Urban Informatics

Fall 2018

dr. federica bianco fbianco@nyu.edu

@fedhere

CUSP

XII: decision trees

Recap:

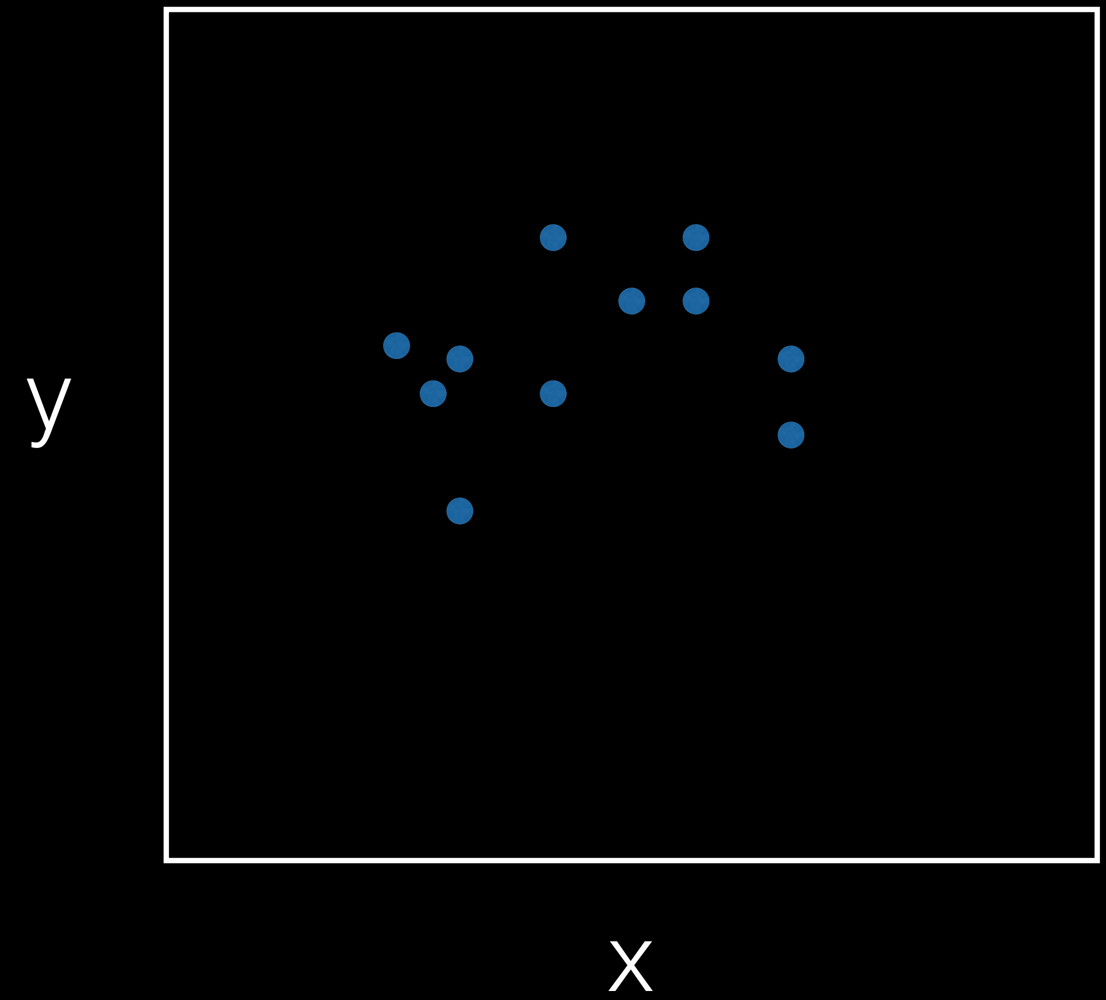
- Good practices with data: falsifiability, reproducibility
- Basic data retrieving and munging: APIs, Data formats
- SQL
- Basic statistics: distributions and their moments
- Hypothesis testing: *p*-value, statistical significance
- Statistical and Systematic errors
- Goodness of fit tests
- Likelihood
- OLS
- Topics in (time) series analysis
- Visualizations
- Geospatial analysis
- Clusters

Today:

- categorical and mixed clustering
- decision and regression trees (CART)
- tips on efficient coding

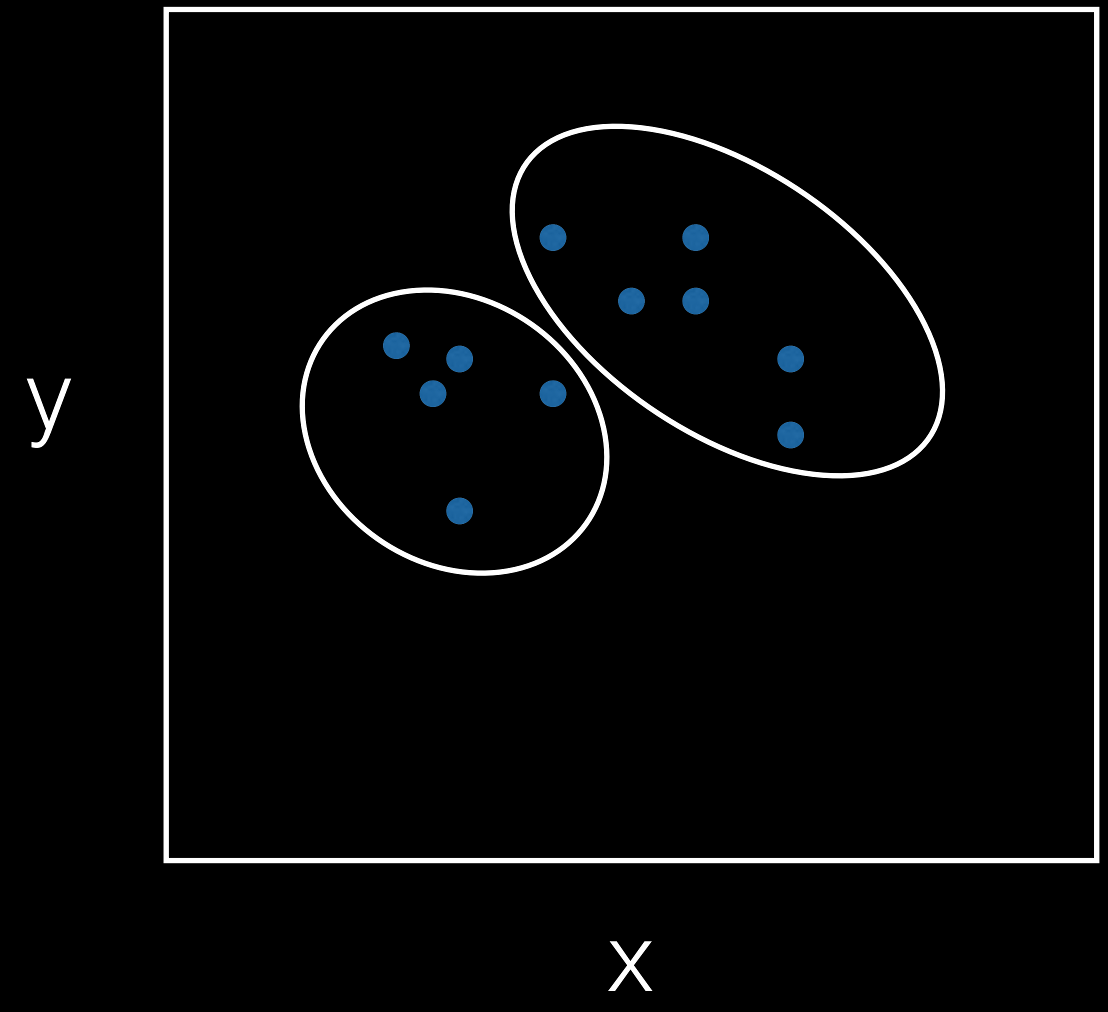
Partitioning methods: clustering

goal is to partition the space so that the observerd variables are separate in maximally homogeneous groups



observed:
(x, y)

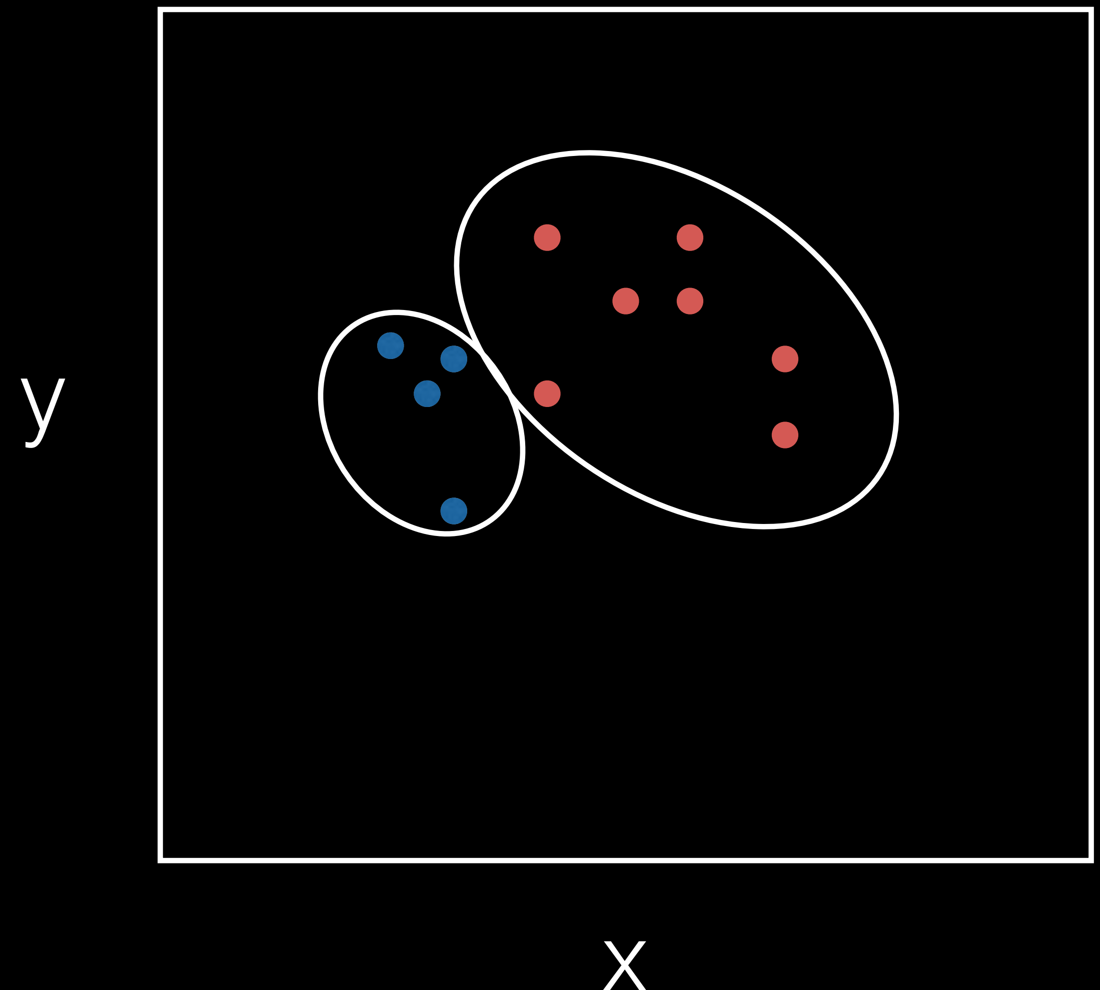
Partitioning methods: clustering



observed:
(x, y)

Partitioning methods: clustering

observed:
(x, y, color)



Summary and Key concepts

clustering is easy, but interpreting results is tricky

Distance metrics:

