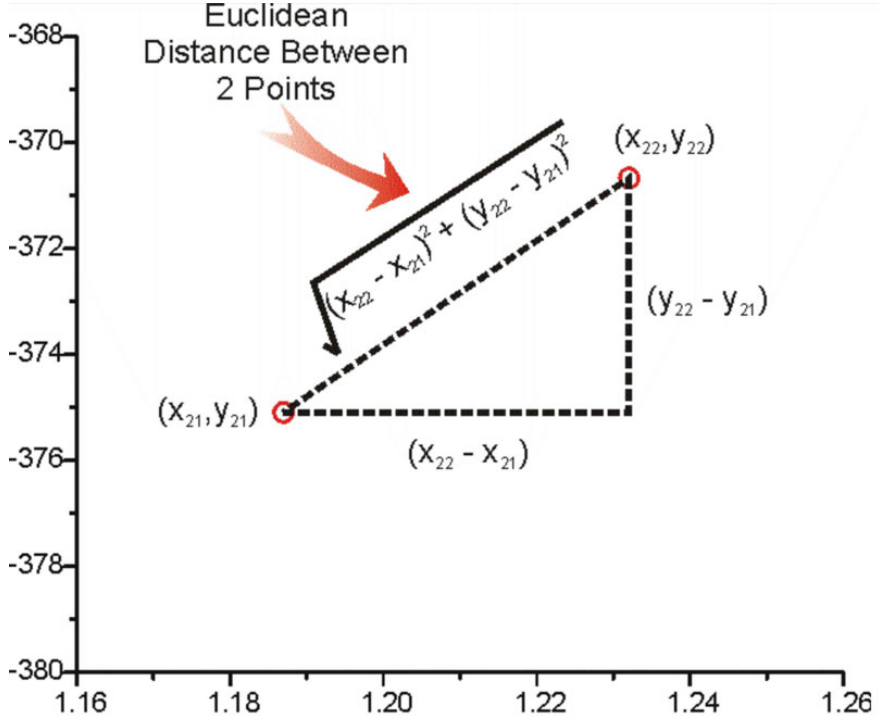
K-means clustering (1)

Why: need to segment dataset

What: assign all datapoints to one of K groups (think of them as “bubbles”, spheres, etc.)

How: Take distances from datapoints to the centers of the clusters & repeat.

Typically using Euclidean Distance (but others possible)



https://hlab.stanford.edu/brian/making7.gif

K-means clustering (2) – How it works

IN: your dataset (N datapoints), which attributes (i.e. columns) you care about, how many clusters you want to find (K is the number of clusters) and select a distance (Euclidean works)

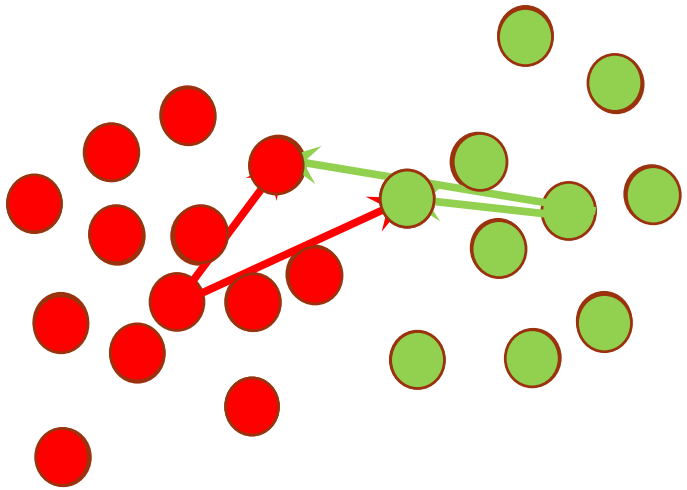
Step 1: Place a "centroid" (center of cluster – think of it as the center of a sphere) at random for each of the K clusters

Step 2: REPEAT the following

- For each datapoint, find closest centroid (based on distance to centroid) & assign the datapoint to your cluster
- ONCE DONE assigning to clusters: **for each cluster, recompute center of cluster** (take the mean of all datapoints in that cluster). That is the new centroid

STOP: stop when step 2 fails to produce any changes in assignments of datapoints to clusters

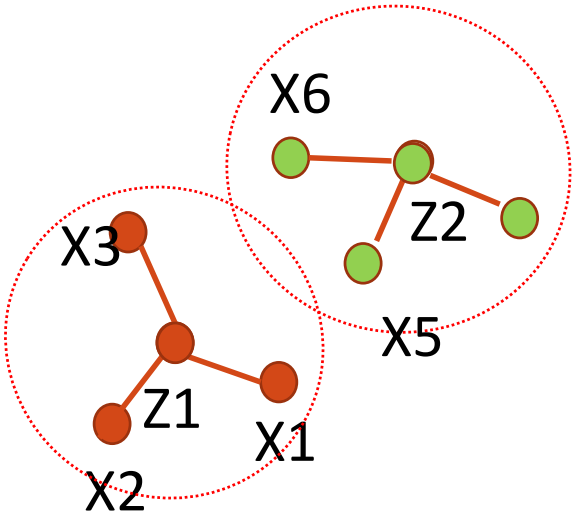
K-means Clustering – Animation Example



1. Randomly Pick Centroids
2. Calculate distances
3. Assign points to clusters

Let’s talk about Classification – similar method (K-N-N)

Classifiers: Majority Vote (K-Nearest-Neighbors or "K-N-N")



Concept: 'neighbors' belong to the same cluster;

For a new datapoint 'Z1', determine its 'k' nearest neighbors; 'Z1' belongs to the cluster 'red' if most of its closest k neighbors belong to the cluster 'red' (in this case, closest k based on distance are X1, X2, X3, all red)

In this example, K=3 so we look at the three closest neighbors

Democracy works

K-N-N Algorithm Steps

Notations: Xi = feature vector for datapoint *i;* Yi = label (or "class") for datapoint *i*; K = number of neighbors we will consider.

INPUT: Training examples – a set of pairs {(Xi, Yi)} where we know the labels/classification for these datapoints; also must choose d – the distance function based on which we determine neighbors

Z is a new datapoint (to be classified)

STEPS:

1. Compute d(Z,Xi) between the new datapoint and *all* existing labeled points
2. Select K closest points to Z, including their labels {set of K pairs (Xi, Yi)}
3. Majority Vote: Pick the most frequent class in {Yi} and assign that label to Z and call it Yz

Popular

Handwriting Recogn

http://cs-people.bu.edu/athitsos

http://www.spam.com/upload/varieties-images/Spam-Teriyaki.png

nalysis

know!"

loads/2014/07/Spam-3.jpg

Choices of Parameters

The value of K (# of neighbors to check)

The distance function

What is a distance?

Definition: Consider a set *X* and an application *d* defined on $d: X \times X \to \mathbb{R}_+$ satisfying:

$$\begin{aligned}
&\text{i).}\ \ d(x,x) = 0 \\
&\text{ii).}\ \ d(x,y) = d(y,x) \\
&\text{iii).}\ \ d(x,y) = 0 \Leftrightarrow x = y \\
&\text{iv).}\ \ d(x,z) \leq d(x,y) + d(y,z).
\end{aligned}$$

Any application *d* satisfying the definition's i)-iv) is a distance metric. If the triangle inequality iv) is not satisfied, *d* is a pseudo-metric.

Euclidean and Minkovski Distances N dimensional space

- The Euclidean distance in N dimensions is:

$$d(v_j, v_k) = \sqrt{\sum_{i=1}^{n} (\mid v_j[i] - v_k[i] \mid)^2} (Euclidean)$$

- Its generalization, the Minkowski distance is:

$$d(v_j, v_k) = \left[\sum_{i=1}^{n} \left(|v_j[i] - v_k[i]|^t \right) \right]^{1/t} (Minkovski)$$

(note t = 2 → Euclidean; t < 1 not a distance metric as it violates triangle inequality; and as $t \longrightarrow \infty$ this becomes the Chebyshev

Chebyshev (chessboard) Distance

$$d(v_j, v_k) = \sum_{i=1}^{n} \left(max(|v_j[i] - v_k[i]|) \right) (Chebyshev)$$

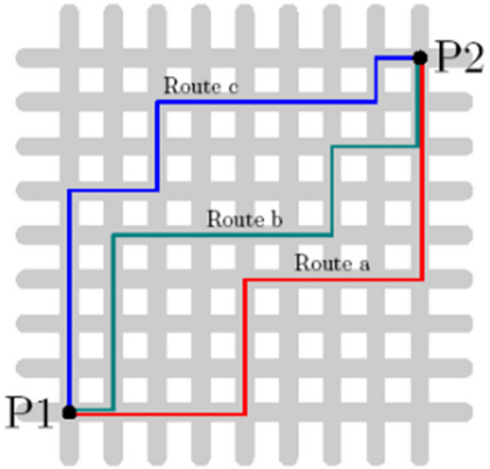
	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	♔	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

