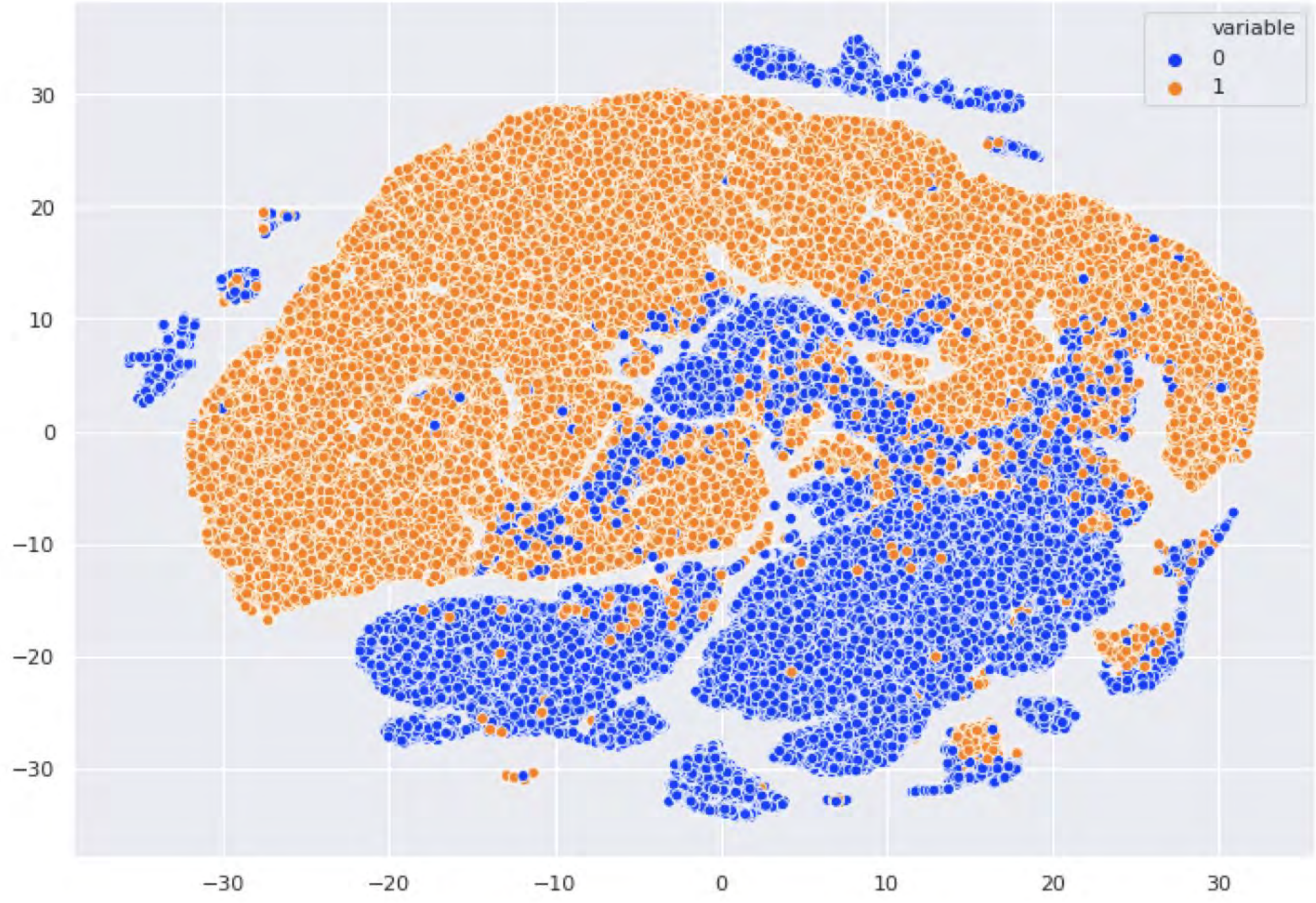
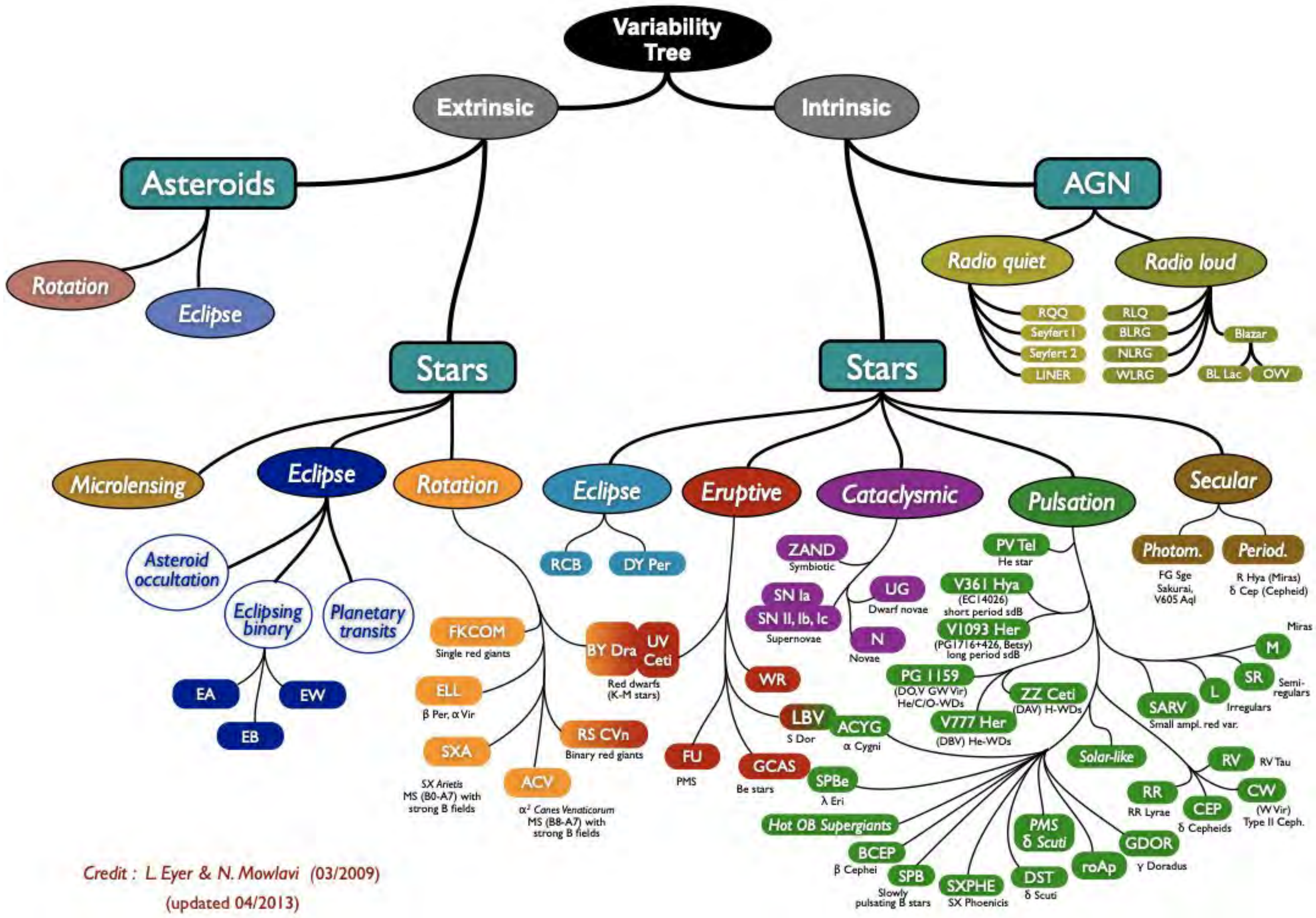
Classification