Eucledian and other Minchowski metrics

geospacial distances

metrics for non continuous data

Partitioning methods: inexpensive, typically non deterministic

Hard methods: *K-means, K-medoids*

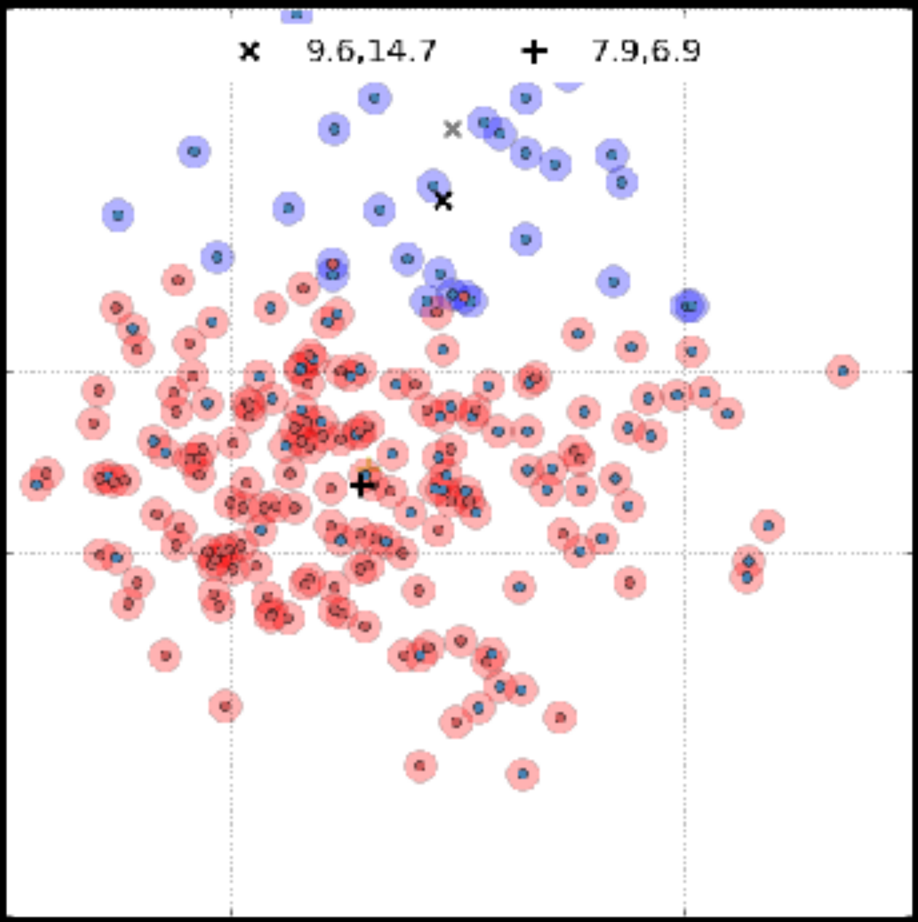
Soft (or fuzzy) methods: (i.e. probabilistic approach)

Expectation Maximization Mixture models

Hierarchical methods:

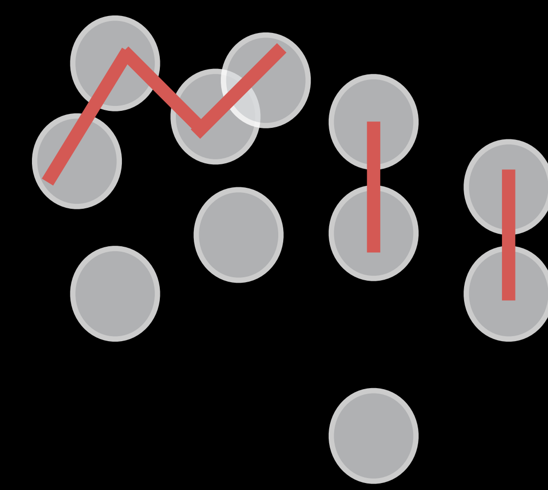
divisive vs agglomerative, dendrograms

Crisp (or hard) clustering - K-means

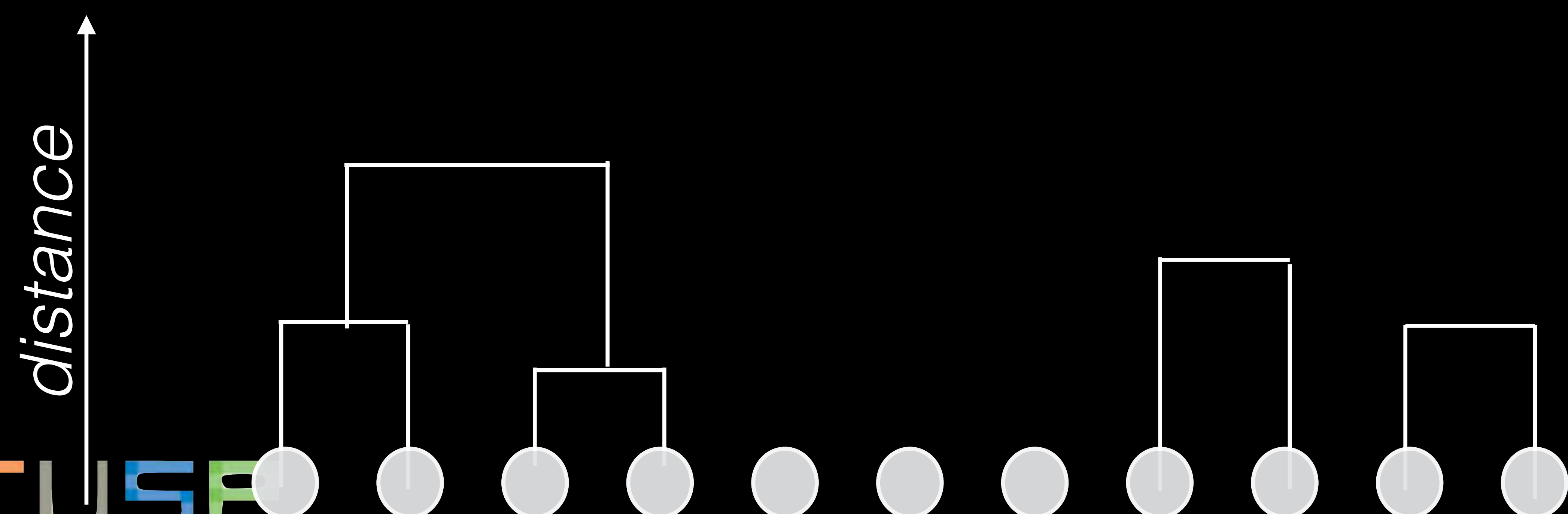


You guess the centers and assign points to clusters based on a predefined distance metric

hierarchical clustering

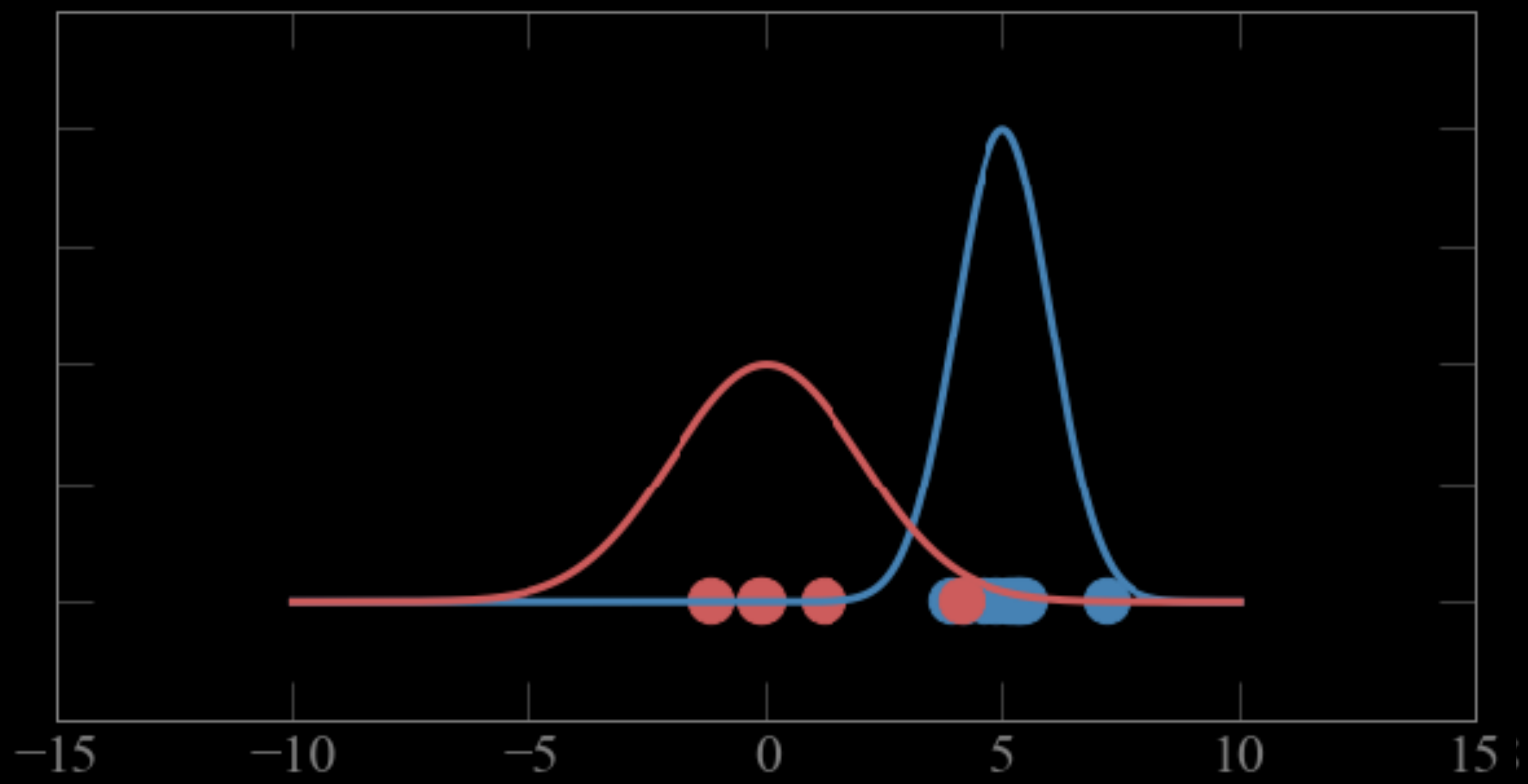


agglomerative
bottom-up



Fuzzy (or soft) clustering - Mixture models

A probabilistic way to do clustering



You adjust the parameters (μ,σ) of the gaussians iteratively based on the probability of the data coming from that gaussian

Hard Clustering:

each object in the sample belongs to only 1 cluster

Soft Clustering:

to each object in the sample we assign a degree of belief that it belongs to a cluster

Distance Metrics   Continuous variables

Minkowski family of distances

$$D(i,j) = \sqrt[p]{\sum_{k=1}^{N} |x_{ik} - x_{jk}|^p}$$

N features (dimensions)

Great Circle distances: $\phi_i, \lambda_i, \phi_j, \lambda_j$

geographical latitude and longitude

$$D(i,j) = R \arccos(\sin\phi_i \cdot \sin\phi_j + \cos\phi_i \cdot \cos\phi_j \cdot \cos(\Delta\lambda))$$

Distance Metrics **Binary variables**

contingency table

	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

CUSP

Distance Metrics

Binary variables

contingency table

	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

Distance Metrics

Binary variables

contingency table

	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

ELEVATOR

ESCALATOR

	1	0	
1	7	3	
0	106	353	

Distance Metrics

Binary variables

contingency table

	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

ELEVATOR

ESCALATOR

	1	0	sum
1	7	3	10
0	106	353	459
sum	113	356	469

Distance Metrics

Binary variables

contingency table

	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

ELEVATOR

ESCALATOR	1	0	sum
1	7	3	10
0	106	353	459
sum	113	356	469

IF SYMMETRIC
(same chance to appear
i.e. roughly same total
Y and N)

$$D_{ij} = \frac{b+c}{a+b+c+d} = \frac{109}{469} = 0.23$$

Distance Metrics

Binary variables

contingency table

	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

ESCALATOR (rows) / ELEVATOR (columns)

	1	0	sum
1	7	3	10
0	106	353	459
sum	113	356	469

IF SYMMETRIC
(same chance to appear
i.e. roughly same total
Y and N)

$$D_{ij} = \frac{M_{i=0j=0}+M_{i=1j=1}}{M_{00}+M_{01}+M_{10}+M_{11}} = \frac{109}{469} = 0.23$$

CUSP

Distance Metrics

Binary variables

contingency table

	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

ELEVATOR

ESCALATOR	1	0	sum
1	7	3	10
0	106	353	459
sum	113	356	469

IF ASYMMETRIC
(not same chance)

$$D_{ij} = \frac{b+c}{a+b+c} = \frac{109}{116} = 0.94$$

Distance Metrics

Binary variables

contingency table

	1	0	sum
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

e.g.: subway station
w ESCALATOR Y/N
w ELEVATOR Y/N

ELEVATOR

ESCALATOR	1	0	sum
1	7	3	10
0	106	353	459
sum	113	356	469

IF ASYMMETRIC
(not same chance)

Jaccard similarity

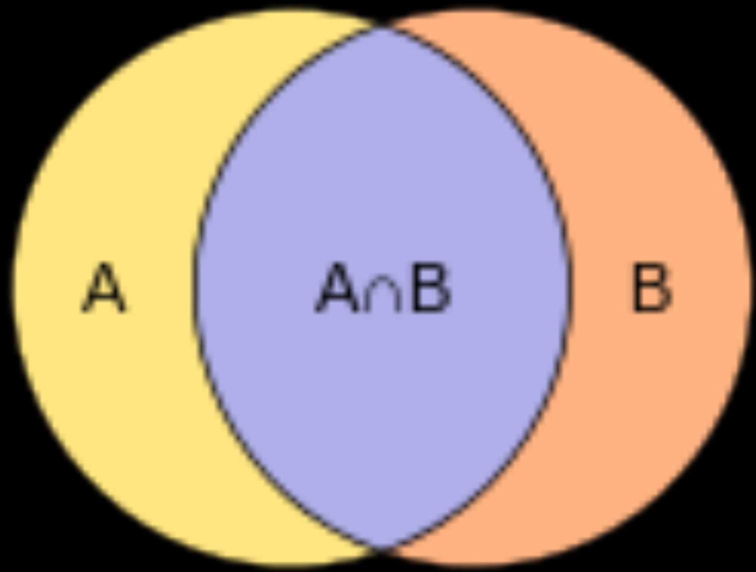
$$J_{ij} = \frac{a}{a+b+c} = \frac{7}{116} = 0.06$$