Maximum distance on any axis

Manhattan (taxicab) Distance

- Sum of absolute distances in coordinates, also called taxicab geometry based on how many blocks it takes from 1 to 2 (driving a taxi in Manhattan)

$$d(v_j, v_k) = \sum_{i=1}^{n} (|v_j[i] - v_k[i]|)(Manhattan)$$



Canberra distance

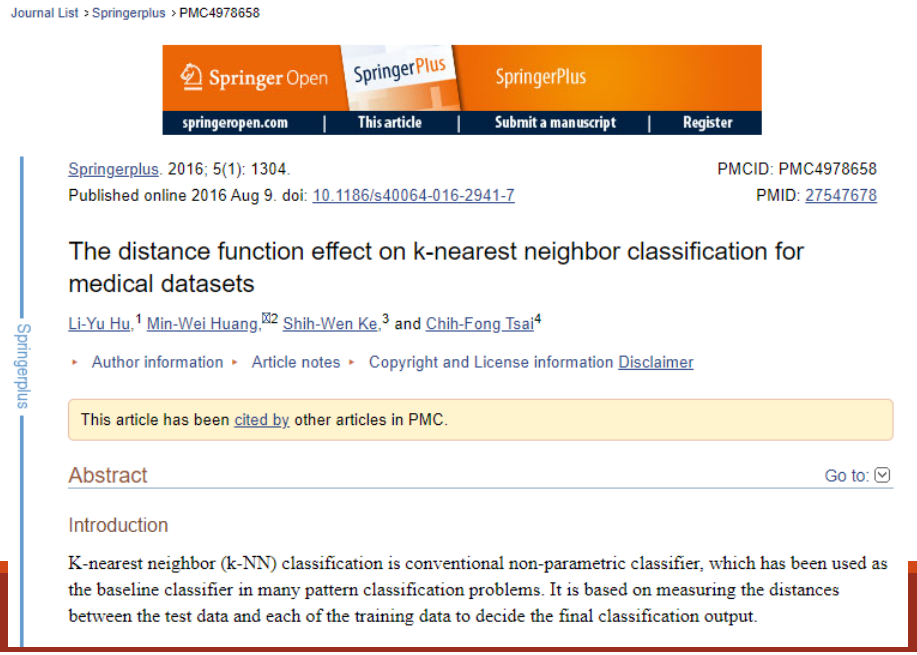
- Practical application is cybersecurity

$$d(v_j, v_k) = \sum_{i=1}^{n} \left(\left| \frac{v_j[i] - v_k[i]}{v_j[i] + v_k[i]} \right| \right) (Canberra)$$

- Somewhat generalizes Manhattan

Wrap-Up K-N-N Distances

- Different distances may be used for different applications – the choice of distance is also a possible parameter in K-N-N in addition to the choice of K and can yield very different results
- If interested in more read: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4978658/



Journal List > Springerplus > PMC4978658

Springer Open | SpringerPlus | SpringerPlus

springeropen.com | This article | Submit a manuscript | Register

Springerplus. 2016; 5(1): 1304. PMCID: PMC4978658

Published online 2016 Aug 9. doi: 10.1186/s40064-016-2941-7 PMID: 27547678

The distance function effect on k-nearest neighbor classification for medical datasets

Li-Yu Hu,[1] Min-Wei Huang,[✉2] Shih-Wen Ke,[3] and Chih-Fong Tsai[4]

Author information · Article notes · Copyright and License information Disclaimer

This article has been cited by other articles in PMC.

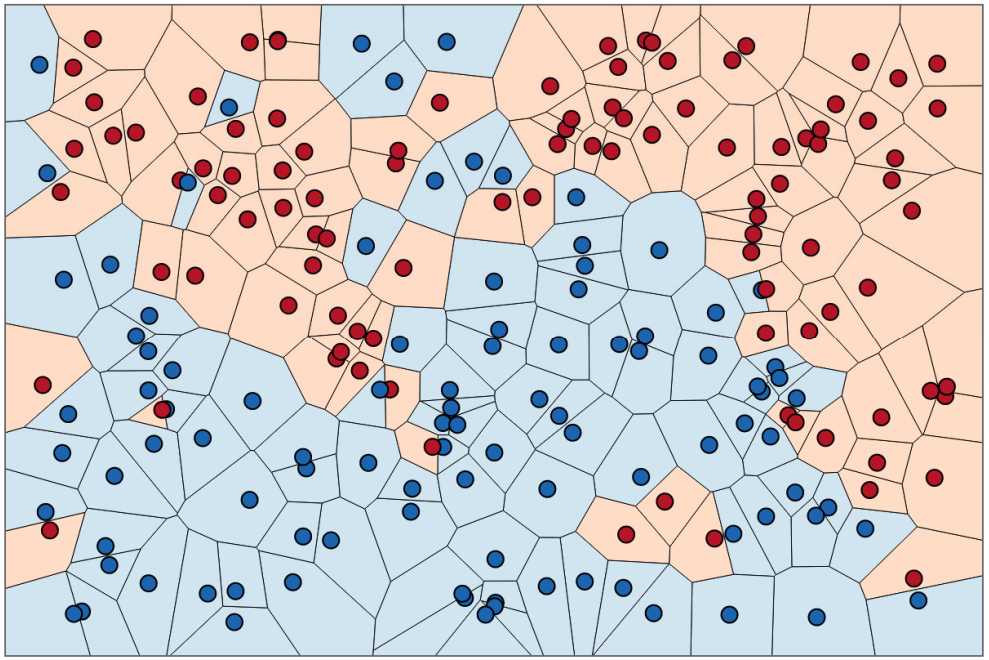
Abstract Go to:

Introduction

K-nearest neighbor (k-NN) classification is conventional non-parametric classifier, which has been used as the baseline classifier in many pattern classification problems. It is based on measuring the distances between the test data and each of the training data to decide the final classification output.

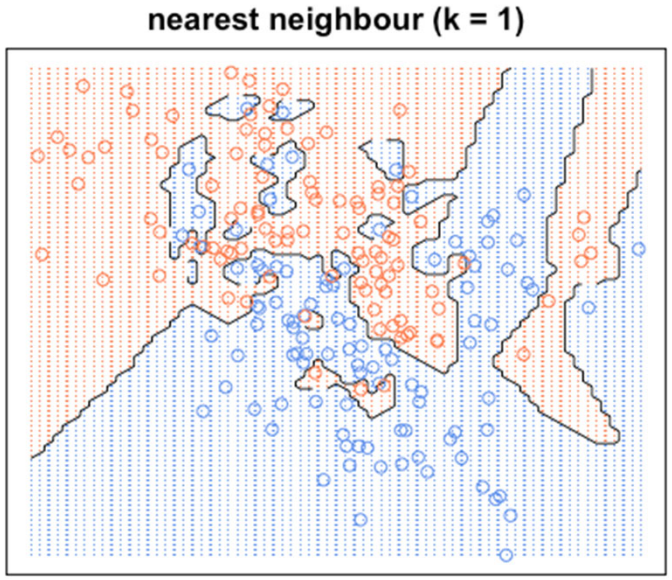
What does the decision boundary look like? (why does K matter?)

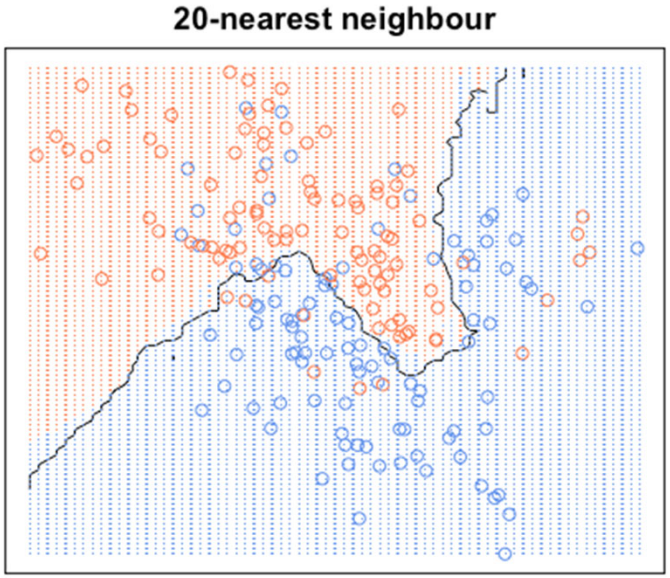
Voronoi Partitions (K=1)



https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Decision Boundary – K-N-N (depends on choice of K)

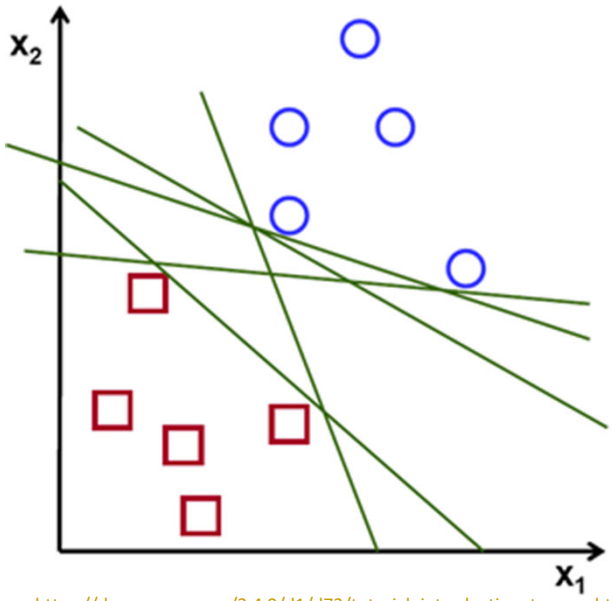




https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Support Vector Machines