Ashish Mahabal
<ashish@caltech.edu>
AY 119 2025-04-15

Topics

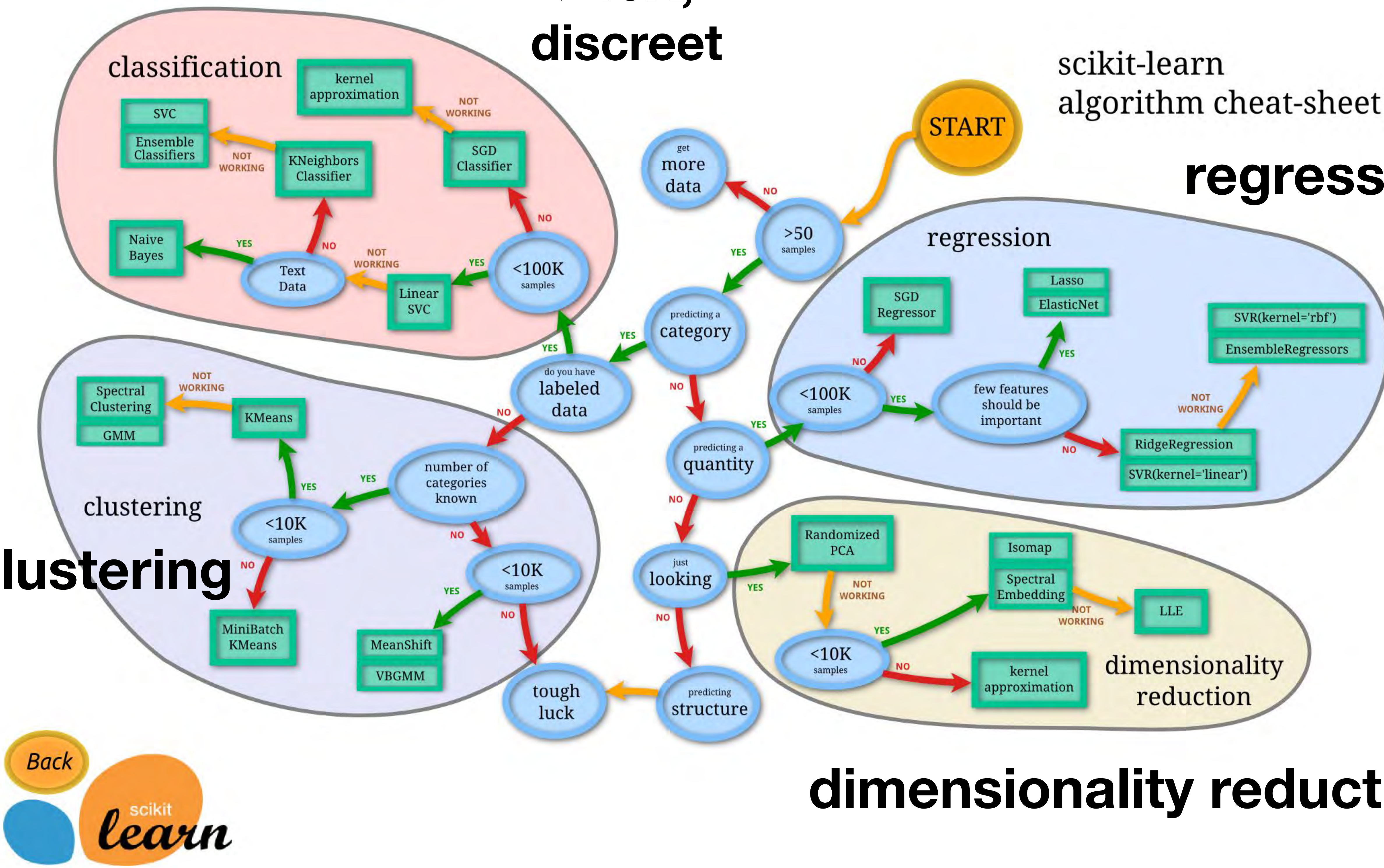
- Supervised classifications (labeled data)
    - Number of classes
        - Binary classifiers
        - Multi-class classifiers
    - Noise
    - Thresholds
    - Metrics
    - Anomalies
    - Classification algorithms
- Unsupervised classification (unlabelled data)