Distance Metrics Binary variables

Uses presence/absence data

Jaccard similarity coefficient S_j

$$S_j = \frac{a}{a+b+c}$$



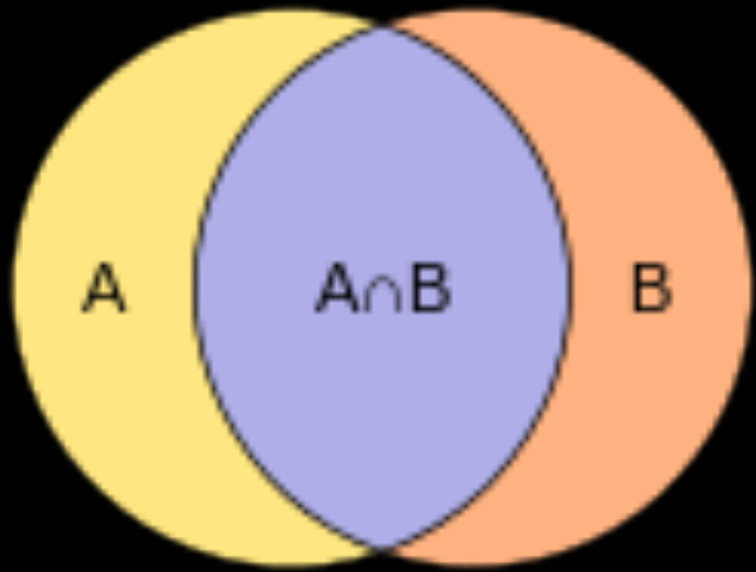
a = number of items in common,
b = number of items unique to the first set
c = number of items unique to the second set

Distance Metrics   Binary variables

Uses presence/absence data

Jaccard similarity coefficient S_j

$$S_j = \frac{A \bigcap B}{A \bigcup B}$$



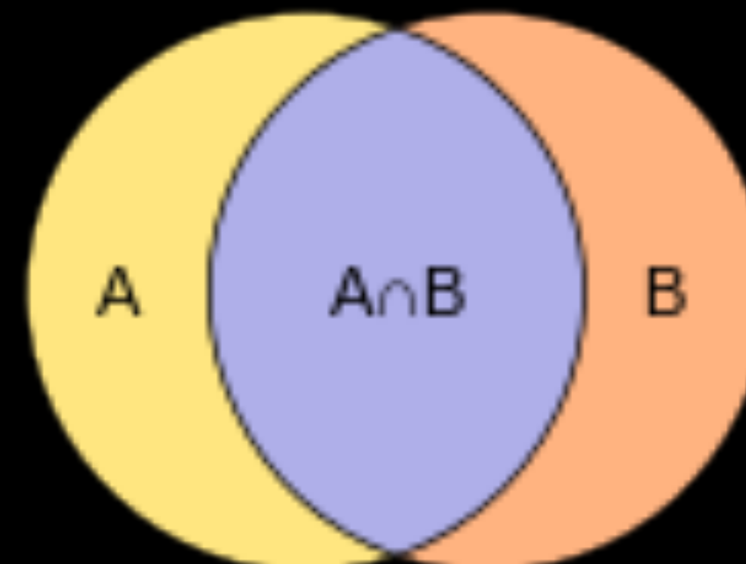
a = number of items in common,
b = number of items unique to the first set
c = number of items unique to the second set

Distance Metrics   Binary variables

Uses presence/absence data

Jaccard distance
$D_j = 1 - S_j$

$$S_j = \frac{A \bigcap B}{A \bigcup B}$$



a = number of items in common,
b = number of items unique to the first set
c = number of items unique to the second set

Distance Metrics Categorical Variables

Uses presence/absence data in two samples (non exclusive)

Simple similarity coefficient

Simple Matching Method

SMC

$$S_{ij} = \frac{p - m}{p}$$

p: number of variables

m: number of matches

jupyter

https://github.com/fedhere/UInotebooks/blob/master/cluster/categorical_clustering.ipynb

CUSP

Distance Metrics Ordinal variables

Uses ranks

map occurrences in a range 0-1

$r_{ij} = \{1 \ldots R_N\}$ -> $\mathbf{z_{ij} = \dfrac{r_{ij} - 1}{R_N - 1}}$

Distance Metrics vector Variables

Uses correlation coefficient!

A time series is a vector:
MTA rides/NYC establishments can be clustered w this distance
clustering time series + other features requires this

Pearson's correlation

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Cosine similarity

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

Distance Metrics MIXED variables

Hybrid dataset containing continuous, ordinal, categorical

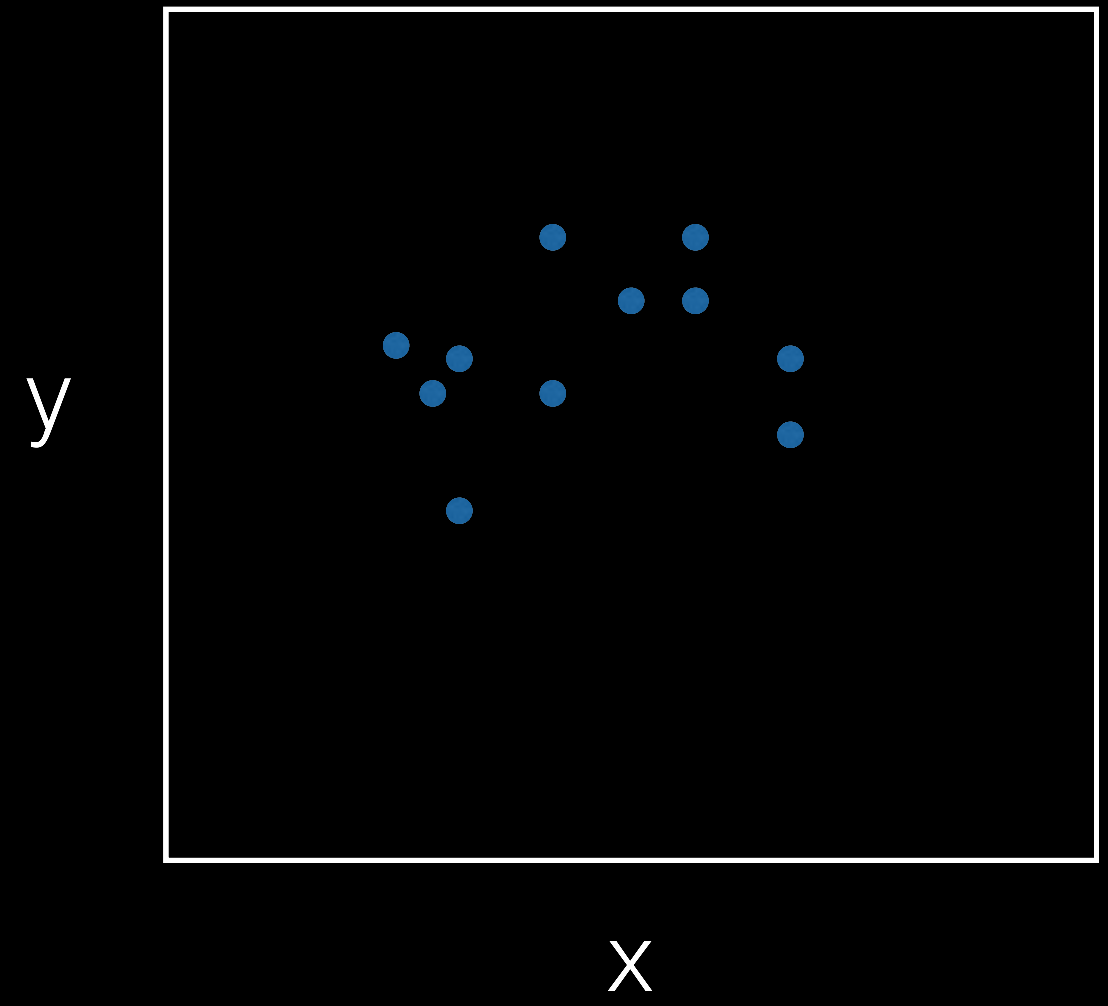
weighted distance

$$D_w = \frac{\sum_{p=1}^{p} w_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{p=1}^{p} w_{ij}^{(f)}}$$

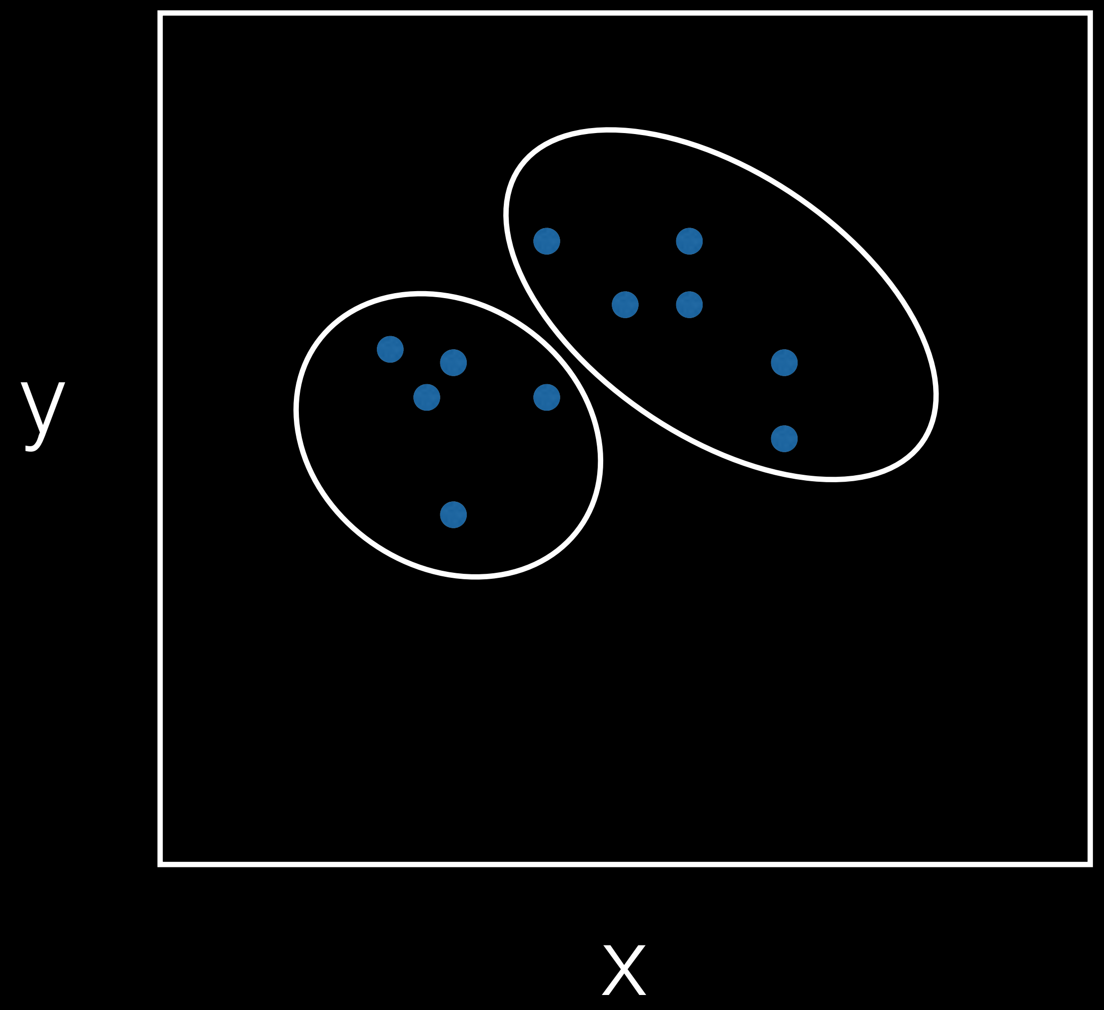
Partitioning methods: clustering

goal is to partition the space so that the observerd variables are separate in maximally homogeneous groups

observed:
(x, y)