Support Vector Machines find the elements on the **boundary** between the classes and finds the hyperplane which separates those classes. **Which one of the following is the best?**



https://docs.opencv.org/3.4.0/d1/d73/tutorial_introduction_to_svm.html

Support Vector Machines (2)

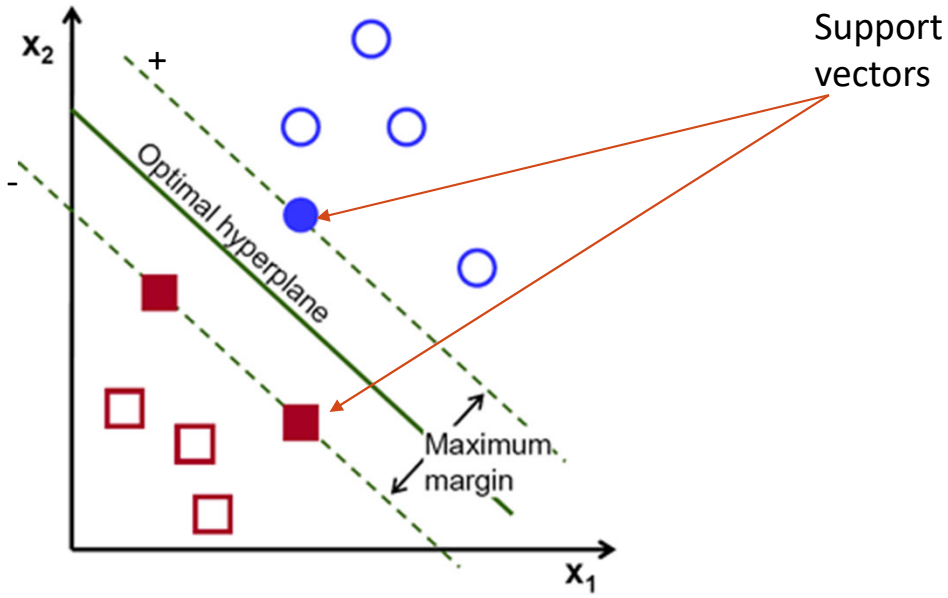
Support Vector Machines (SVMs) find the boundary which maximizes the margins between the two classes:

The optimal hyperplane has the form:

$$w^T x + b = 0$$

x = predictors (input),

w = weights, b = bias



https://docs.opencv.org/3.4.0/d1/d73/tutorial_introduction_to_svm.html

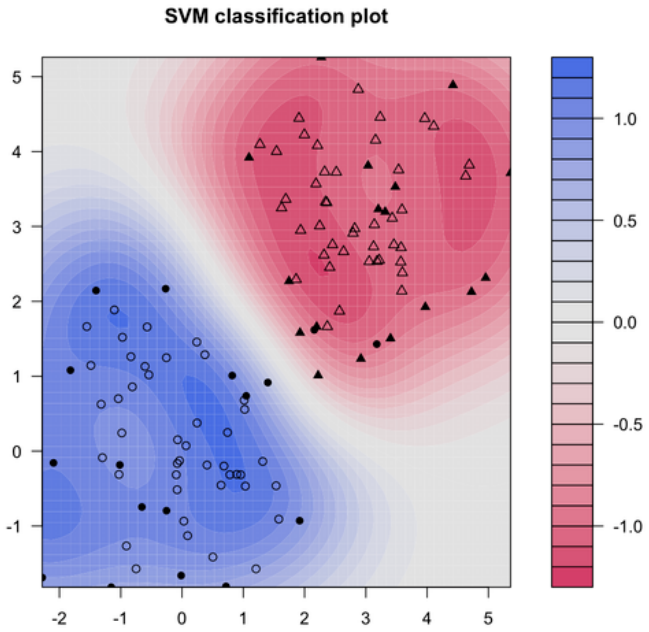
Support Vector Machines (3)

- For data which is not linearly separable, the standard SVM implementation is to find the best balance between misclassification error (points coming across the boundary from the opposite class) and width of the margin between classes.
- The decision is an optimization based on (typically) a small subset of the data, those points from each class that are right at the boundary – a.k.a. “support vectors”
- If data is not linearly separable, then you can use a kernel function to transform the data a higher dimensional space where you are able to better separate the data

Support Vector Machines (4)

- Newer method than decision trees (1990s)
- More computationally expensive (generally) than a decision tree approach
- Has had numerous applications in pattern recognition, machine fault analysis, optical text recognition, financial analysis, medical imaging and more
- Relatively faster than a “lazy” learner like K-N-N

Decision Boundary – SVM



https://stats.stackexchange.com/questions/5056/computing-the-decision-boundary-of-a-linear-svm-model

SVM separates the classes by constructing a hyperplane

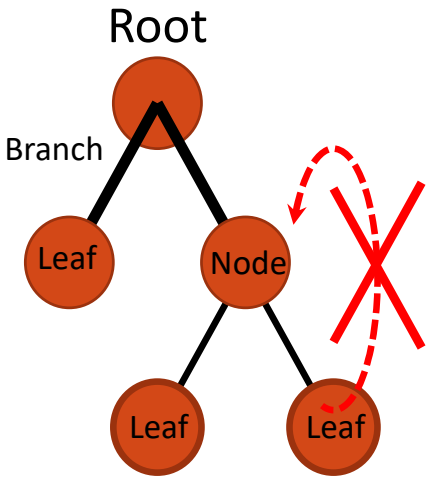
Naïve-Bayes Approach

- Naïve as it assumes the features to be independent of one another (silly in many cases, like wages being independent of worker quality, or college independent of earning potential)
- Given a Class C_k, and a set of features xi, the probability of belonging to the Class conditional on the predictors is:

$$P(C_k|x_1, x_2, \ldots, x_n) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)}{\prod_{i=1}^{n} P(x_i)}$$

- Label based on the highest computed value ($\hat{Y} = argmax(P(Y = C_k)\prod P(x|Y = C_k))$
- Advantages: very fast, often used in real time prediction; recommender systems; spam detection; sentiment analysis (works well enough for document classification because the independence of features assumption can hold there depending on context).

An Intuitive Algorithm: Decision Trees



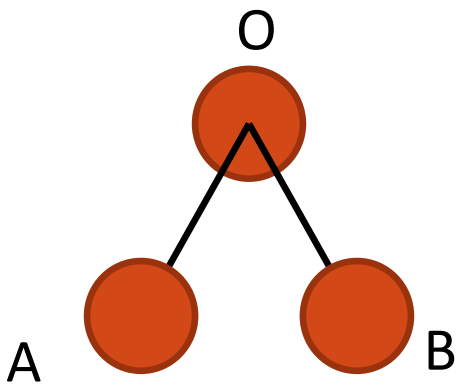
A tree is a mathematical graph construct resembling the natural tree; typically drawn downwards

It has a ‘starting node’, named “root”, from which branches divide; at some point (node), branches subdivide up to the leaves which are the ‘most distant nodes”

No branch comes backward ! There is no loop in trees

From a node, 2 (binary tree) or more branches can start.

Decision Trees (2)



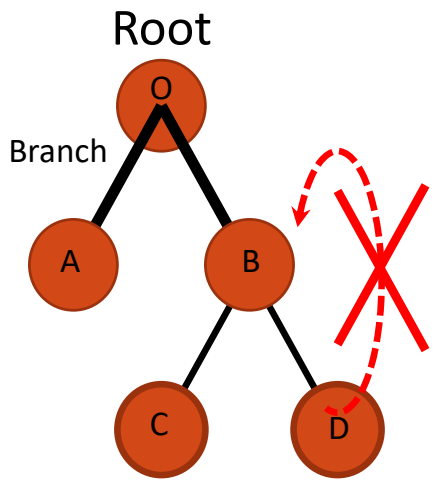
Consider two classes, A and B and an object that can belong to one of these classes. When we classify an object O as being from category (class) A, we make a decision. Thus, the distinction between decision and classification is utilitarian only.