classification

labels, >10K, discreet

regression



clustering

dimensionality reduction

Labeled data, versus continuous variables

A few simple tools for Unsupervised data

number of classes generally not known

- **Self Organizing Maps (SOM)**
- **t-SNE**
- **UMAP**

Simple classification problem

Determine the number of classes

Determine the number of classes

Stars
Galaxies

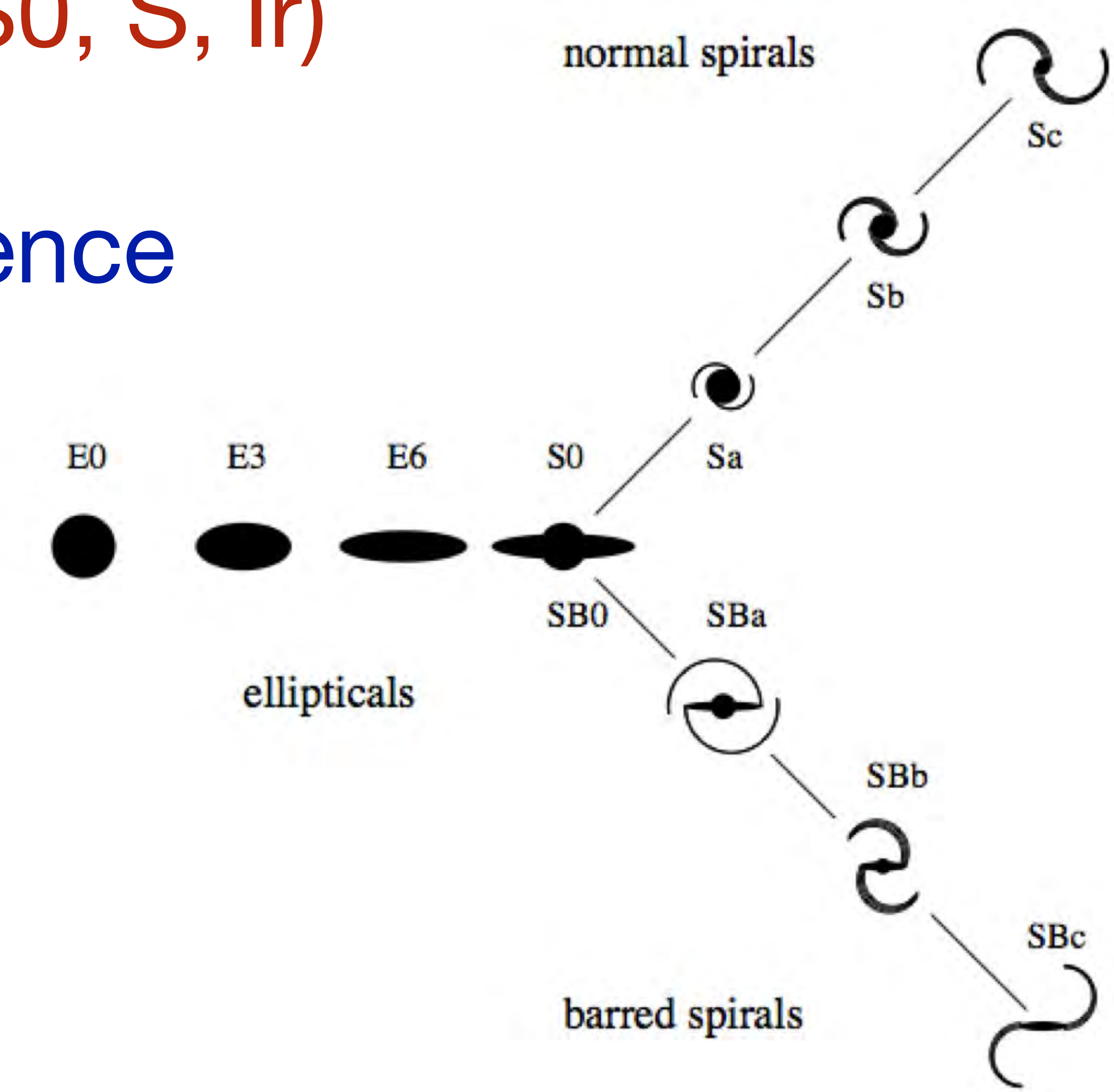
Possible complications

Star - galaxy

Galaxy - galaxy (E, S0, S, Ir)

Quasar - star

Dwarfs - main sequence



Understand their properties

Determine the number of classes

Understand their properties

- Extendedness
- Light concentration

Determine the number of classes

Understand their properties

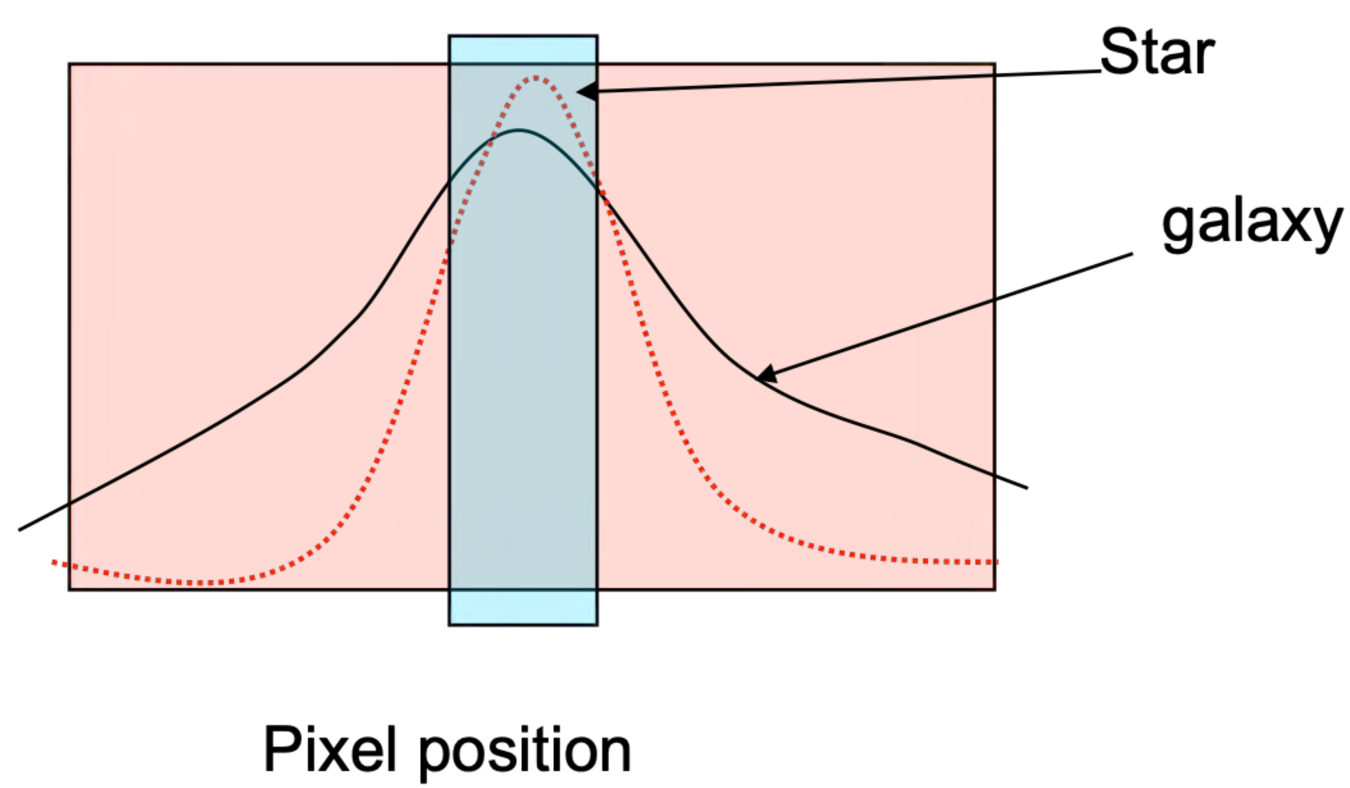
Measure parameters that are handles for these properties

Determine the number of classes

Understand their properties

Measure parameters that are handles for these properties

- Pixels occupied
- Flux in two apertures



Determine the number of classes

Understand their properties

Measure parameters that are handles for these properties

Plot the parameters

Determine the number of classes

Understand their properties

Measure parameters that are handles for these properties

Plot the parameters

“Separate” the clusters

Classification is an integral part of Astronomy

Clustering is the means to separate the classes

Complications

How many classes are there?