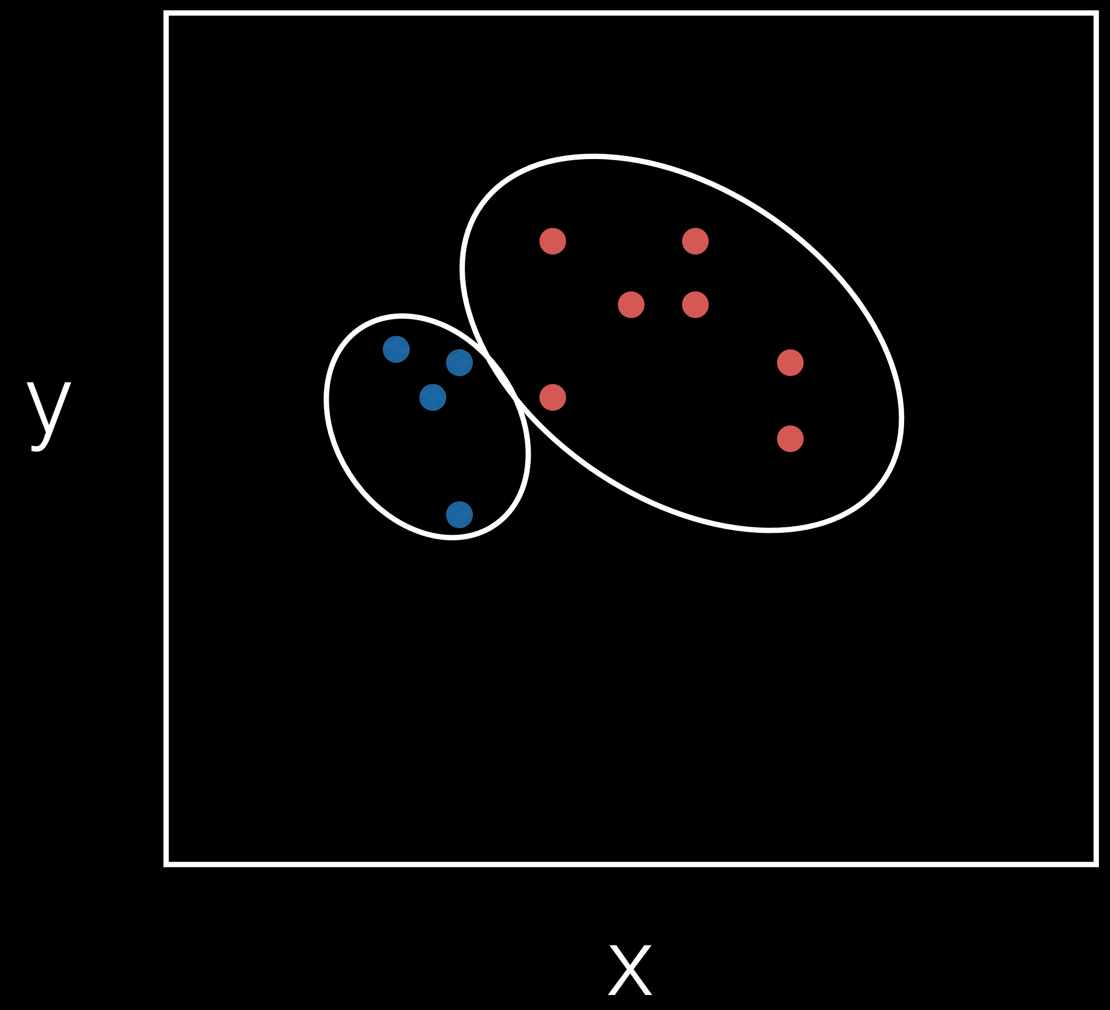
Partitioning methods: clustering



observed:
(x, y)

Partitioning methods: clustering

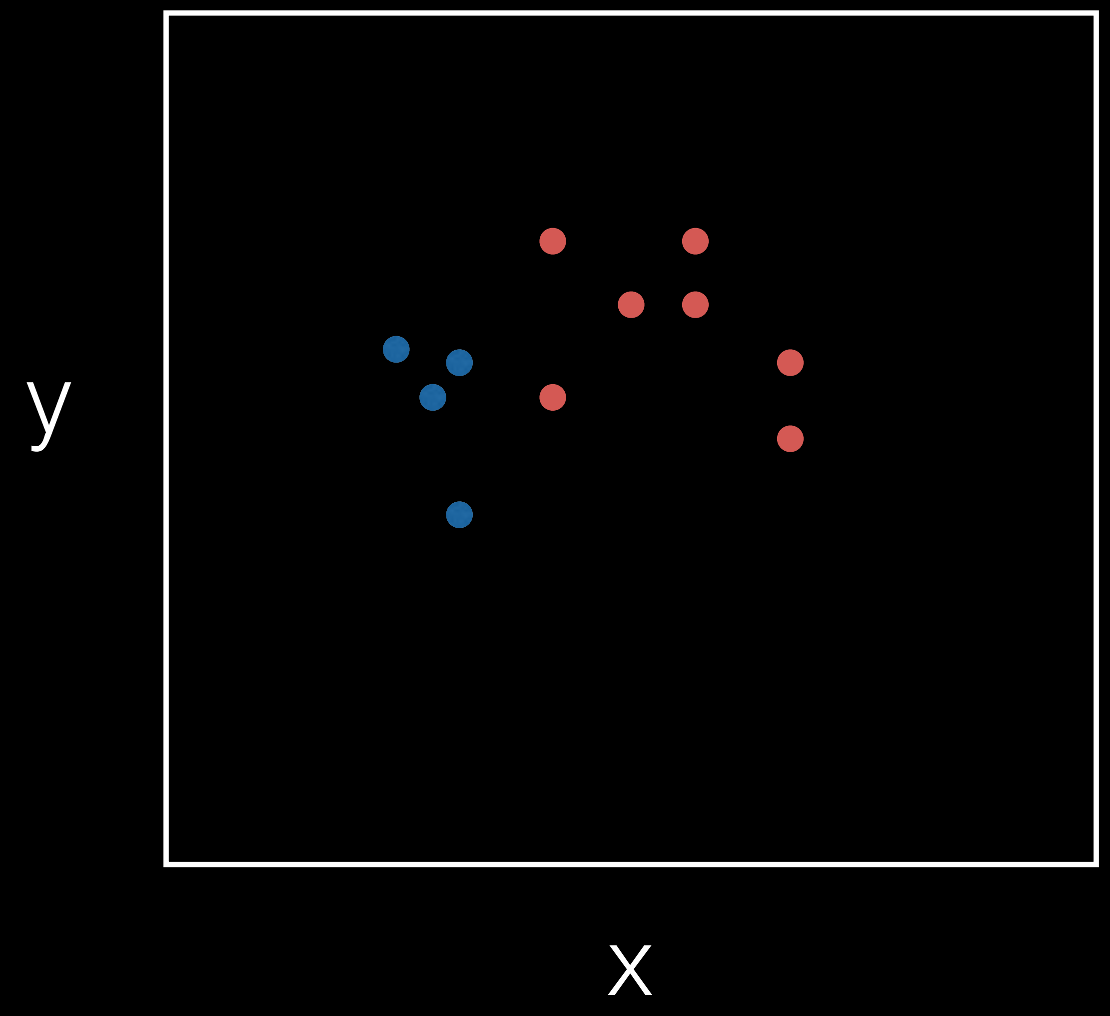
observed*:*
(x, y, color)



Partitioning methods: classifying (SVM, CART)

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)

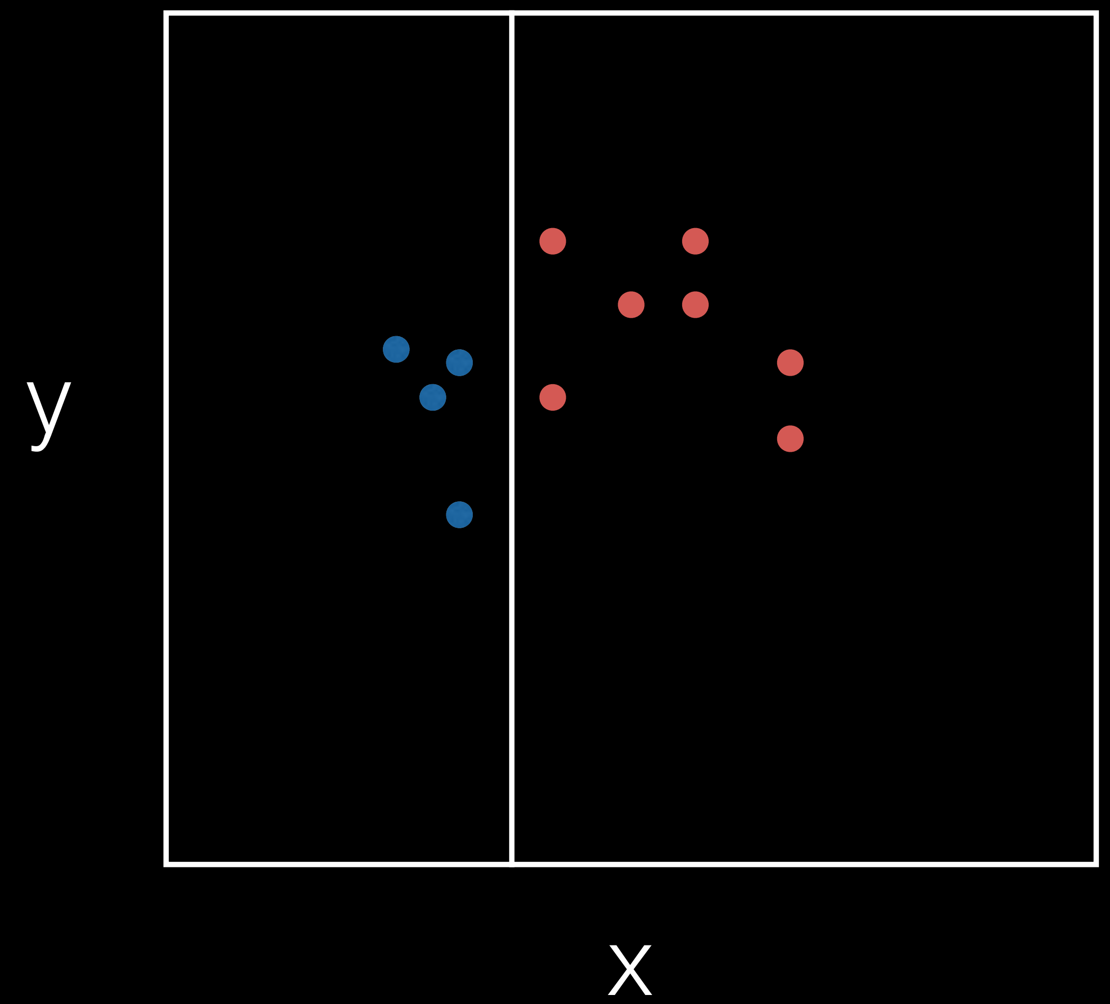


target:
(color)

Partitioning methods: classifying

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)

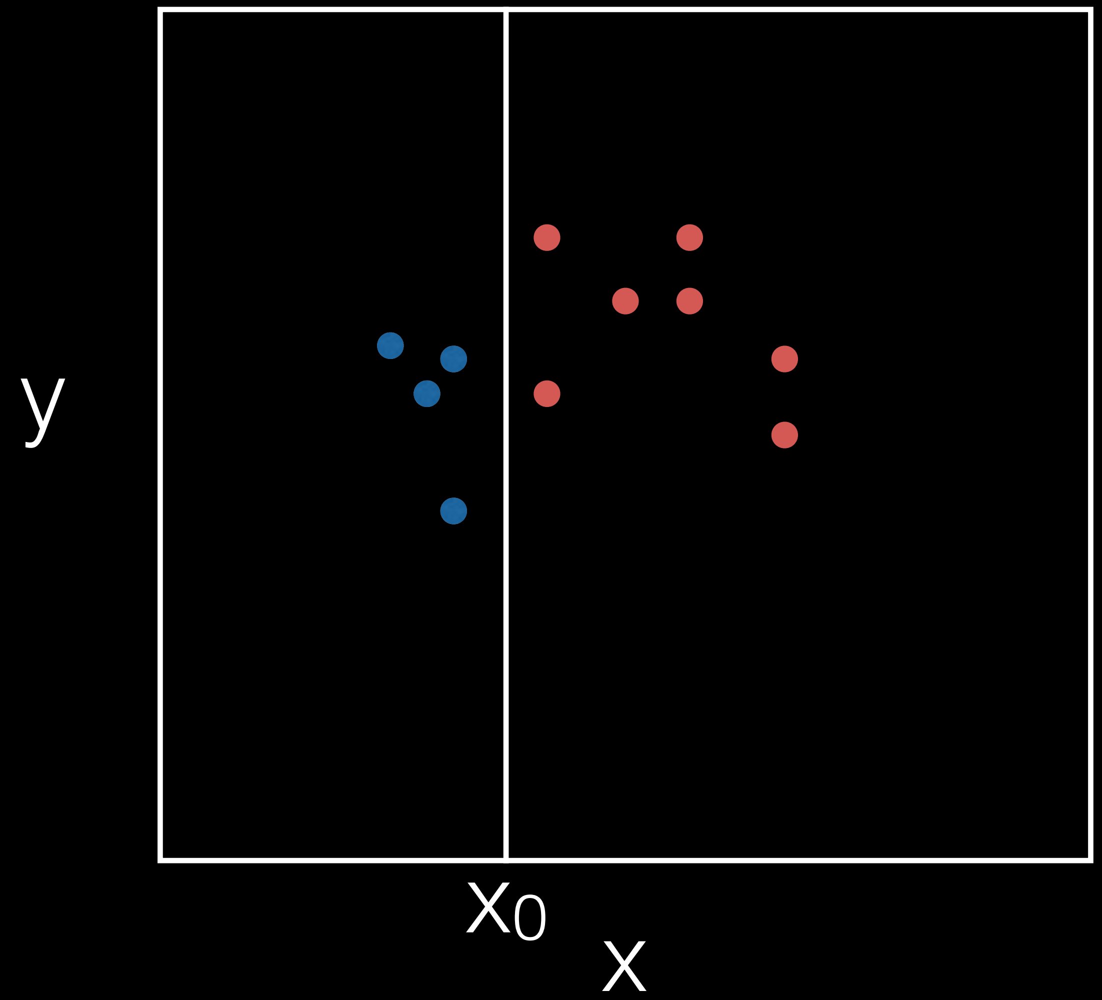


target:
(color)