A decision O→A or O →B can be graphically represented by a branching where the two **choices have to be mutually exclusive**

The path from the root to the leaf (end node) represents the decision path which leads to an outcome

Each branching node is called a "decision node". Each end node is a "leaf"

Decision Trees (3)



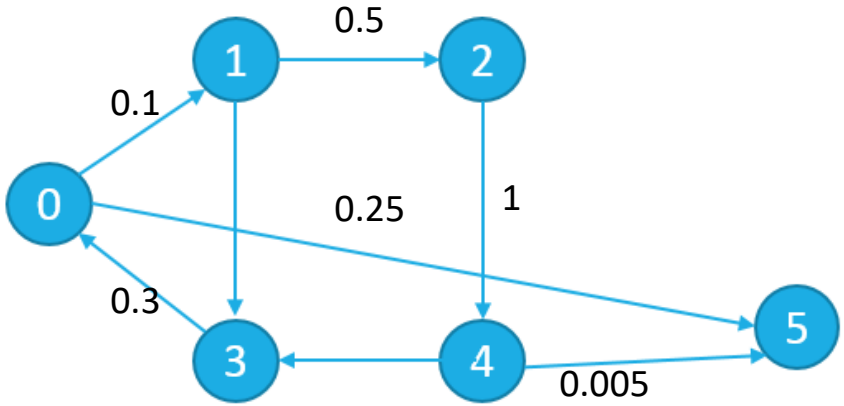
O is the root of the tree (top most decision)

B is a "child node of O" (an outcome resulting from **splitting the dataset based on the decision at O)**

C is a terminal node/end node/ leaf node and represents a subset of the data based on the choices made at branching nodes O and B

Branches (also called "edges") connect nodes. They are useful in helping us understand paths to a decision

A Tree is a special case of a Graph

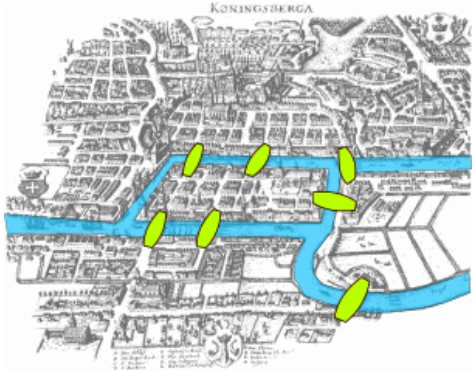


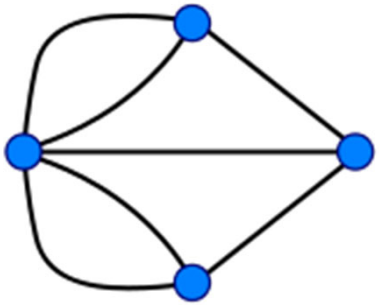
Definition: A graph is a pair G = (V,E) where V is the set of Nodes (vertices) and E is the set of edges (pairs of nodes (i,j) that are connected). Edges directed or undirected

- A node can represent a person, website, organization, city, etc.
- Edges can be used to represent connections between people, including a weight measure for strength of connection
- An edge ***can have a direction*** (as in the case of measuring a *network flow – like traffic around a city*) or be ***undirected***

Earliest Days of Graphs: Eulerian Path

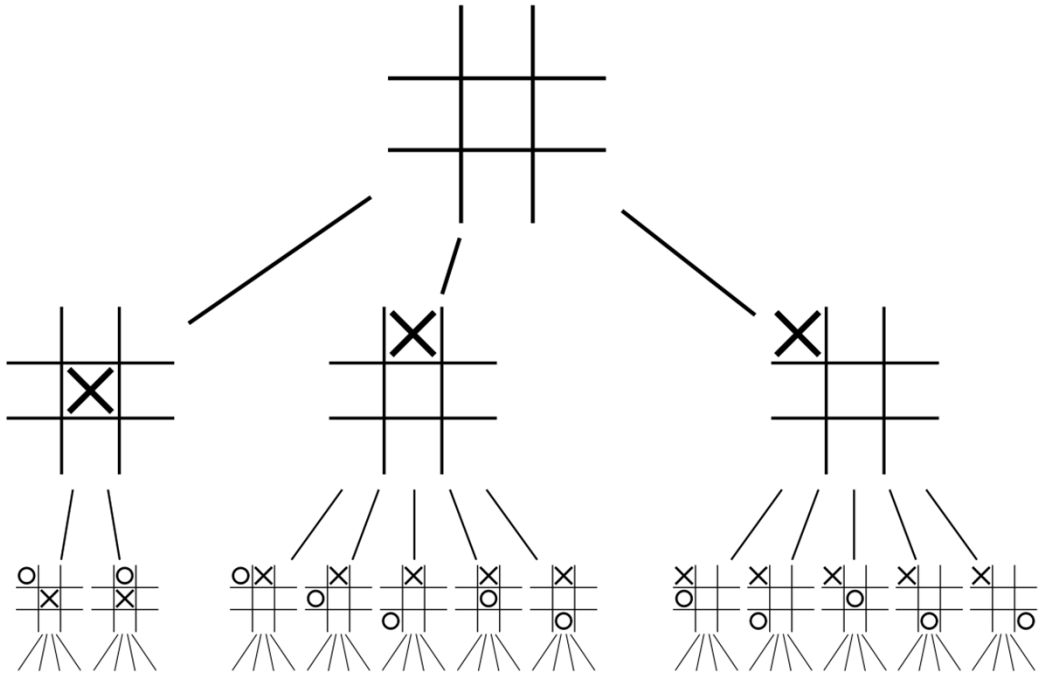
- Konigsberg problem (Prussia) – can you walk over the Pregel river and cross every bridge exactly once?
- Euler's theorem: An undirected graph contains an Euler cycle iff it is connected and all vertices are of even degree.



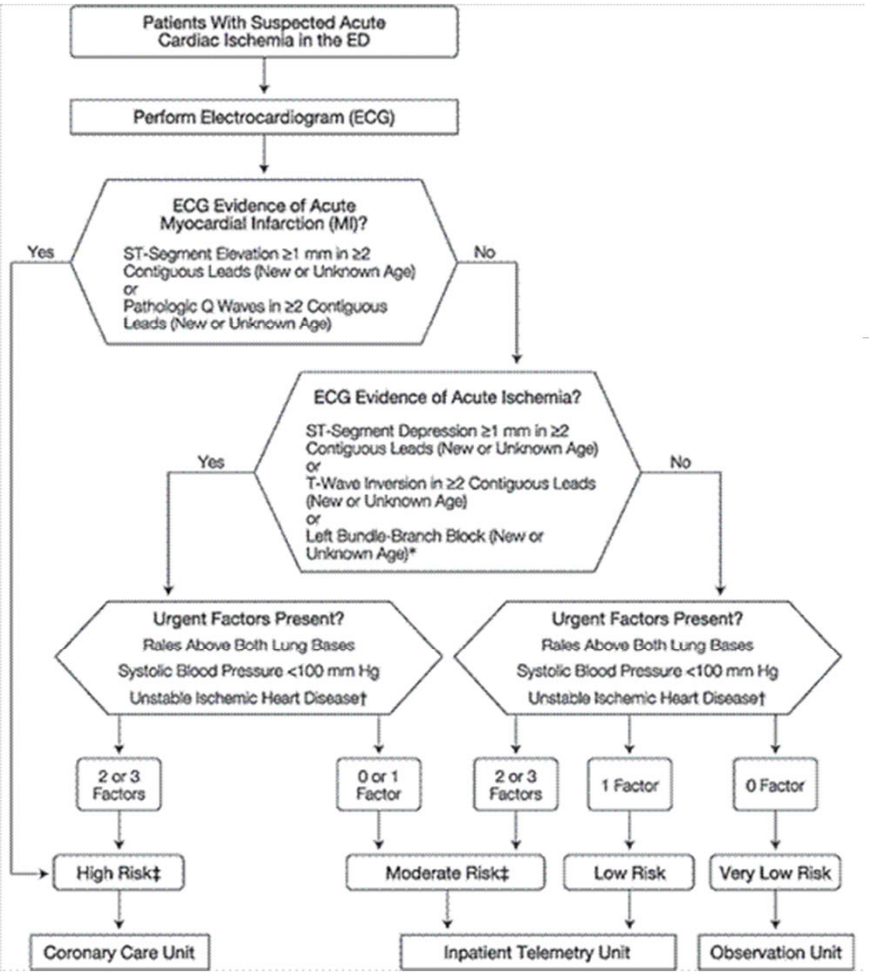


Images from http://en.wikipedia.org/wiki/Eulerian_path#mediaviewer/File:Konigsburg_graph.svg, http://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg#mediaviewer/File:Konigsberg_bridges.png

We often use Decision Trees Intuitively:



https://www.analyticsvidhya.com/blog/2019/01/monte-carlo-tree-search-introduction-algorithm-deepmind-alphago/1200px-tic-tac-toe-game-tree-svg_/



Decision Trees in Medicine

Doctors often use decision tree heuristics to quickly treat or triage conditions

Some of these are automated in question-answering systems (i.e. phone apps or computer apps) which replace the old paper questionnaires

Doctors are trained to think in terms of such heuristics

Many clinical papers feature better clinical decision trees

(see JAMA, NEJM, Annals of Emergency Medicine, etc.)

Figure from https://jamanetwork.com/journals/jama/fullarticle/195118

Decision Trees – Splitting and Pruning

Splitting = the process by which we divide a node into two (or more) subsets. This is based on a decision

Pruning = removing a node from two or more subsets (opposite of splitting)

While you will not implement the actual algorithm (we call an R command which already implements this), why is pruning useful?