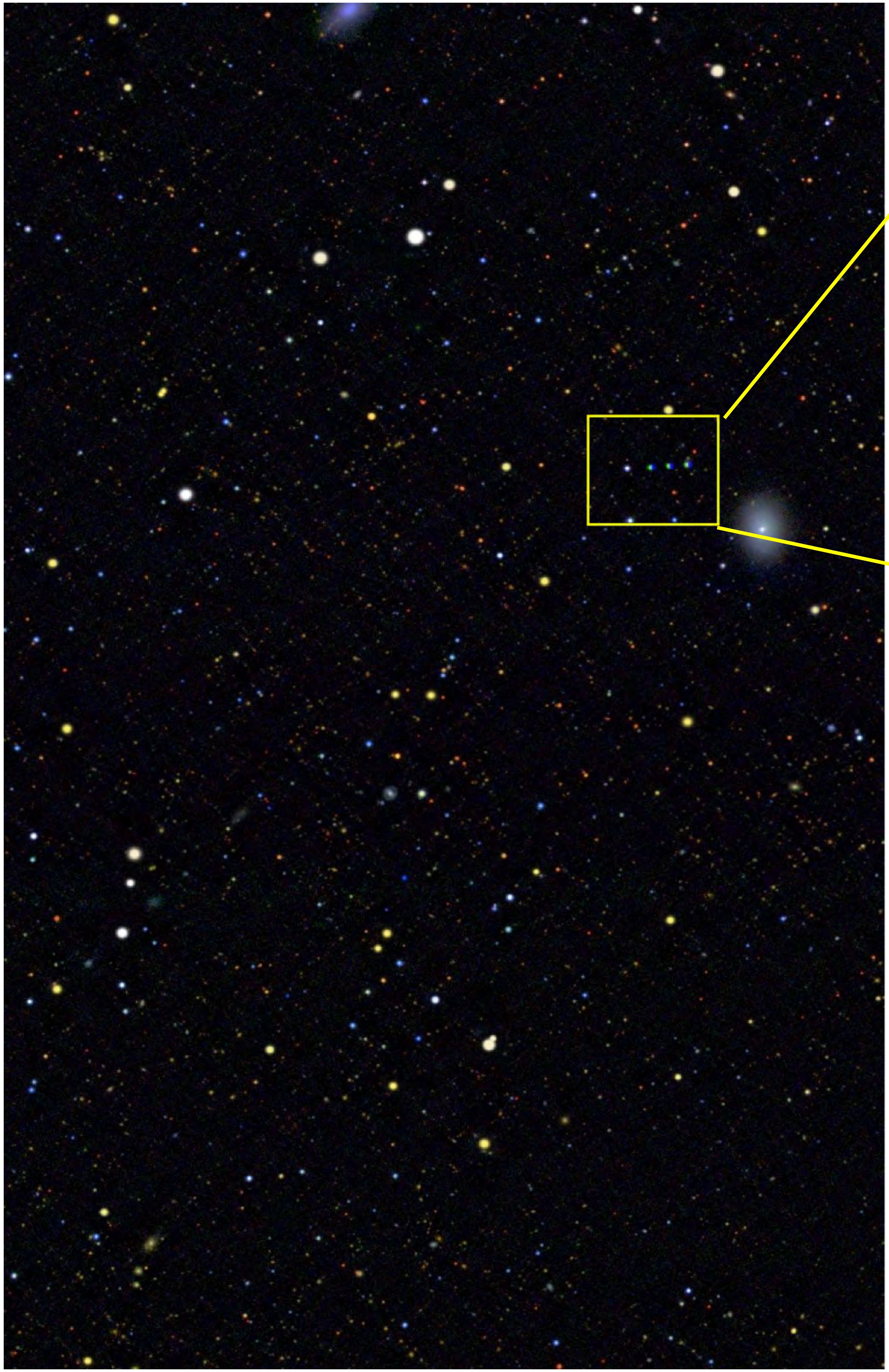
Complications

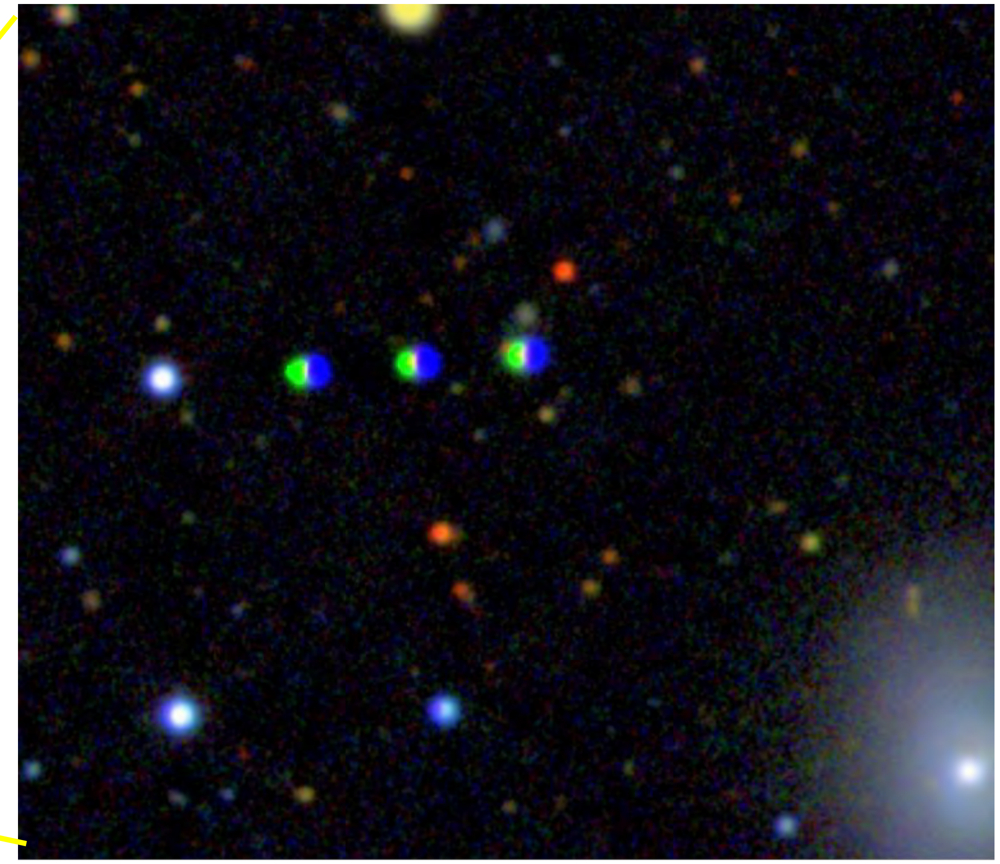
How many classes are there?

Just stars and galaxies?

Simple classification problem

Stars
Galaxies
CCD defects
Cosmic rays
Bleed trails
Satellite trails

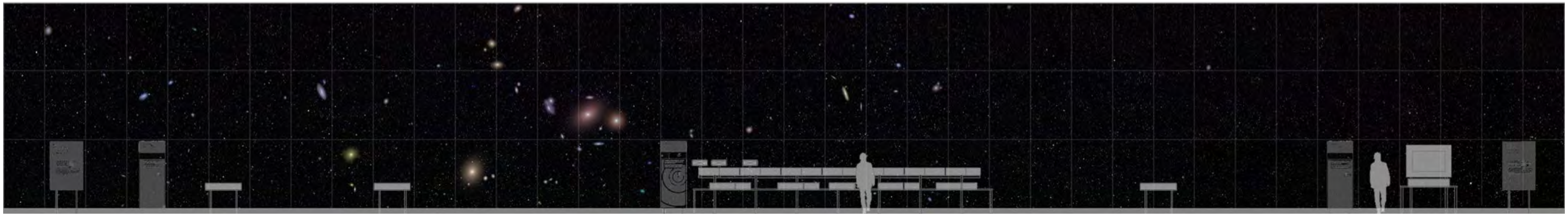




Asteroids in the Big Picture made for Griffith Observatory

Complications

How many classes are there?
Are they cleanly separated?



Complications

How many classes are there?