Partitioning methods: classifying

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)



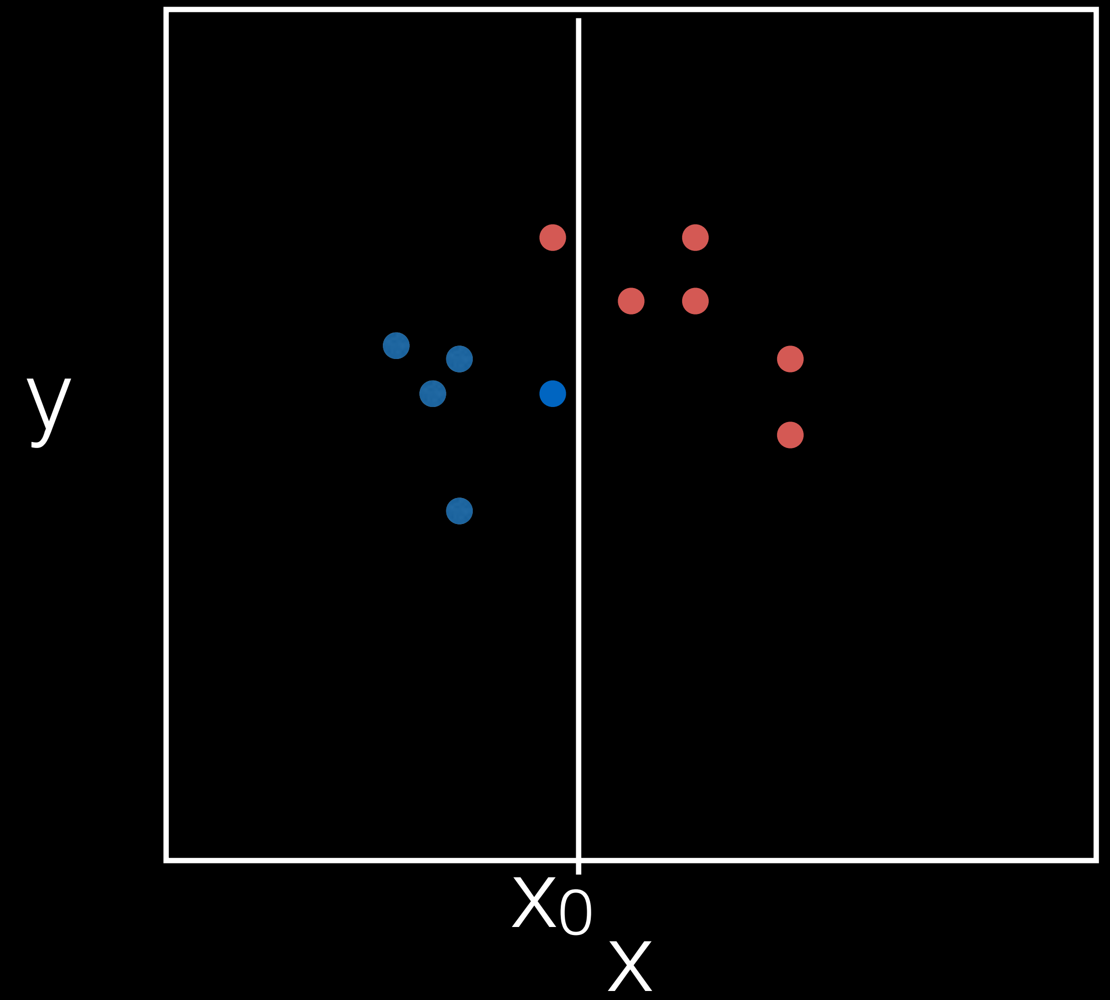
target:
(color)

if $x > x_0$ => ball is red

Partitioning methods: classifying

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)

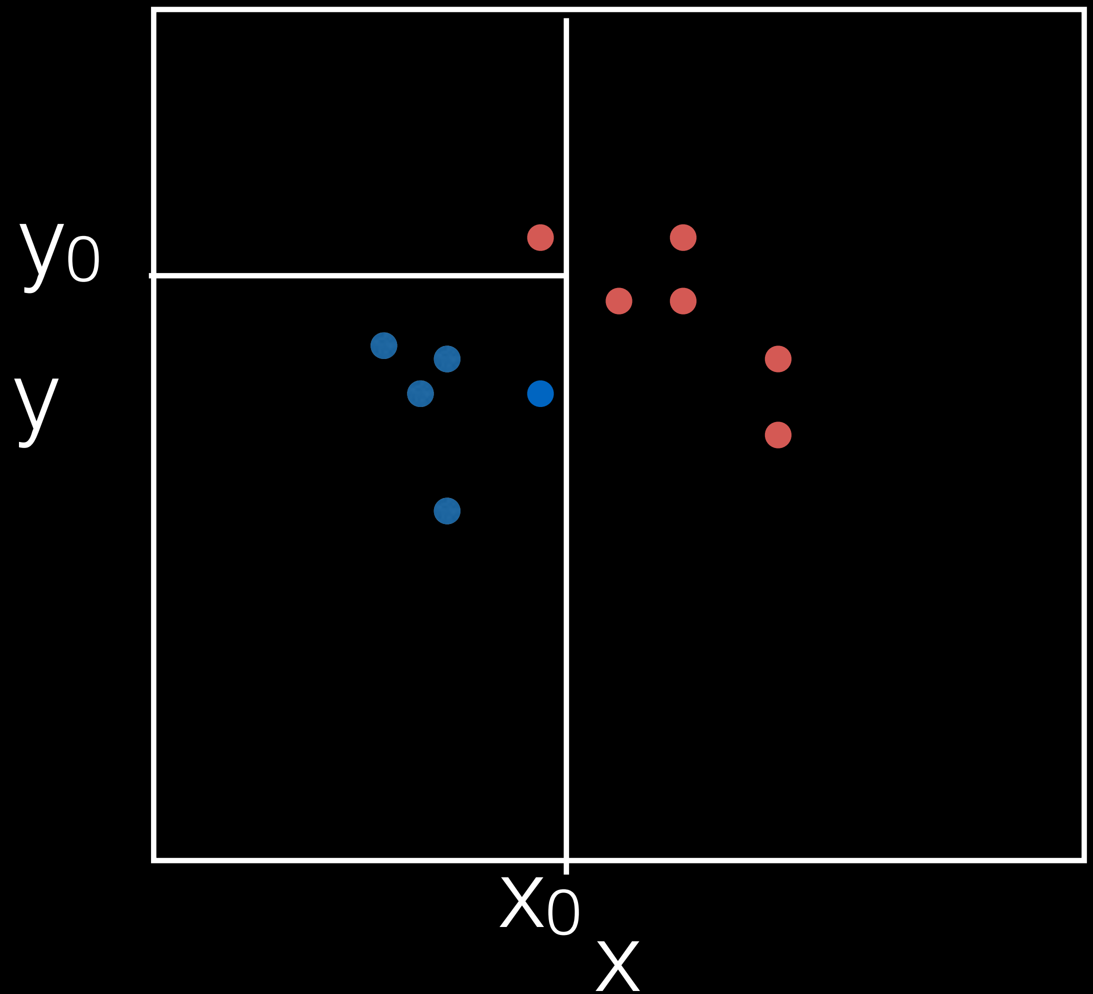


target:
(color)

Partitioning methods: classifying

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)



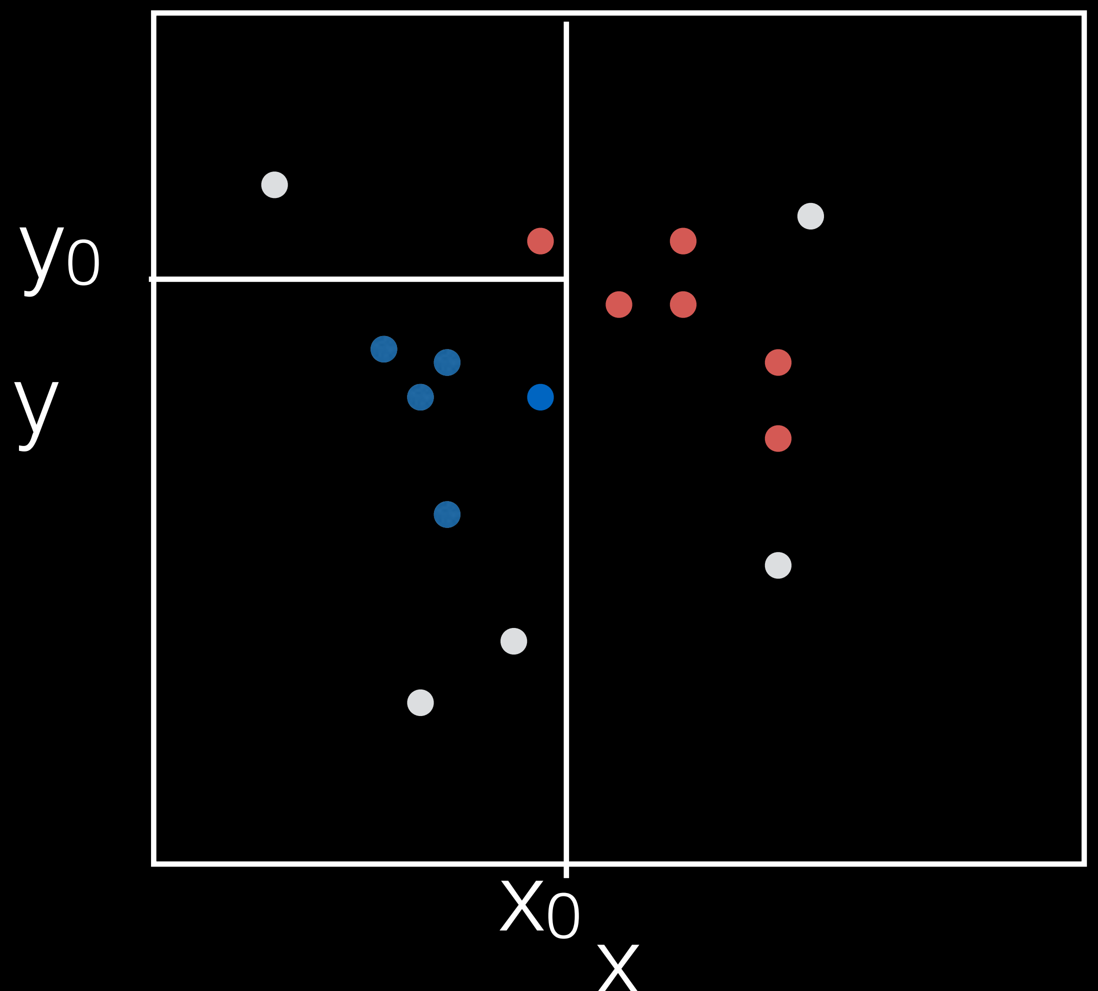
target:
(color)

if $x>x_0$ or $y>y_0$ => ball is red

Partitioning methods: classifying

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)