Preventing overfitting

Some + and – of Decision trees

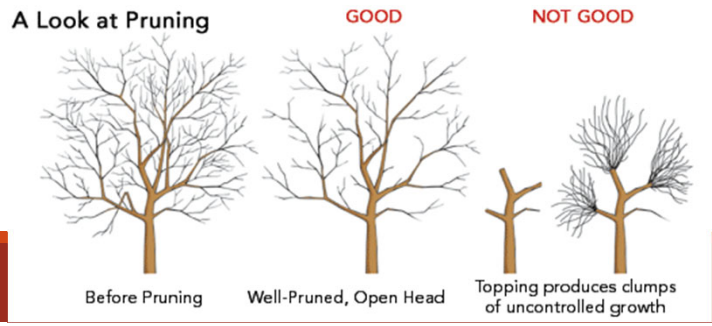
Positives:

- Easy to visualize and understand the phenomenon (many of us learn through visuals, this is an easy algorithm to visualize)
- Easy to prioritize: just pick the nodes from the root down and you know you're addressing the key points of variation
- Easy to see decisions and outcomes as well as paths to outcomes

Drawbacks:

- Prone to overfitting (more on solutions later)
- Small changes propagate throughout the tree (e.g. a change at one node affects all decisions – splits – under that node so one change can result in very different trees)
- Paralysis by analysis: in some cases the tree can become overly complex and you have to choose **to prune it** and what to ignore

https://forestrypedia.com/general-guidelines-on-tree-pruning/



Gentle Math (1)

- Decision trees are built top down (from the root down)
- **How does the algorithm decide which attribute to split on first?**
- **What about second split, third split, etc?**
- **It has to do with the purity of the node**

"Impure" node

The node does not contain only elements of one class

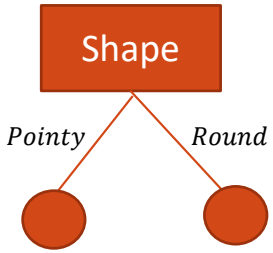
"Pure" node

Desirable: one leaf contains one class (no ambiguity)

Gentle Math (2)

Definition: A pure node contains only elements of one class.

A perfect split between two classes yields all leaf nodes to be pure:



Pure nodes mean that the decision paths we took to the terminal (i.e. leaf) nodes yielded no misclassified data. This is what we want the algorithm to do.

A stop criterion for the algorithm could be: all leaf nodes are pure.

Gentle Math (3)

While all nodes being pure means that we have perfectly split the data into subsets, this could also mean that we took so many splits (the depth of the tree is so large) that in the end we have leaf nodes with just one element in each node. This is **overfitting the data** and will result in very poor prediction performance on new data.

Sounds like something we may not want…

We need somewhat more complex criteria based on which to optimize the tree.

Purity is a measure of certainty: for example, a subset with half and half of two categories would be impure and we would know that there is missing information

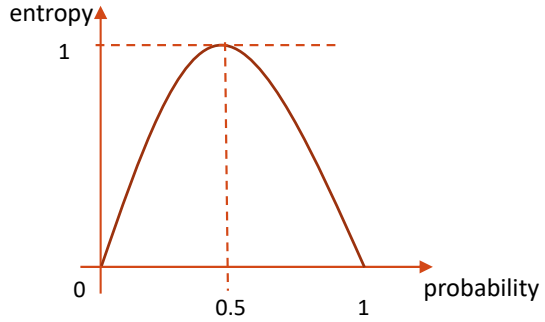
Gentle Math (4)

In physics, entropy is the degree of randomness in the system, or lack of information about the system.

Definition: Entropy is the measurement of the degree of missing information in a decision.

In other words, entropy measures the degree of impurity in each split.

We cannot say anything about a subset with 50%-50% split being different than random – therefore it has the highest entropy (of 1). However, a split of 100%-0% or 0%-100% is pure and we can say it has no entropy (entropy of 0).



Gentle Math (5)

$Entropy = -\sum p(x) \cdot \log(p(x))$

Where $p(x)$ is the fraction of examples of a given class within the node

Pure nodes have an entropy of 0

How does the Decision Tree Optimize?

Minimize the impurity at each split.

This can be done by comparing the *Entropy of the parent node* to the *Entropy of the nodes resulting from the split.*

This method is called **Information Gain:**

$$\boldsymbol{Information\ Gain = Entropy(parent) - Weighted\ Average\ (Entropy(children))}$$

The algorithm always picks the split with **the highest information gain at each step** thus it is a **greedy algorithm** (picks locally optimal choice at each step).

(You can read more about this in the Machine Learning for Strategy Research working paper)

More advanced details (for your own reading)

- Ending algorithm when all nodes are pure can work, though may not be ideal (overfitting)
- There are a variety of decision tree algorithms based on the stop criterion:
  - Stop when you reach a maximum depth of the tree you set (you pick max depth)
  - Stop when you reach a minimum level of information gain (you pick the min information gain)
  - Stop when a subset contains fewer than m datapoints (you pick m)
- If you are curious, here are some decision tree implementations:
  - ID3 (oldest, straightforward, does not prune, cannot handle missing values)
  - C4.5
  - CART
  - CHAID

So how well do these three approaches separate classes? What does the boundary look like?

Decision Boundary

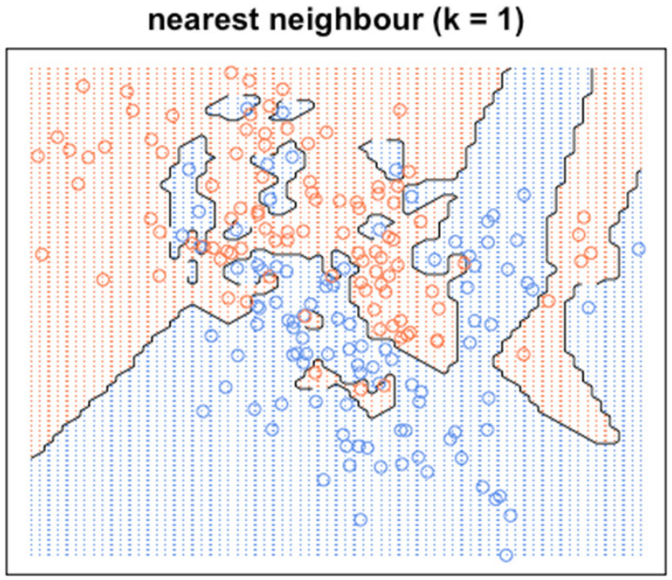
Sometimes you need to look at the split of the classes within your data to pick the best classifier.

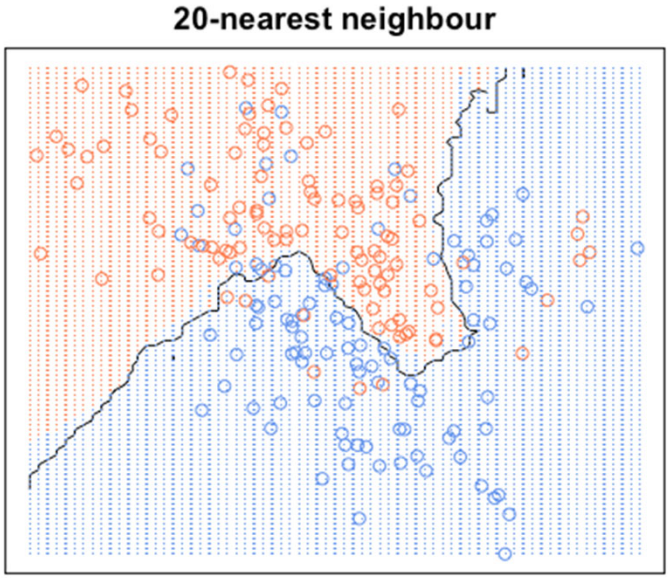
We've seen linearly separable data, but what if it's like this?

Elkanah Tisdale (1771-1835) Boston *Centinel*, 1812.

From Wikipedia: "The district depicted in the cartoon was created by Massachusetts legislature to favor the incumbent Democratic-Republican party candidates of Governor Elbridge Gerry over the Federalists in 1812."

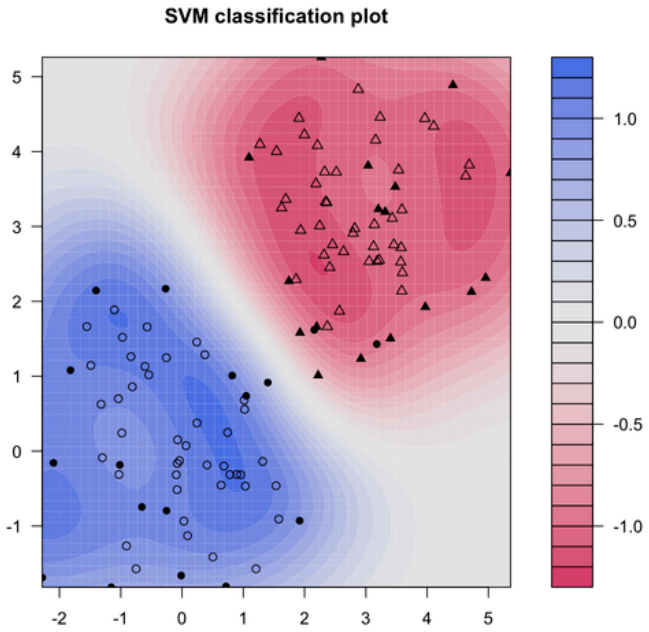
Decision Boundary – K-N-N (depends on choice of K)