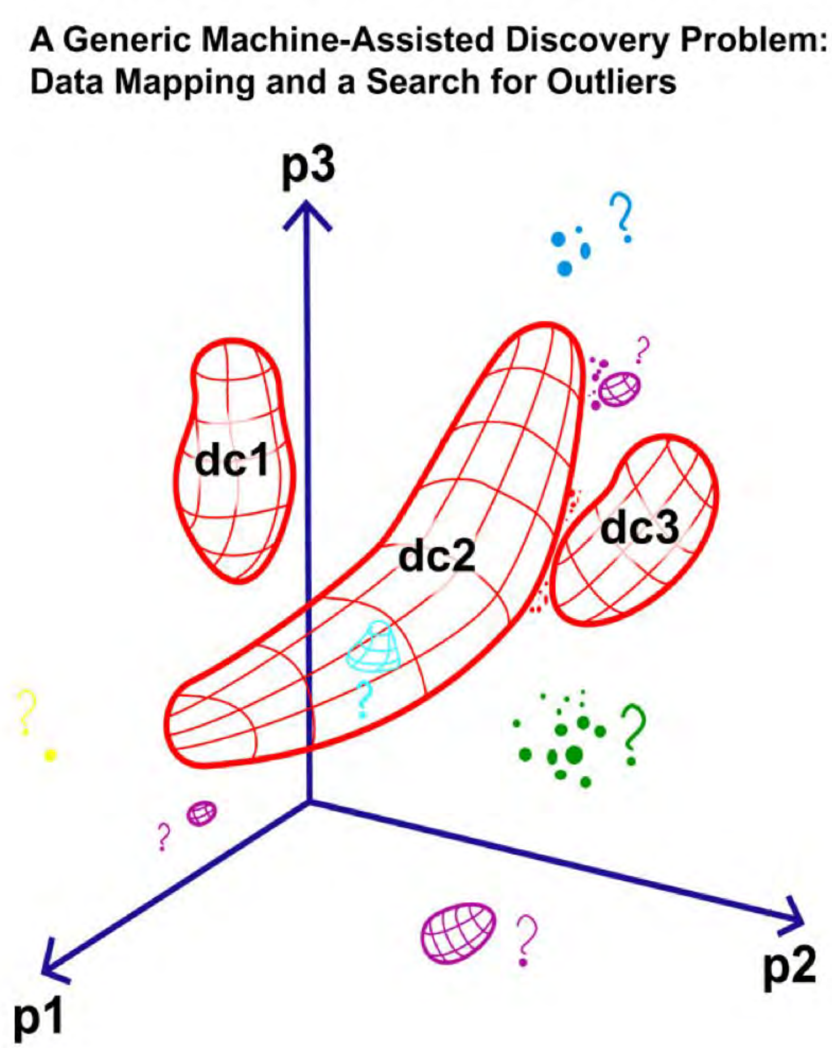
Are they cleanly separated?

- Brighter stars
- Distant galaxies
- Grazing cosmic rays

image from JWST

Complications

How many classes are there?
Are they cleanly separated?
Do all objects belong to these classes?

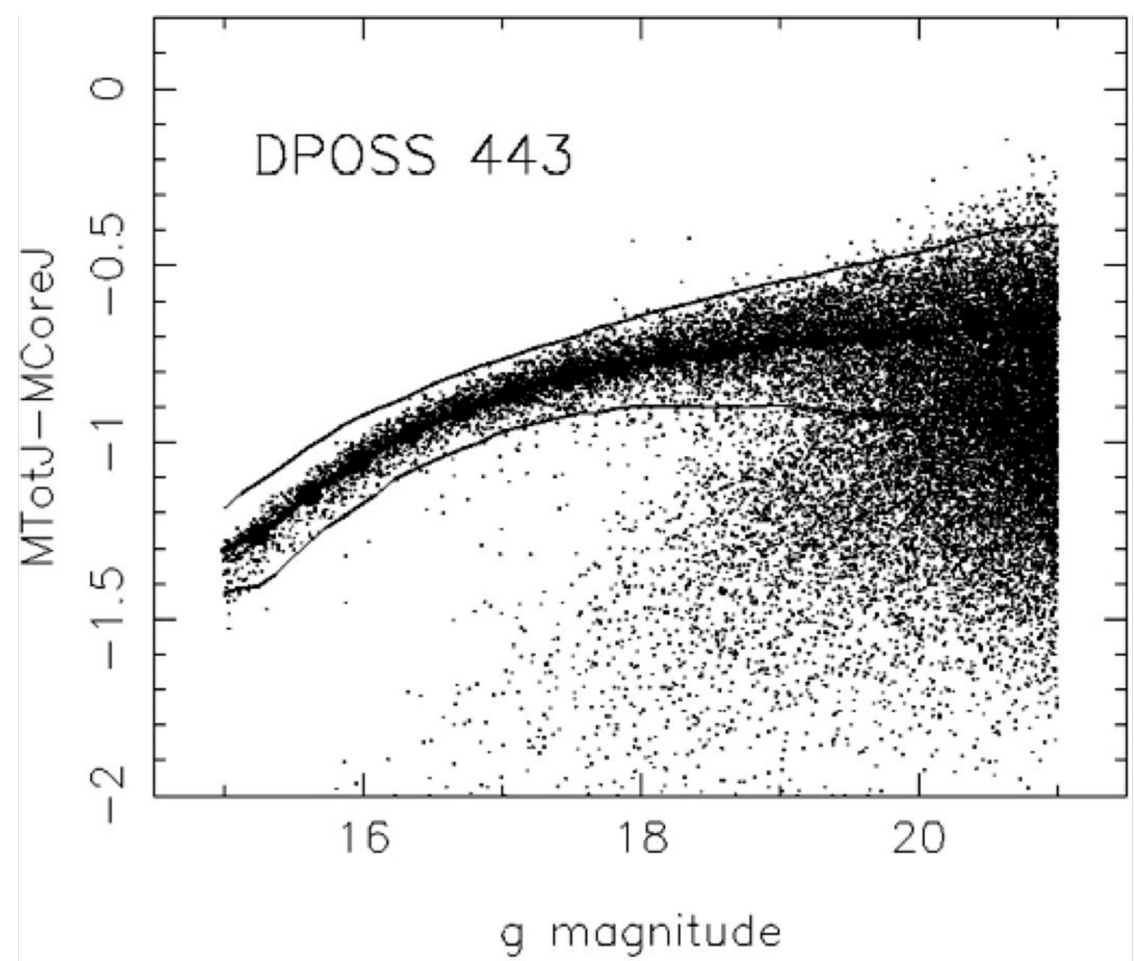


Complications

How many classes are there?
Are they cleanly separated?
Do all objects belong to these classes?
Could we add observables to classify better
and find rarer objects?

Complications

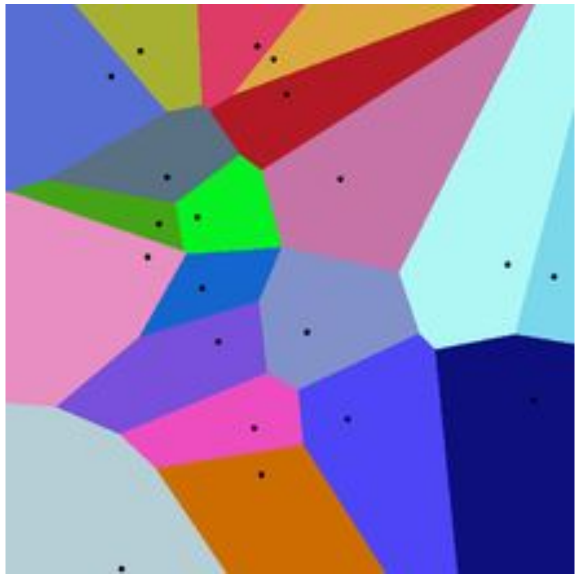
How many classes are there?
Are they cleanly separated?
Do all objects belong to these classes?
Could we add observables to classify better
  and find rarer objects?
  Another waveband?
  A third one?
  More epochs?