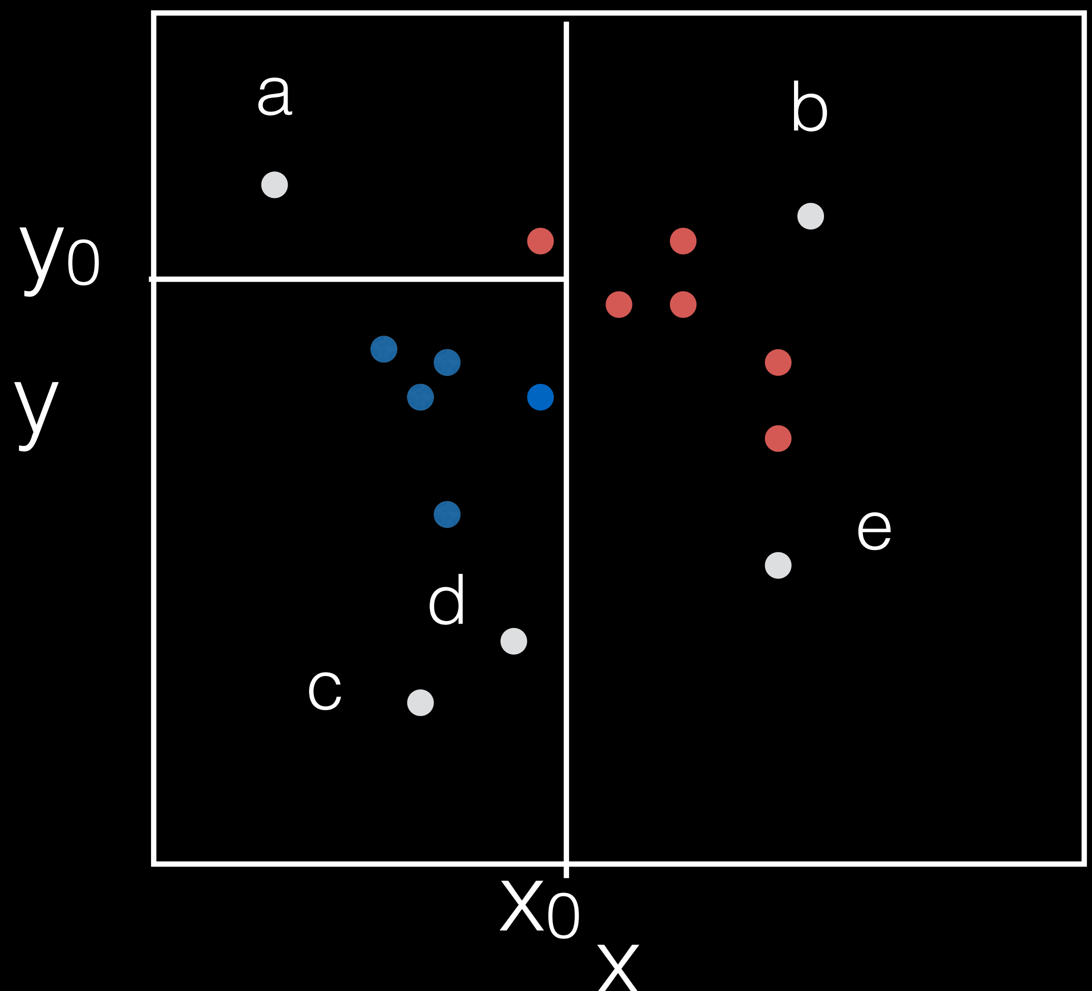
target:
(color)

if $x>x_0$ or $y>y_0$ => ball is red

Partitioning methods: classifying

goal is to partition the space of observed variables
to separate the space of unobserved (target variables)

observed:
(x, y)



a: red
b: red
c: blue
d: blue
e: red

target:
(color)

if $x>x_0$ or $y>y_0$ => ball is red

Decision Trees

The good

- Non-Parametric
- White-box: can be easily interpreted
- Works with any feature type and mixed feature types
- Works with missing data
- Robust to outliers

The bad

- High variability (-> use *ensamble* methods)
- Tendency to overfit
- (not as easily interpretable after all…)

a single tree

714 passengers
Ns=424 Nd=290

Application:
a robot to predict surviving the Titanic (Kaggle)

features:
gender
ticket class
age

target variable:
survival (y/n)

714 passengers
Ns=424 Nd=290

Application: a robot to predict surviving the Titanic (Kaggle)

gender (binary)

M
Ns=93 Nd=360
purity 79%

F
Ns=197 Nd=64
purity 75%

features:
gender 79/75%
ticket class
age

target variable:
survival (y/n)

714 passengers
Ns=424 Nd=290

Application:
a robot to predict surviving the Titanic (Kaggle)

class (categorical)

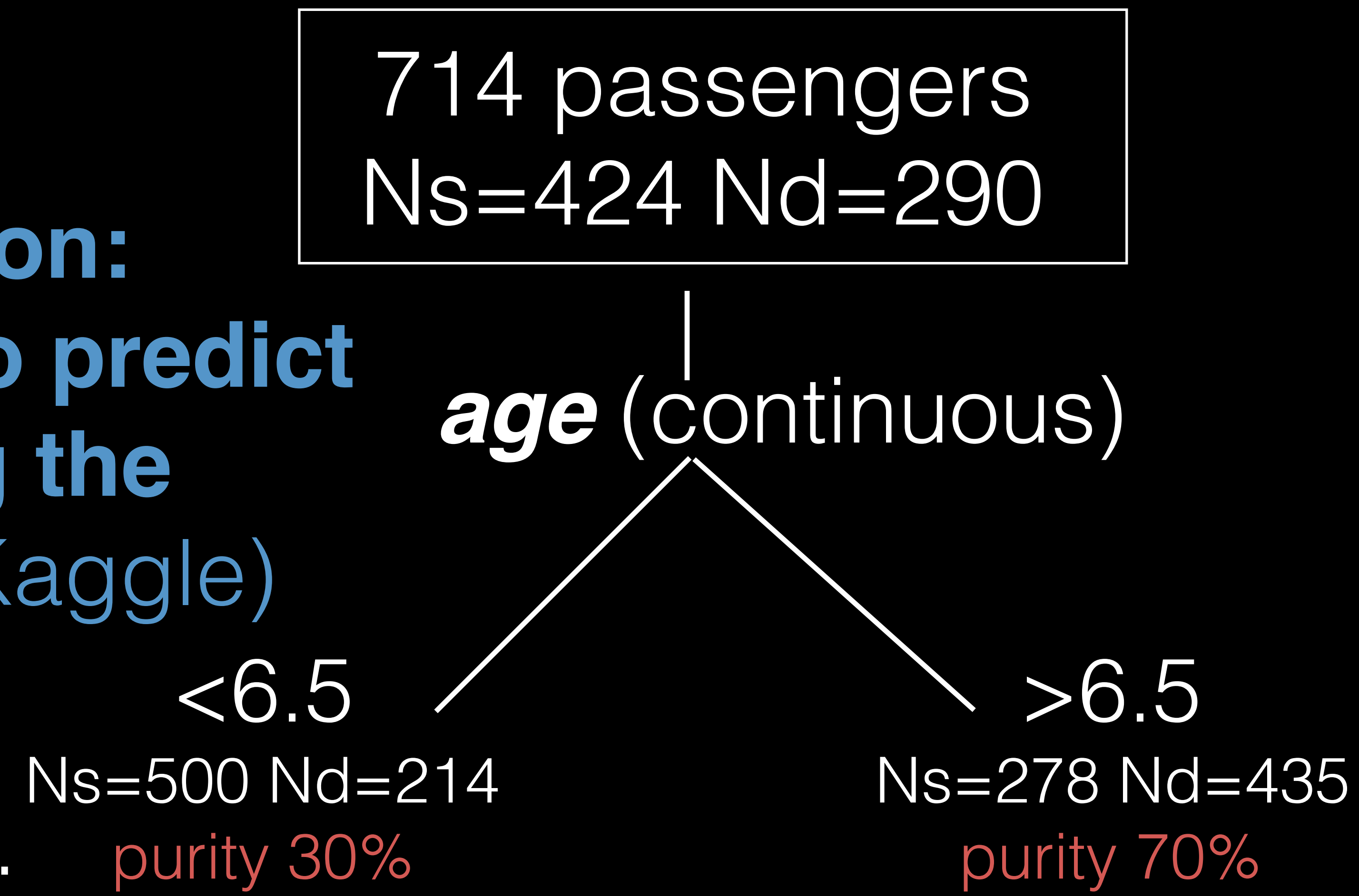
1st
Ns=471 Nd=242
purity 66%

2nd,3rd
Ns=335 Nd=378
purity 44%

features:
gender 79/75%
ticket class 66/44%
age

target variable:
survival (y/n)

Application: a robot to predict surviving the Titanic (Kaggle)



features:
gender 79/75%
ticket class 66/44%
age 30/70%

target variable:
survival (y/n)

Application: a robot to predict surviving the Titanic (Kaggle)

714 passengers
Ns=424 Nd=290

age (continuous)

<6.5
Ns=500 Nd=214
purity 30%

>6.5
Ns=278 Nd=435
purity 70%

features:
gender 79/75%
age 66/44%
ticket class 30/70%

target variable:
survival (y/n)

Application:
a robot to predict surviving the Titanic (Kaggle)

714 passengers
Ns=424 Nd=290

gender (binary)

M
Ns=93 Nd=360
purity 79%

F
Ns=197 Nd=64
purity 75%

Application: a robot to predict surviving the Titanic (Kaggle)

714 passengers
Ns=424 Nd=290

gender (binary)

M
Ns=93 Nd=360
purity 79%

F
Ns=197 Nd=64
purity 75%

features:
gender 79/75%
age: M 74/67% F 96/40%
ticket class: M 40/15% F 96/65%

target variable:
survival (y/n)

CUSP

Application: a robot to predict surviving the Titanic (Kaggle)

714 passengers
Ns=424 Nd=290

gender (binary)