https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

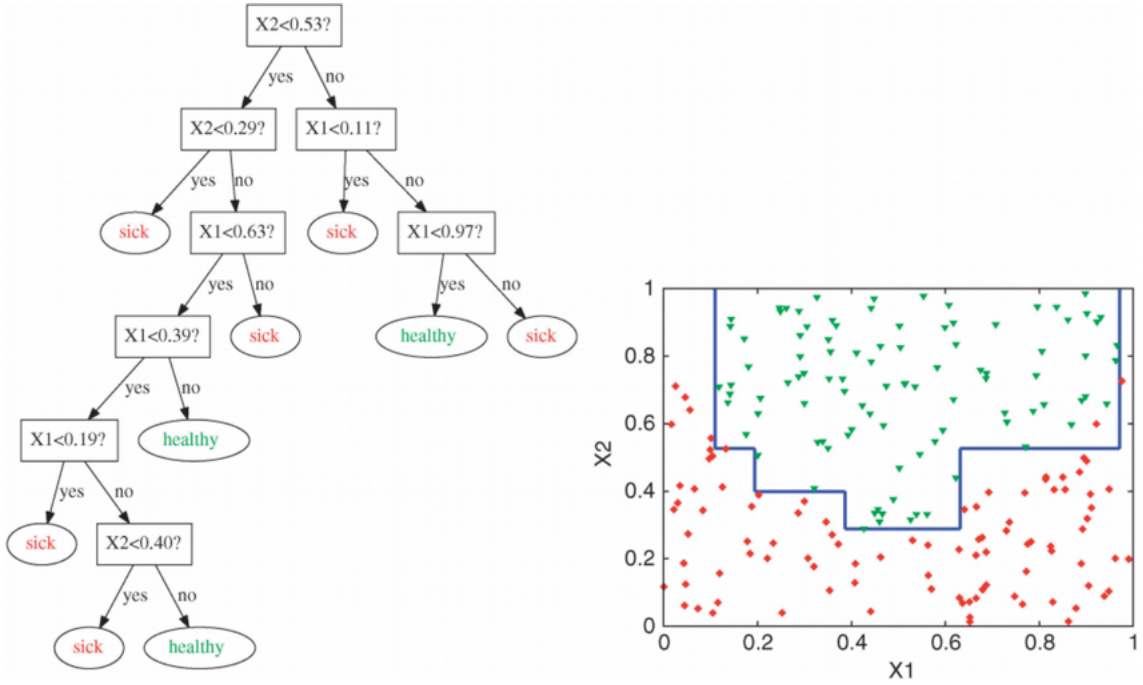
Decision Boundary – SVM



https://stats.stackexchange.com/questions/5056/computing-the-decision-boundary-of-a-linear-svm-model

Hint: SVM separates the classes by constructing a hyperplane

Decision Boundary - Tree



https://www.researchgate.net/figure/An-example-of-a-decision-tree-left-with-the-decision-boundary-for-two-features-X1_fig5_313720565

Cross-Validation

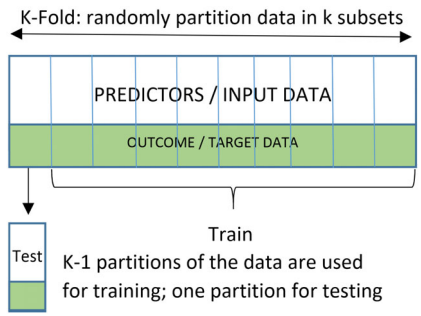
- **How do you pick the value of *K* in *K-N-N* (K-Nearest-Neighbors?)**

- **How do you pick the size of the node for the decision tree?**

- **Whenever there are parameters for a model, we can either pick them ourselves or let another algorithm pick the best parameters. Cross-Validation comes into play**

Cross-Validation

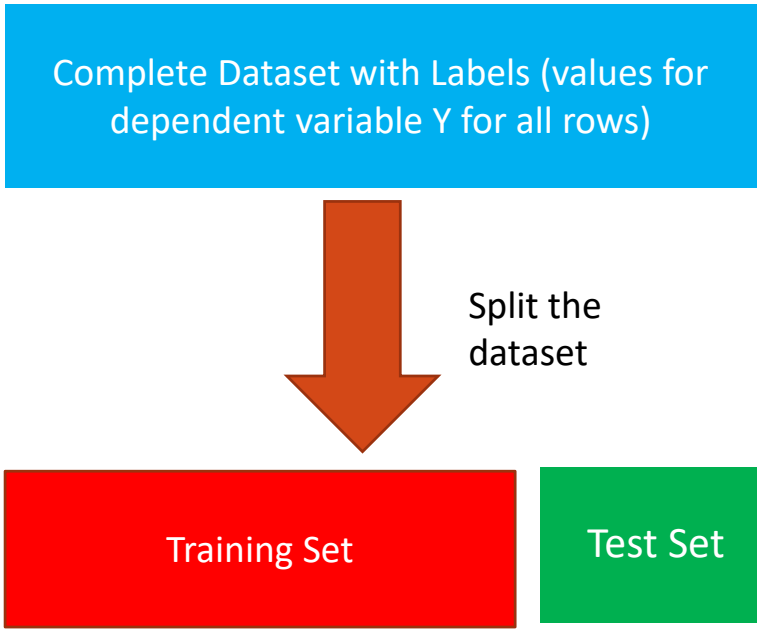
Cross-validation involves measuring algorithm accuracy and comparing different runs of the algorithm across the various splits of the data to optimize model parameters.



The k-fold cross-validation approach is the typical one in most applications:

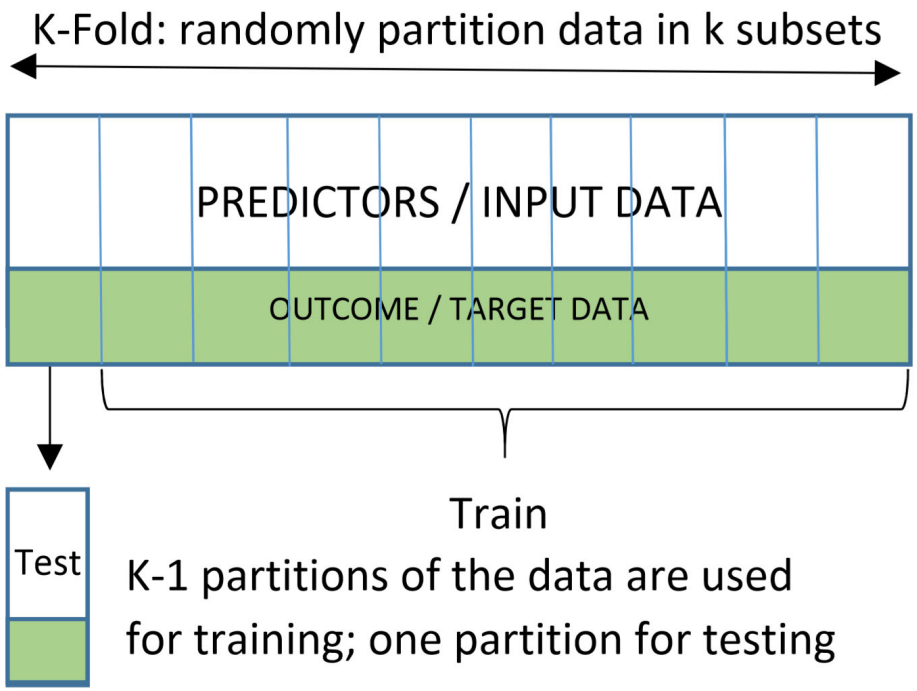
1. Partition randomly data into k mutually-exclusive subsets;
2. Run your model k times, with each run on a different set of k-1 subsets joined as a training set and with testing done on the remaining subset;
1. the k runs produce k different parameter sets for the algorithm, and the classification performanc es of these runs can be compared to each other.
2. K is normally selected to be at least ten

This essentially repeats the process we did last week for one train+test pair



Key point: you should randomly sample the training set and the test set

Cross-Validation



The CARET Package in R

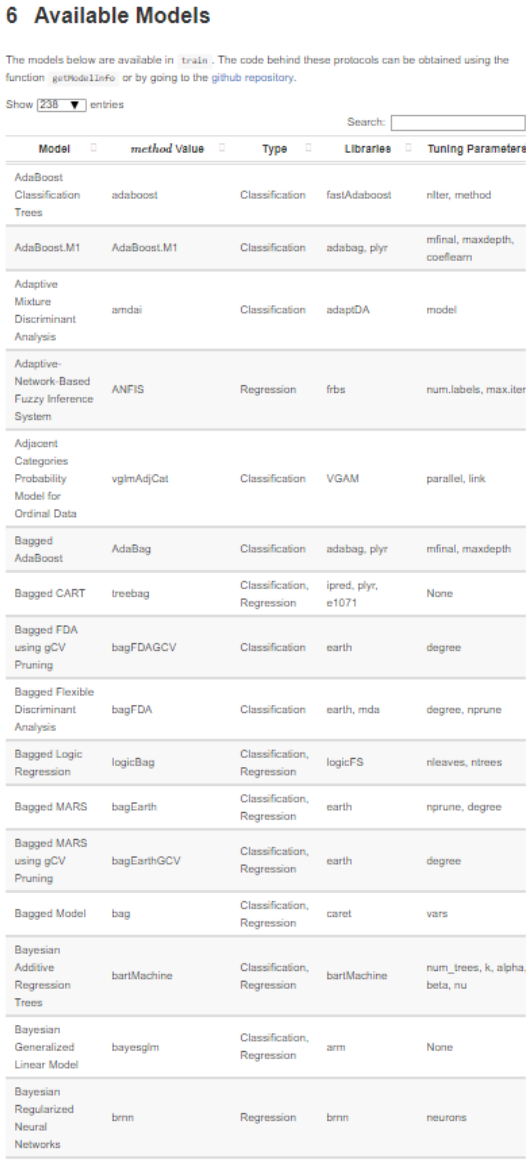
Available at:

http://topepo.github.io/caret/index.html

Huge list of models! All trainable by cross-validation.