This is where we perhaps get in to supervised classification

K-means

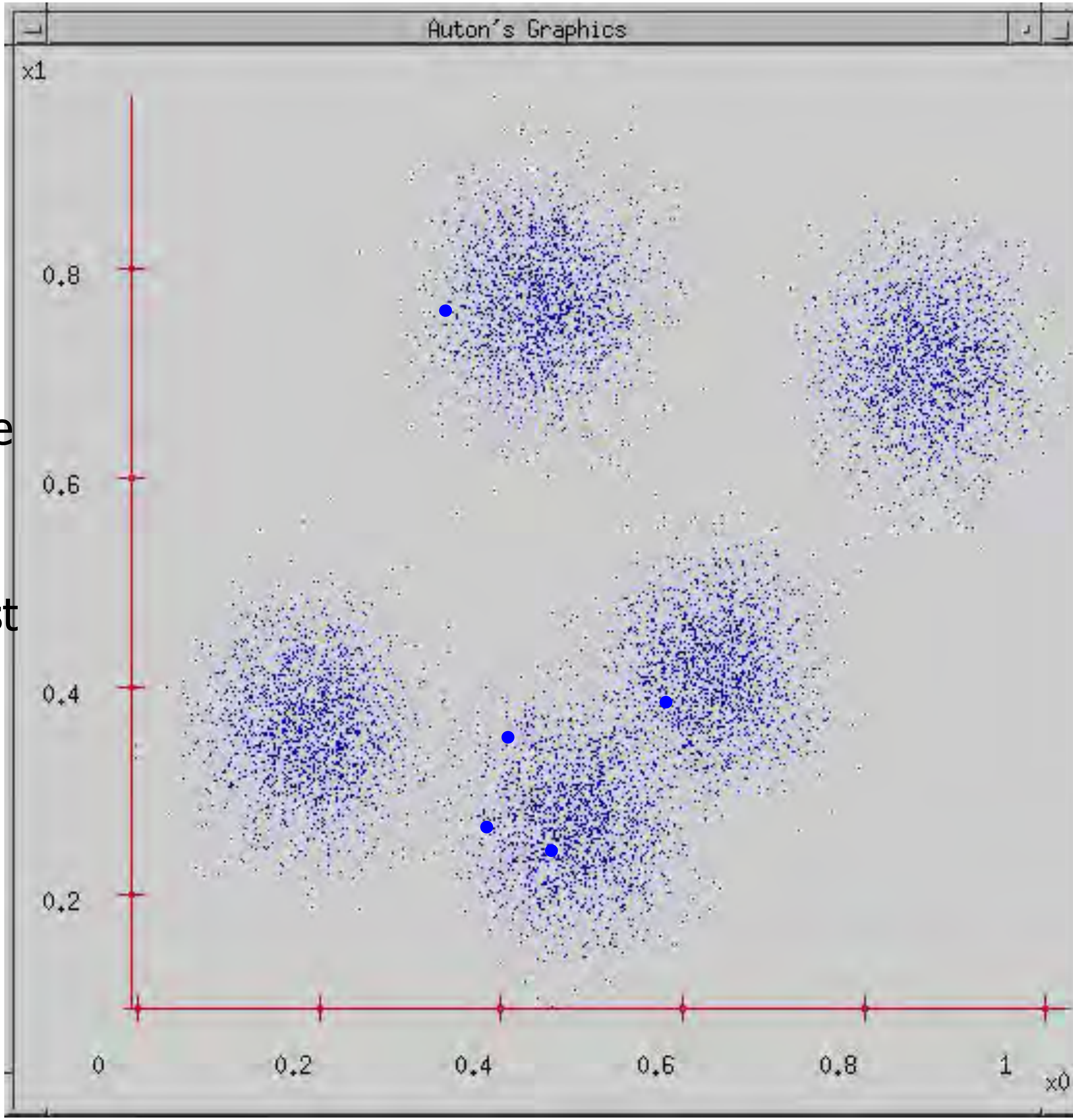
$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$



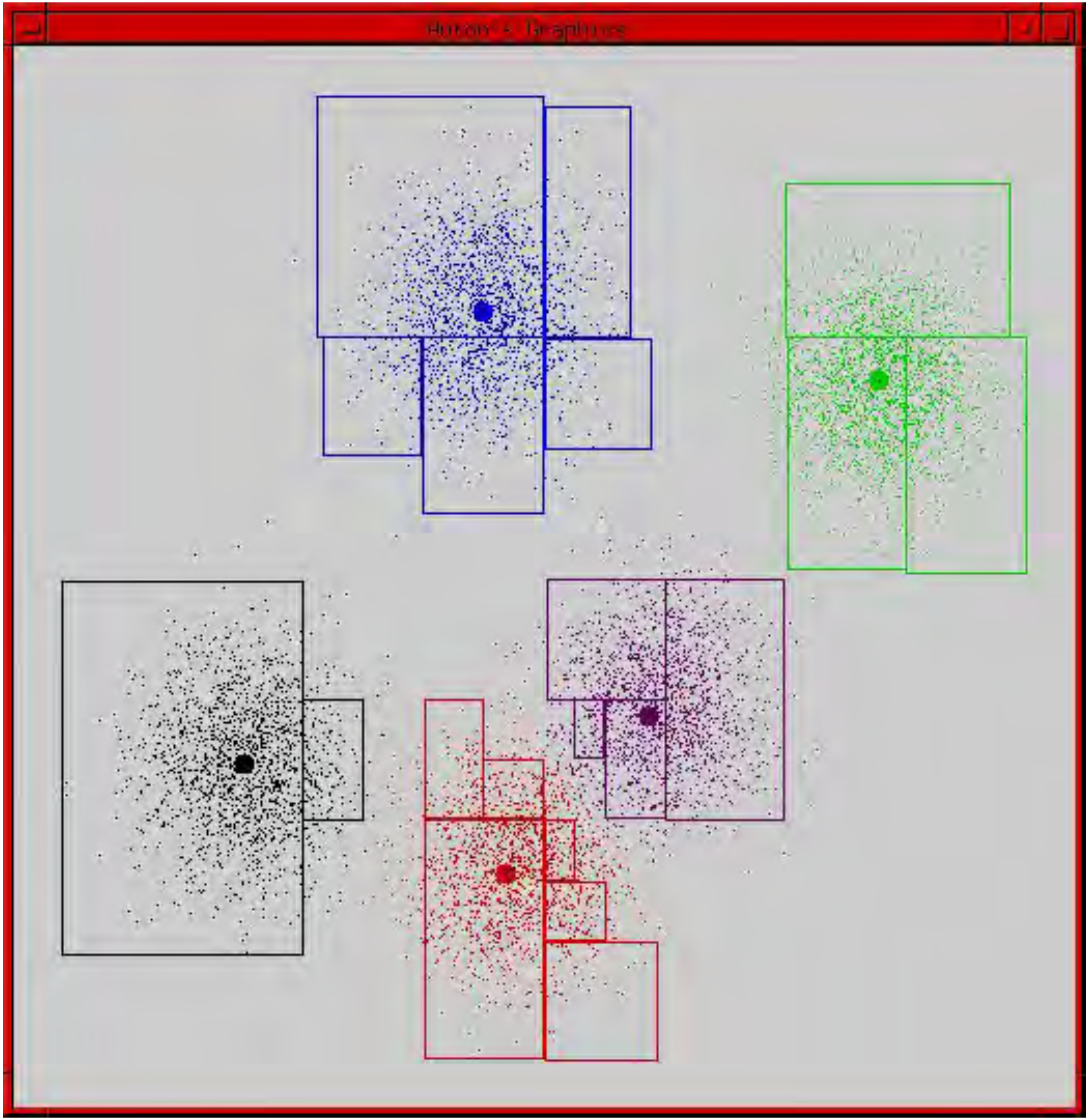
- Clustering
- heuristic algorithm
- start with guessing cluster centers
- similar spatial extent clusters
- k-clusters {S_1,…, S_k}
- n-dim space
- minimize distance squares within a cluster
- NP-hard O(n^{dk+1} log n)

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly (?) guess k cluster Center locations (alternative methods)
3. Associate each data point with its nearest center (reduces WCSS)
4. Compute the new mean centers
5. Iterate until some convergence criterion is reached



K-means terminates



K-means steps

1. **Initialization:**

   Choose the number of clusters K. Randomly select K initial centroids from the dataset.