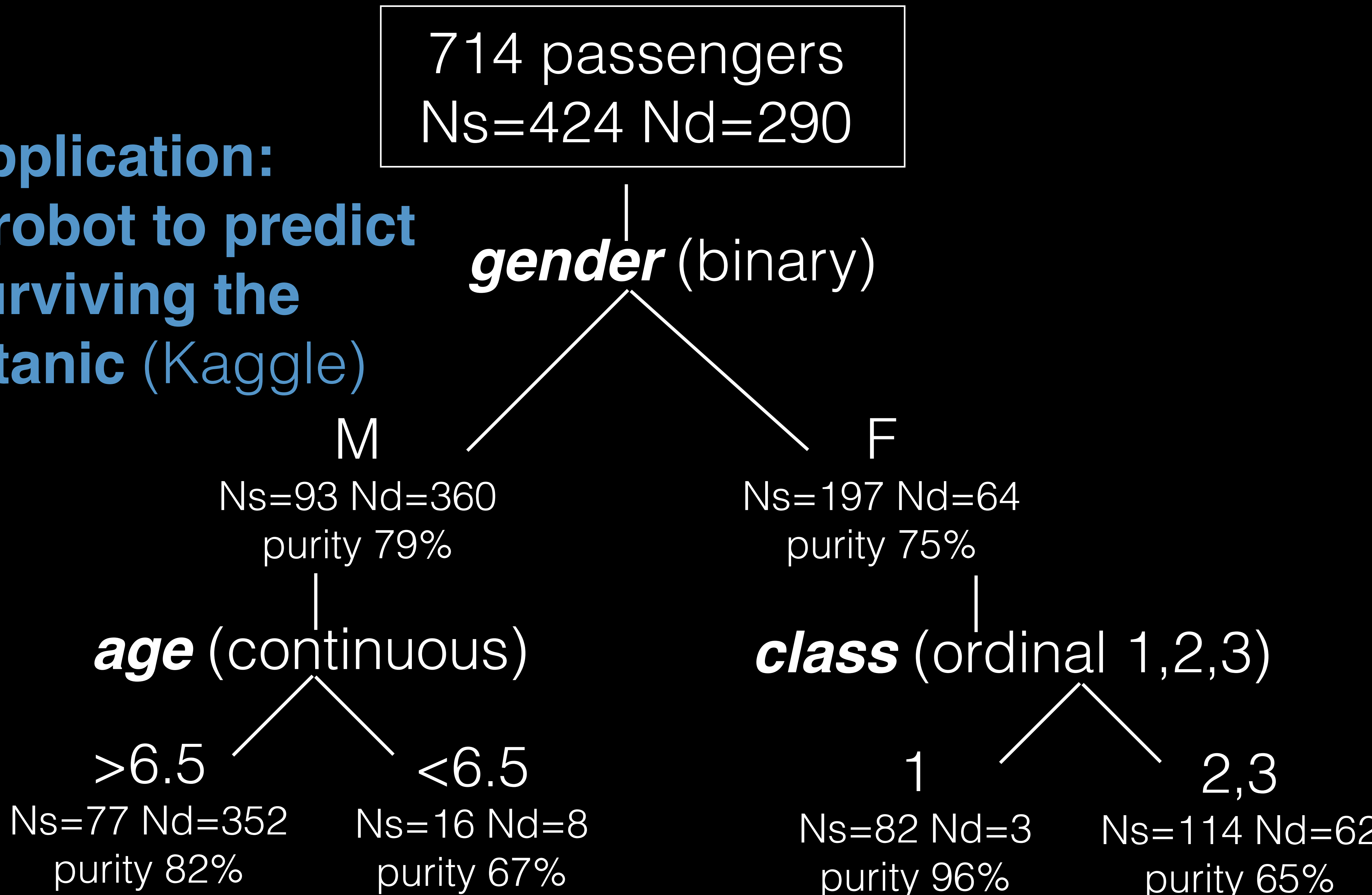
M
Ns=93 Nd=360
purity 79%

F
Ns=197 Nd=64
purity 75%

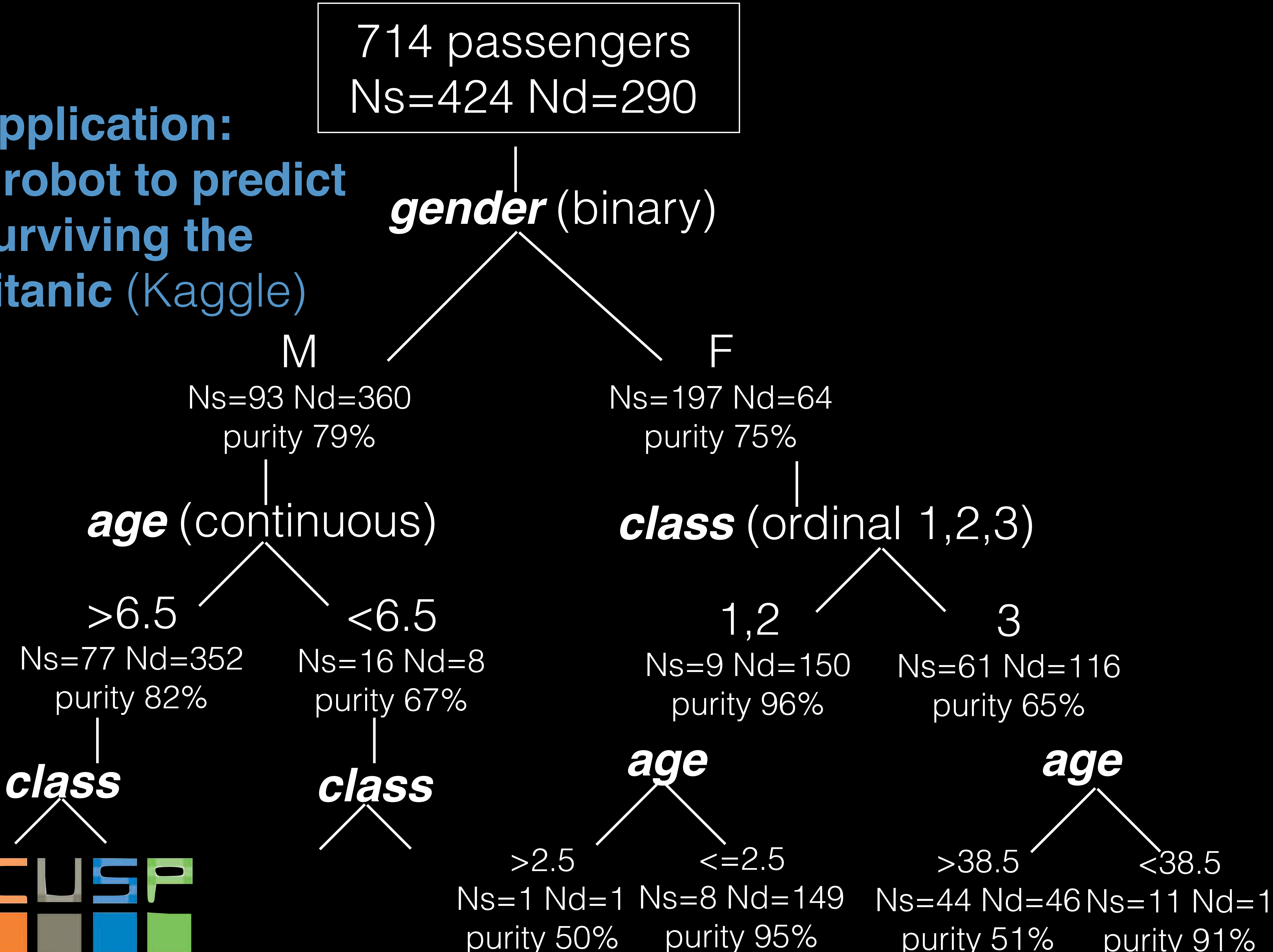
features:
gender 79/75%
age: M 67/82% F 74/76%
ticket class: M 40/15% F 96/65%

target variable:
survival (y/n)

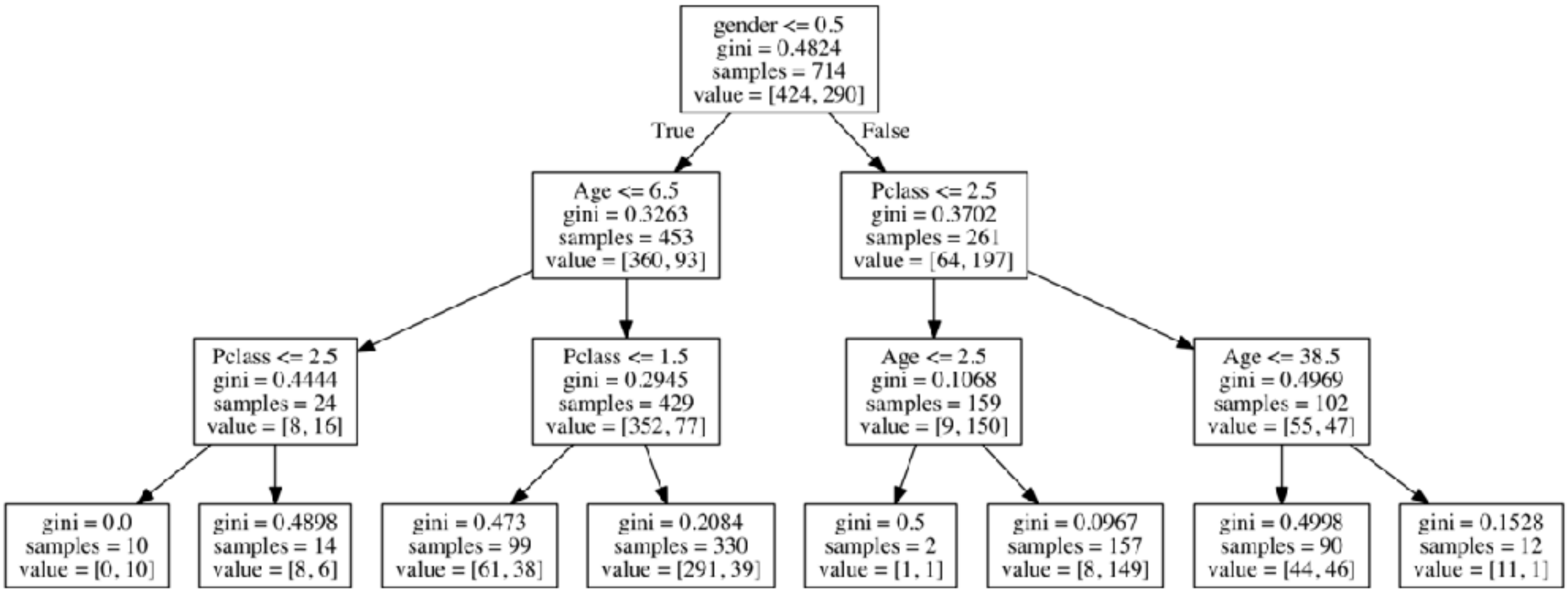
Application: a robot to predict surviving the Titanic (Kaggle)



Application: a robot to predict surviving the Titanic (Kaggle)



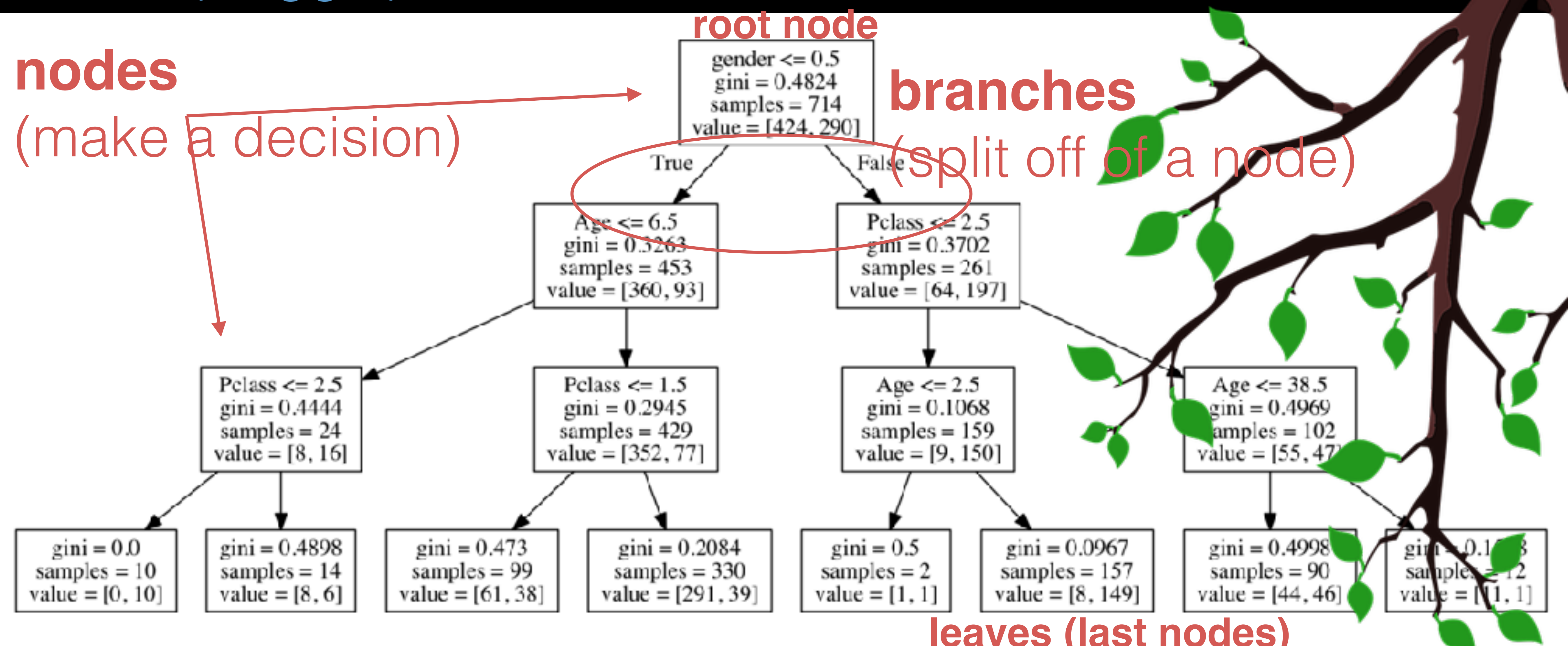
Application: a robot to predict surviving the Titanic (Kaggle)



CUSP

https://github.com/fedhere/PUI2017_fb55/blob/master/Lab12_fb55/TitanicByCART.ipynb