Available at:

https://topepo.github.io/caret/available-models.html



6 Available Models

The models below are available in `train`. The code behind these protocols can be obtained using the function `getModelInfo` or by going to the github repository.

Show 238 entries

Search:

Model	*method* Value	Type	Libraries	Tuning Parameters
AdaBoost Classification Trees	adaboost	Classification	fastAdaboost	nIter, method
AdaBoost.M1	AdaBoost.M1	Classification	adabag, plyr	mfinal, maxdepth, coeflearn
Adaptive Mixture Discriminant Analysis	amdai	Classification	adaptDA	model
Adaptive-Network-Based Fuzzy Inference System	ANFIS	Regression	frbs	num.labels, max.iter
Adjacent Categories Probability Model for Ordinal Data	vglmAdjCat	Classification	VGAM	parallel, link
Bagged AdaBoost	AdaBag	Classification	adabag, plyr	mfinal, maxdepth
Bagged CART	treebag	Classification, Regression	ipred, plyr, e1071	None
Bagged FDA using gCV Pruning	bagFDAGCV	Classification	earth	degree
Bagged Flexible Discriminant Analysis	bagFDA	Classification	earth, mda	degree, nprune
Bagged Logic Regression	logicBag	Classification, Regression	logicFS	nleaves, ntrees
Bagged MARS	bagEarth	Classification, Regression	earth	nprune, degree
Bagged MARS using gCV Pruning	bagEarthGCV	Classification, Regression	earth	degree
Bagged Model	bag	Classification, Regression	caret	vars
Bayesian Additive Regression Trees	bartMachine	Classification, Regression	bartMachine	num_trees, k, alpha, beta, nu
Bayesian Generalized Linear Model	bayesglm	Classification, Regression	arm	None
Bayesian Regularized Neural Networks	brnn	Regression	brnn	neurons

What could we parametrize in a cross-validation setting – for the decision tree?

What about combining the voting method from K-N-N with the decision tree?

Accuracy

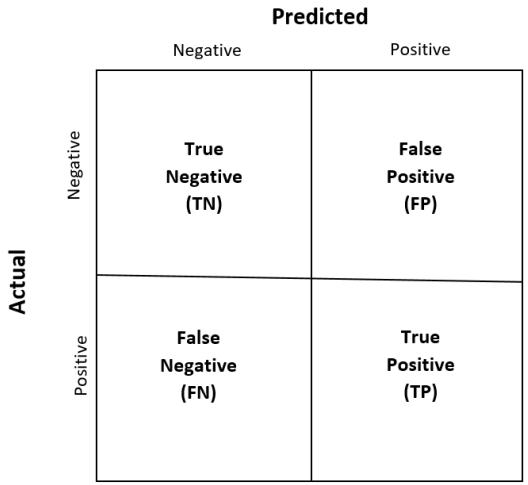
TP = true positive (Correctly classified as Positive)

TN = true negative (Correctly classified as Negative)

FP = false positive (Incorrectly classified as Positive)

FN = false negative (Incorrectly classified as Negative)

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$



An accuracy of 0.5 is no better than random.

Problem?

Doesn't tell us anything about prediction of negatives; can mislead if the two classes are imbalanced (i.e. 90% of the test sample is positive, 10% negative)

Receiver Operating Curve (ROC)

Closer to top left corner: better prediction (perfect True Positives, no False Positives)
Sensitivity =

Exploring Fairness in Machine Learning for International Development - MIT D-Lab | CITE

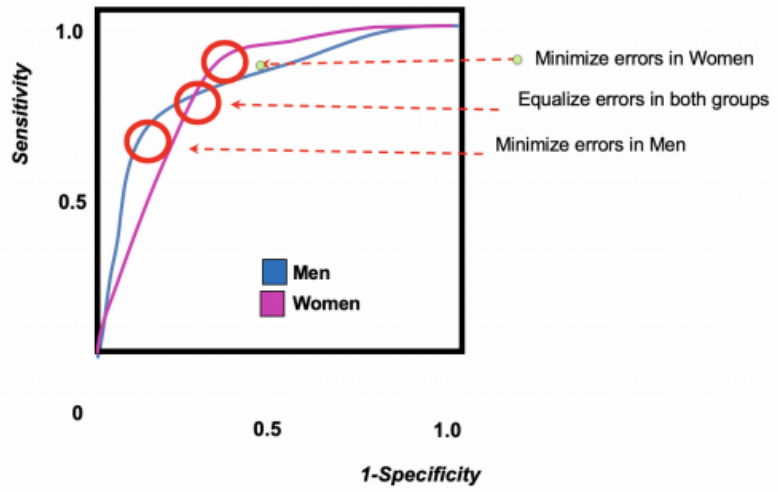


Figure 2 - Sample ROC curve for multiple groups, showing different possible operating points for the algorithm

The diagonal represents a random classifier (50-50 guessing)

$Sensitivity = Recall = TPR = \frac{TP}{TP+FN}$ *(how many relevant positives were missed? Important if FN cost is high)*

$Specificity = TNR = \frac{TN}{TN+FP}$ *(how many of the predicted false are actually false?)*

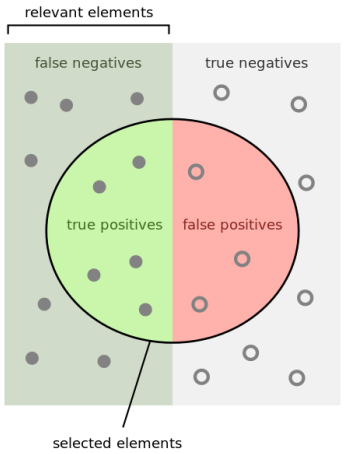
Precision, Recall, F-1 measure

$Precision = \frac{TP}{TP+FP}$ (aka PPV = positive predictive value)

$Recall = \frac{TP}{TP+FN}$ (aka Sensitivity)

$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} = \frac{TP}{TP+\frac{FN+FP}{2}}$

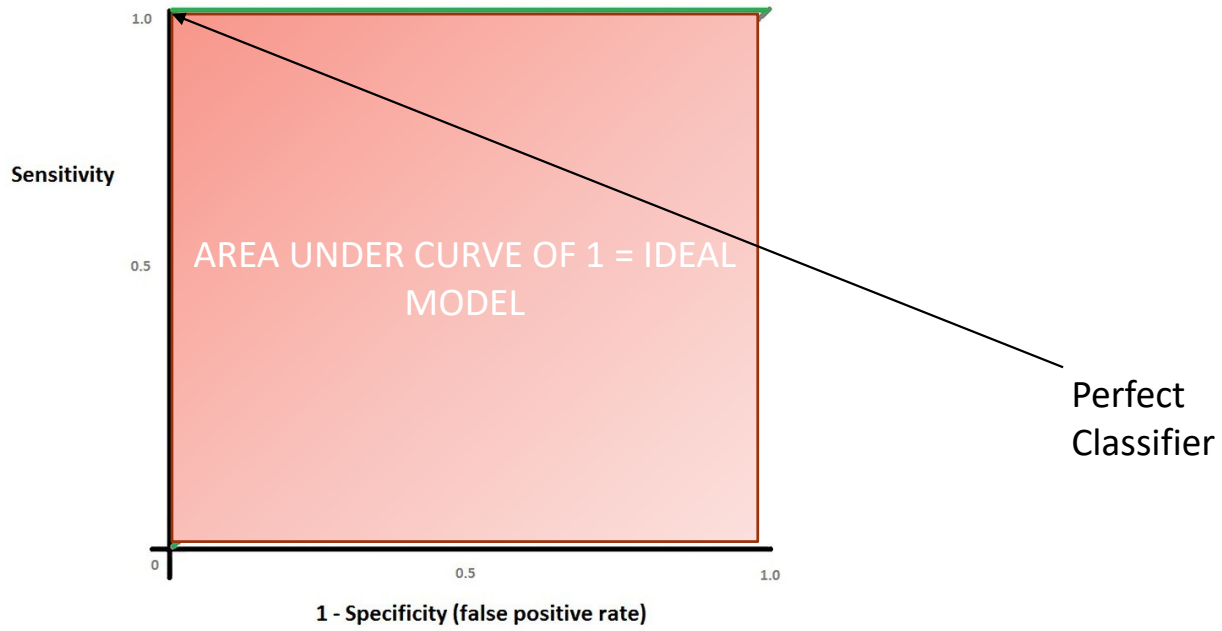
(harmonic mean of Precision and Recall)



https://www.dcode.fr/tools/precision-recall/images/precision-recall.png

Area Under Curve: Classification Can be Compared Across Models

Area Under the Curve (AUC): The closer you are to top left corner (to 1), the better the model is:



What if classes are unbalanced?

I.e., what if the total number of actual positives and actual negatives in the dataset is not the same, and say you have a dataset with 90% positives, 10% negatives?