2. **Assignment Step:**

   Assign each data point x_i to the nearest centroid based on Euclidean distance.

$$\text{Cluster assignment for point } x_i = \arg\min_k \|x_i - \mu_k\|^2$$

   where:

   - x_i is a data point.
   - μ_k is the centroid of cluster k.
   - $\|\cdot\|$ is the Euclidean norm.

K-means steps

3. **Update Step:**

   Update the position of each centroid to the mean of all data points assigned to it:

   $$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

   where:

   - C_k is the set of data points assigned to cluster k.
   - $|C_k|$ is the number of data points in cluster k.

4. **Repeat:**

   Repeat Steps 2 and 3 until convergence. Convergence typically occurs when centroids no longer change significantly, or cluster assignments become stable.

K-Means Objective Function (Cost Function):

The algorithm minimizes the within-cluster sum of squared errors (also called inertia):

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

- J: total within-cluster variance.
- The lower J is, the more compact the clusters are.

Choosing the Number of Clusters (K):

- **Elbow Method:**

  Plot the within-cluster sum of squares against different values of K. Choose the K at the "elbow" (where the decrease rate sharply reduces).

- **Silhouette Score:**

  Measures how similar an object is to its own cluster compared to other clusters (higher is better).

$$\text{Silhouette Score for point } i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

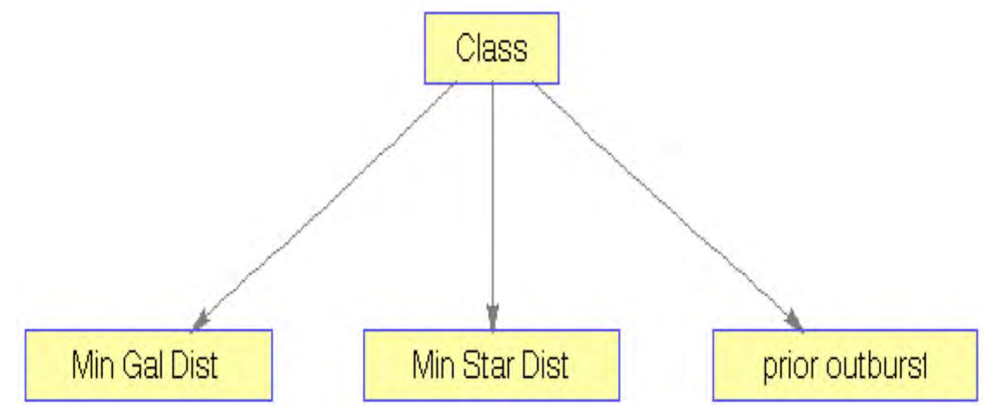
  where:

  - a_i: Mean intra-cluster distance (to own cluster points).
  - b_i: Mean nearest-cluster distance (to points in the next closest cluster).

K-means exercise

At the end of the class

Few supervised classifiers

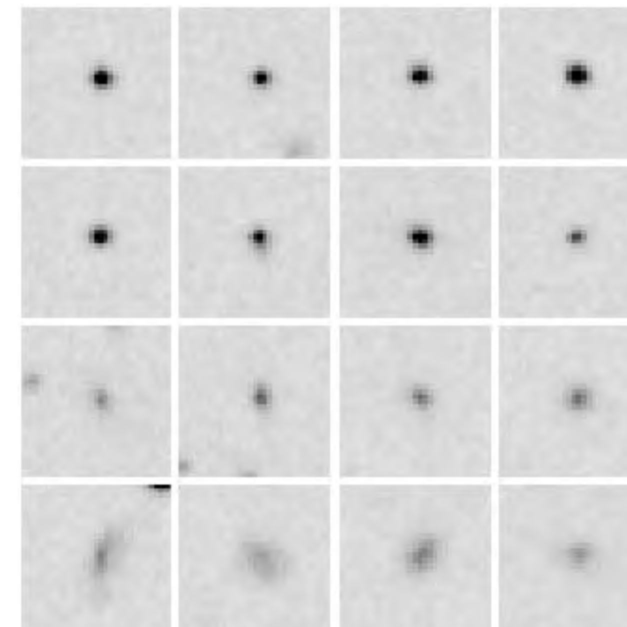


- Support vector machines
- Naive Bayes
- Logistic/Linear Regression
- Decision Trees
- Random Forests
- Neural Networks
- Convolutional Neural Networks

Aspects of training data

Nature of input data
Images
Features
Dimensionality

DPOSS



Number of classes
Differentiability
Balancedness
Overwhelming class?

Bias-Variance tradeoff

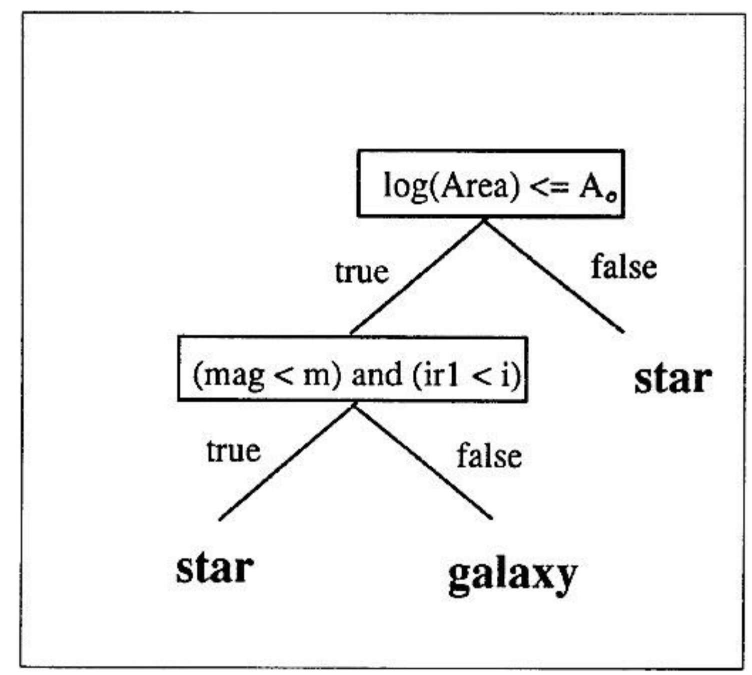


FIG. 1. In this sample decision tree, one starts at the top node(root), following the appropriate path to a final leaf (class) based upon the truth of the assertion at each node.