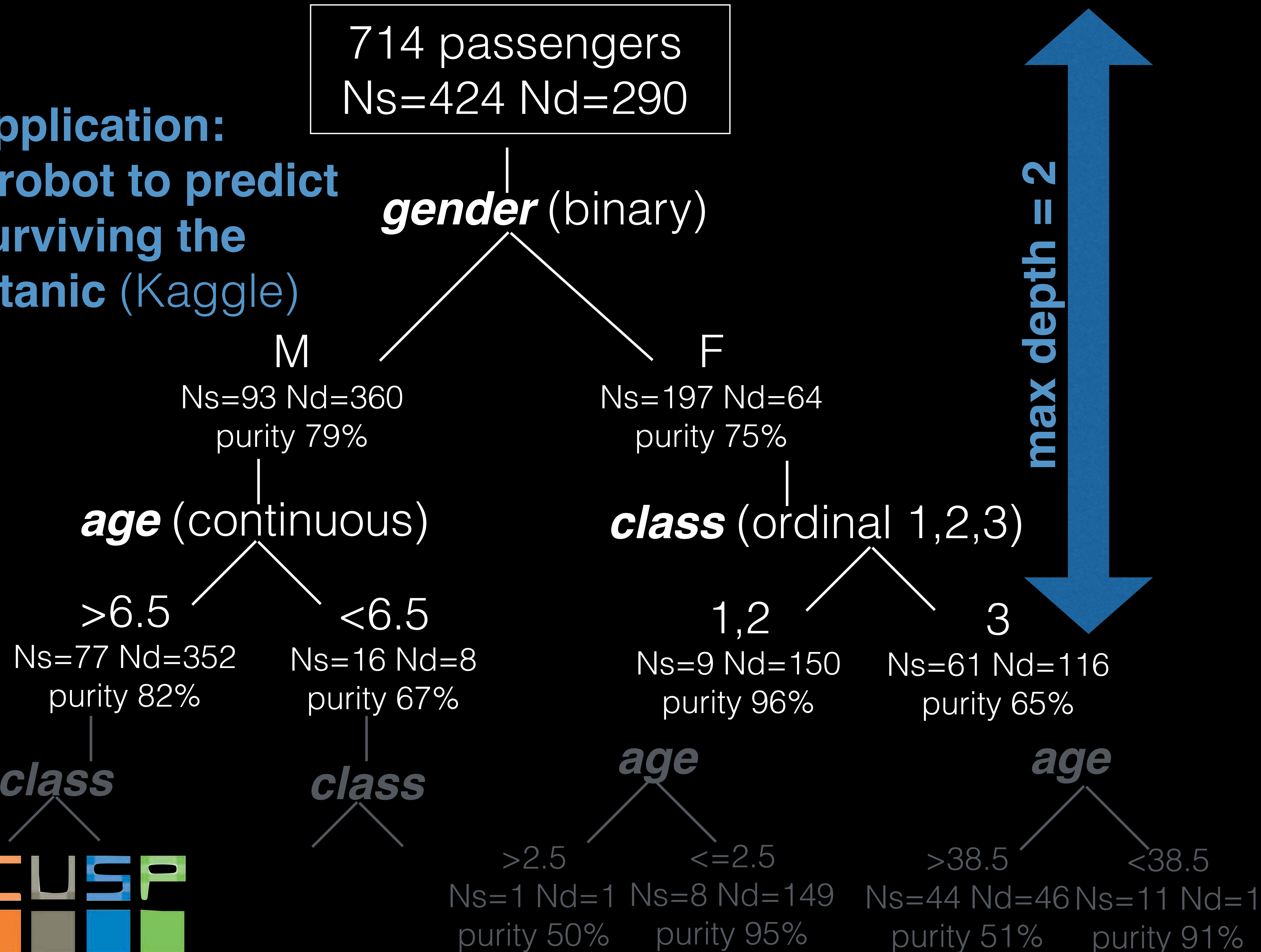
Application: a robot to predict surviving the Titanic (Kaggle)

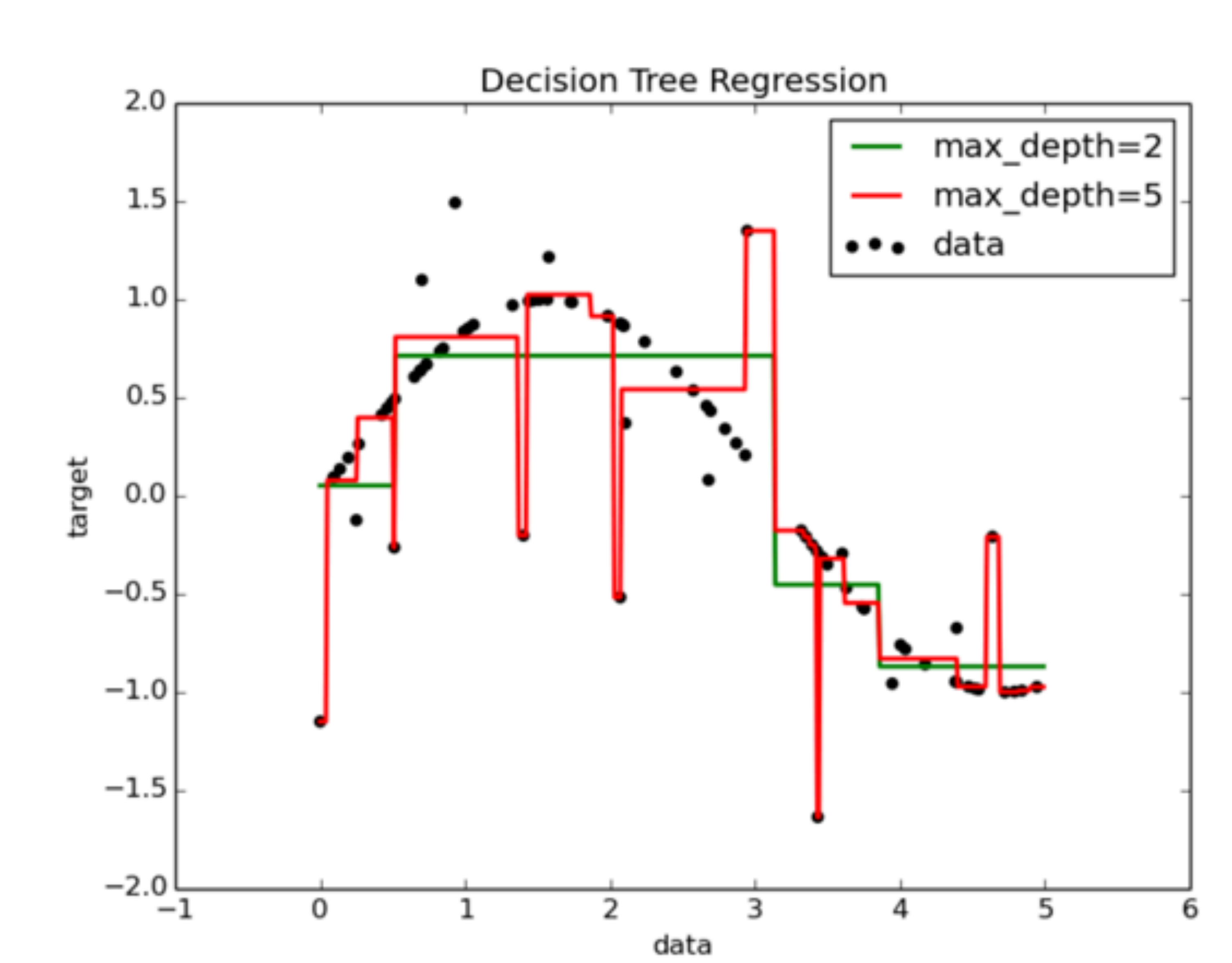


https://github.com/fedhere/PUI2017_fb55/blob/master/Lab12_fb55/TitanicByCART.ipynb a single tree

parameters:
maximum depth (controls overfitting)
maximization scheme

Application: a robot to predict surviving the Titanic (Kaggle)





http://scikit-learn.org/0.16/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

a single tree

parameters:
maximum depth (controls overfitting)
maximization scheme

gini, entropy (information content), variance…

a single tree

issues **:**

variance - different trees lead to different results

solution **: a forest**

a single tree

issues **:**

variance - different trees lead to different results

solution **: a forest**

- run many tree models,
- look at the ensemble result

Ensemble methods:

Random forest:

- trees run in parallel (independently of each other)
- each tree uses a random subset of observations/features (boostrap - bagging)
- class predicted by *majority vote:* what class do most trees think a point belong to?

Gradient boosted trees:

- trees run in series (one after the other)
- each tree uses different weights for the features learning the weighs from the previous tree
- the last tree has the prediction

Ensemble methods:

Random forest:

- trees run in parallel (independently of each other)
- each tree uses a random subset of observations/features (boostrap - bagging)
- class predicted by *majority vote:* what class do most trees think a point belong to?

Gradient boosted trees:

- trees run in series (one after the other)
- each tree uses different weights for the features learning the weighs from the previous tree
- the last tree has the prediction

More parameters:

- BOTH: depth, criterion, min sample to split, min sample in leaf
- RF: number of trees, number of features/tree
- GB: loss function, learning rate, number of boosts

super important missing topic:
pruning!
when is my tree overfitting?

dont just do linear regression!

http://scikit-learn.org/0.16/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

is your code optimized:

check CPU AND MEMORY usage

vectorize (slice and avoid for loops)

avoid storing information you do not need in memory

use local variables

remove all redundant calculations from inside loops

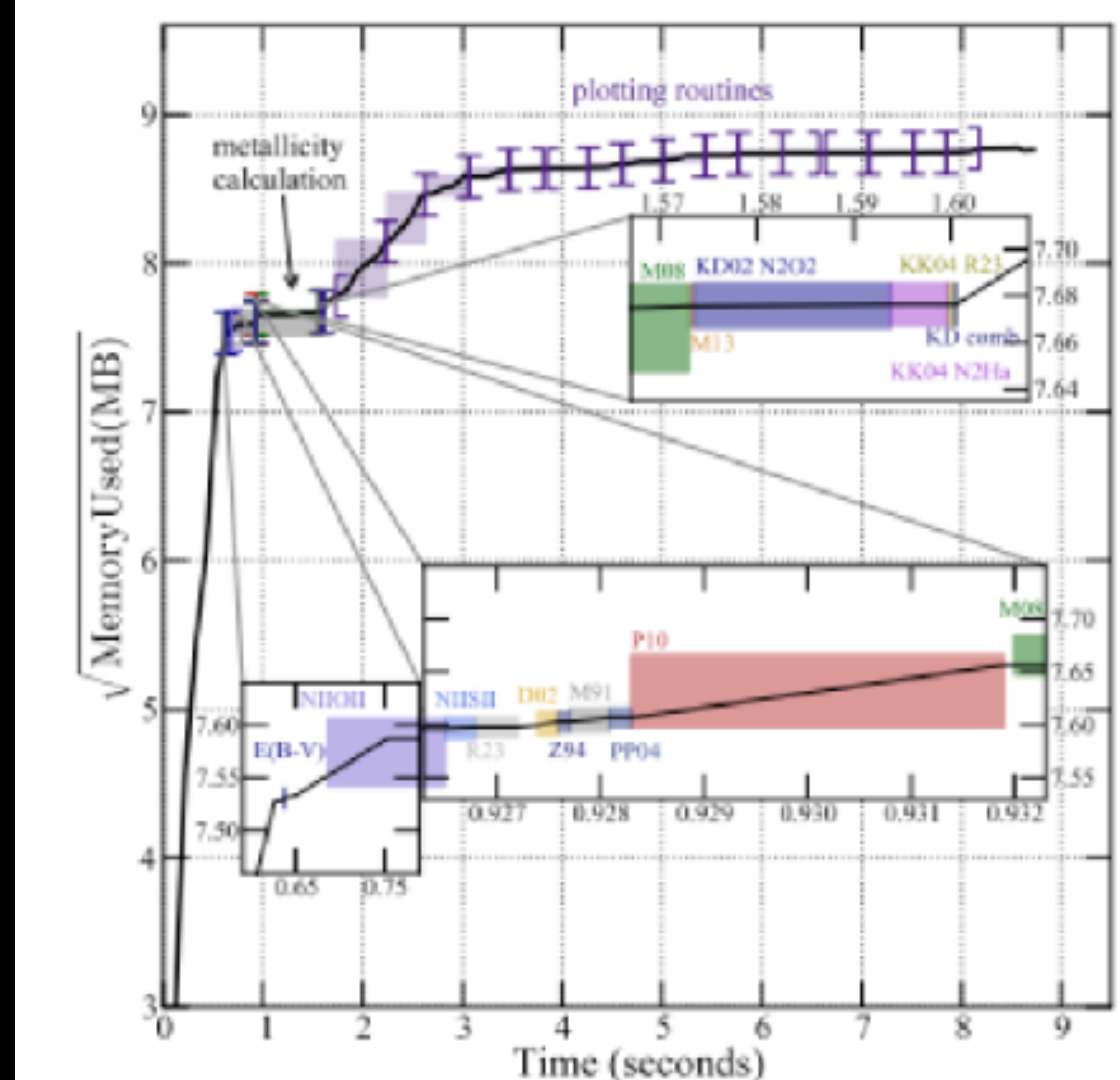


FIG. 6.— Memory usage: we plot the square root of the memory usage in Megabytes as a function of time for running our code (using N=2,000 and all default metallicity scales except the D13 *pyqz* ones) on a single set of measured emission lines (Table 2, host galaxy of SN 2008D). The square root is plotted, instead of the natural value, to enhance visibility. Three inserts show the regions where most of the metallicity scales are calculated, zoomed in, since the run time of the code is dominated by plotting routines, including the calculation of the bin size with Knuth's rule. Each function call is represented by an opening and closing bracket in the main plot, and by a shaded rectangle in the zoomed-in insets. The calculation of $N2O2$, which requires 0.25 seconds, is split be-

Reading:

An excellent use of viz for data exploration and transition to inferential analysis

https://blog.data.gov.sg/how-we-caught-the-circle-line-rogue-train-with-data-79405c86ab6a#.iz1r655xo

Lee Shangqian, Daniel Sim & Clarence Ng

Distance measures for clustering:

http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/tutorials/mvahtmlnode79.html

Decision trees:

http://what-when-how.com/artificial-intelligence/decision-tree-applications-for-data-modelling-artificial-intelligence/

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4380222/

Efficient python coding:

https://wiki.python.org/moin/PythonSpeed/PerformanceTips