Then the measure to use is the *Matthews Correlation Coefficient,* which if close to 0 shows the classifier is no better than random:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}$$

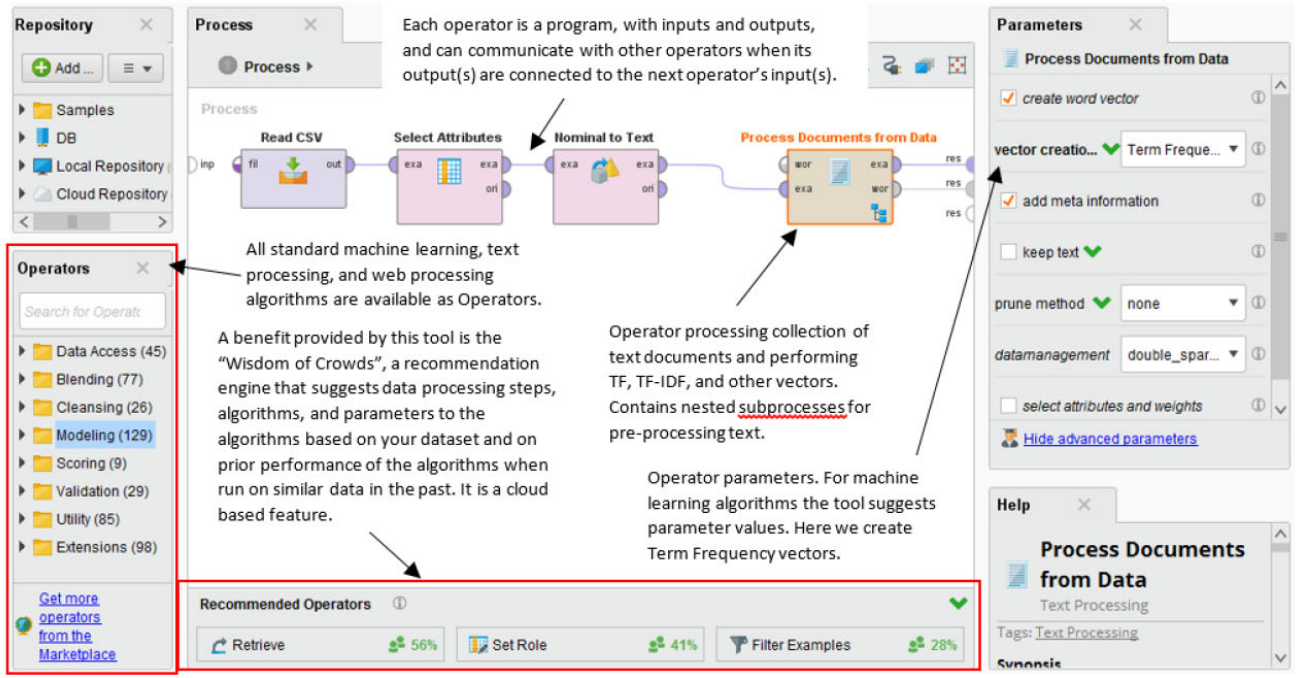
$$MCC \in [-1, 1]$$

An MCC of 1 indicates perfect prediction, -1 total disagreement between predicted and actual, and 0 random classifier.

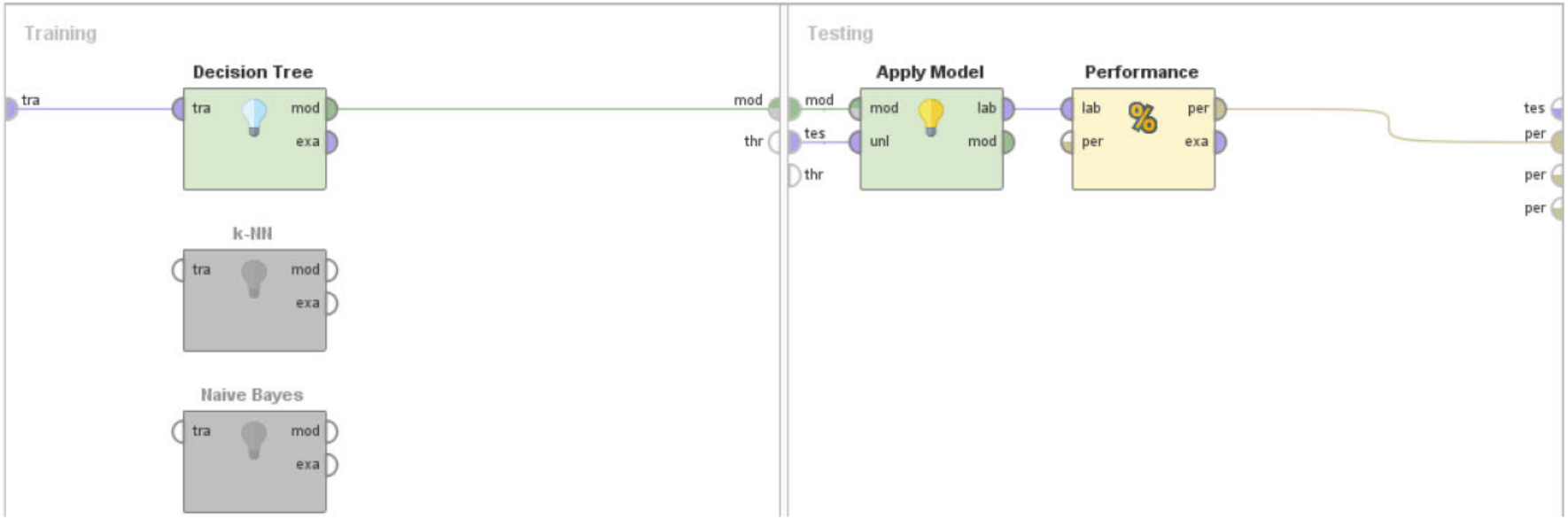
Want to learn more (easy reads?)

- Check out appendix of my work on fairness in ML – "Appendix: Fairness and Bias Considerations for Specific ML techniques" (pp. 62-73) "Exploring Fairness in Machine Learning for International Development", Awwad, Y., Fletcher, R., Frey, D., Najafian M., Teodorescu, M., MIT D-Lab CITE Reports Series, 2020 http://oastats.mit.edu/handle/1721.1/126854
- Check out the working paper "Machine Learning Methods for Strategy Research", Teodorescu M., HBS Working Paper Series 18-011, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3012524 pp 27 – on for an overview of trees, classification, and applications in RapidMiner, a drag and drop toolkit that makes "programming" super easy (and some examples of text analytics)

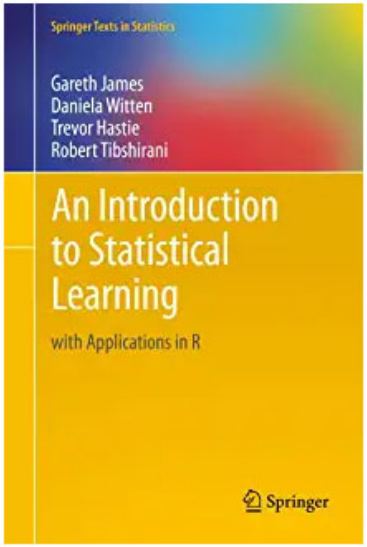
RapidMiner (free for academics)



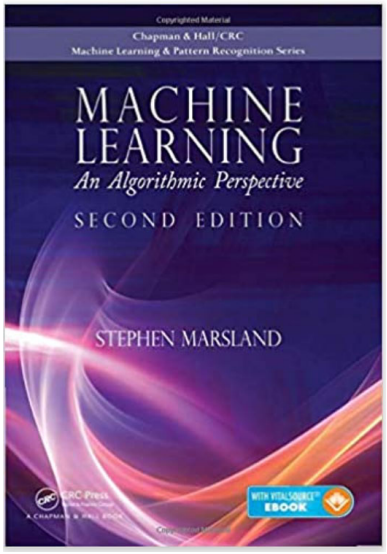
RapidMiner Cross-Validation



Good Books



Applied: Intro to Statistical Learning with Applications in R



Theory: Marsland's Machine Learning An Algorithmic Perspective

To Practice: look for an example backed by a paper

Audit Data Data Set

Download: Data Folder, Data Set Description

Abstract: Exhaustive one year non-confidential data in the year 2015 to 2016 of firms is collected from the Auditor Office of India to build a predictor for classifying suspicious firms.

Data Set Characteristics:	Multivariate	**Number of Instances:**	777	**Area:**	N/A
Attribute Characteristics:	Real	**Number of Attributes:**	18	**Date Donated**	2018-07-14
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	7055

Source:

Nishtha Hooda, CSED, TIET, Patiala

Data Set Information:

The goal of the research is to help the auditors by building a classification model that can predict the fraudulent firm on the basis the present and historical risk factors. The information about the sectors and the counts of firms are listed respectively as Irrigation (114), Public Health (77), Buildings and Roads (82), Forest (70), Corporate (47), Animal Husbandry (95), Communication (1), Electrical (4), Land (5), Science and Technology (3), Tourism (1), Fisheries (41), Industries (37), Agriculture (200).

Attribute Information:

Many risk factors are examined from various areas like past records of audit office, audit-paras, environmental conditions reports, firm reputation summary, on-going issues report, profit-value records, loss-value records, follow-up reports etc. After in-depth interview with the auditors, important risk factors are evaluated and their probability of existence is calculated from the present and past records.

Relevant Papers:

Hooda, Nishtha, Seema Bawa, and Prashant Singh Rana. 'Fraudulent Firm Classification: A Case Study of an External Audit.' Applied Artificial Intelligence 32.1 (2018): 48-64.

Some Basic Code Examples (R Studio)