Weir et al. 1995

Labeled training set

$$x_i = i^{th}\,Featurevector$$

$$y_i = i^{th}\,Label$$

N training examples $\longrightarrow$ $\{(x_i, y_i), \ldots, (x_n, y_n)\}$

$$g : X \rightarrow Y$$

$$g(x) = \arg \max_{y} f(x, y)$$

f: scoring function from the space F

Loss minimizing map

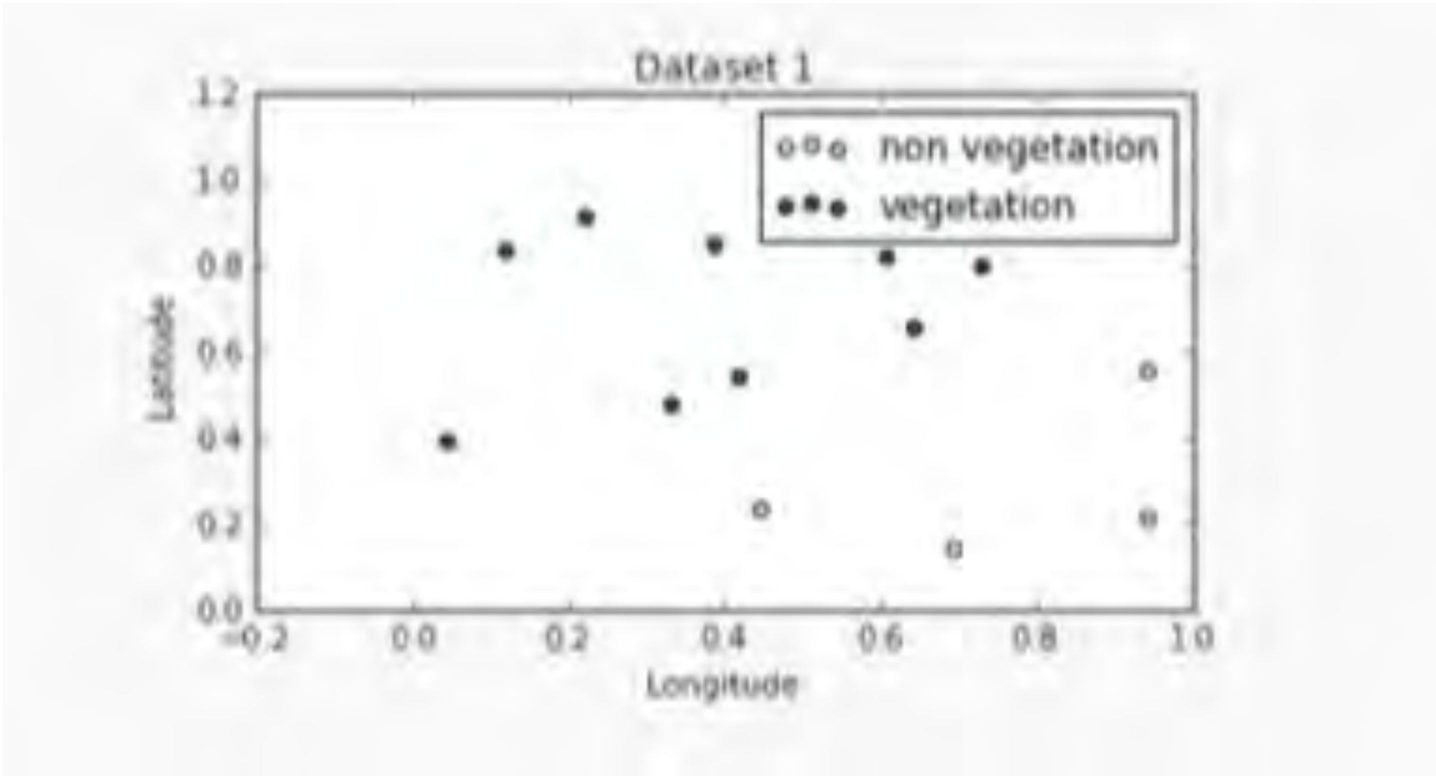
Loss mapping $\longrightarrow$ $L : Y \times Y$

Loss for a single example $(x_i, y_i) : \hat{y} = L(y_i, \hat{y})$

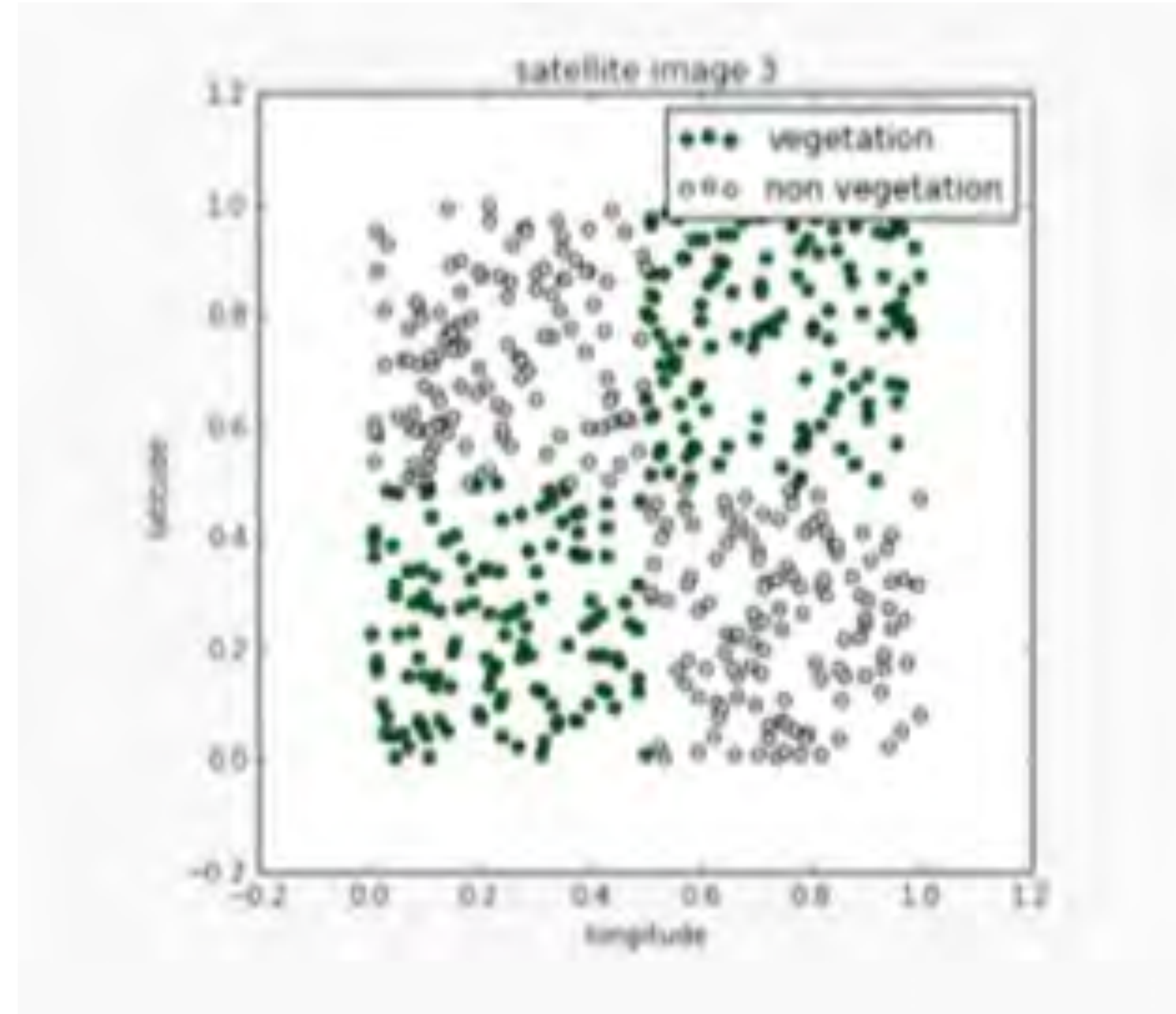
Minimize:

$$R(g) = \frac{1}{N} \sum_{i}^{N} L(y_i, g(x_i))$$

if many possible functions, or few examples, overlearning happens



linear separability

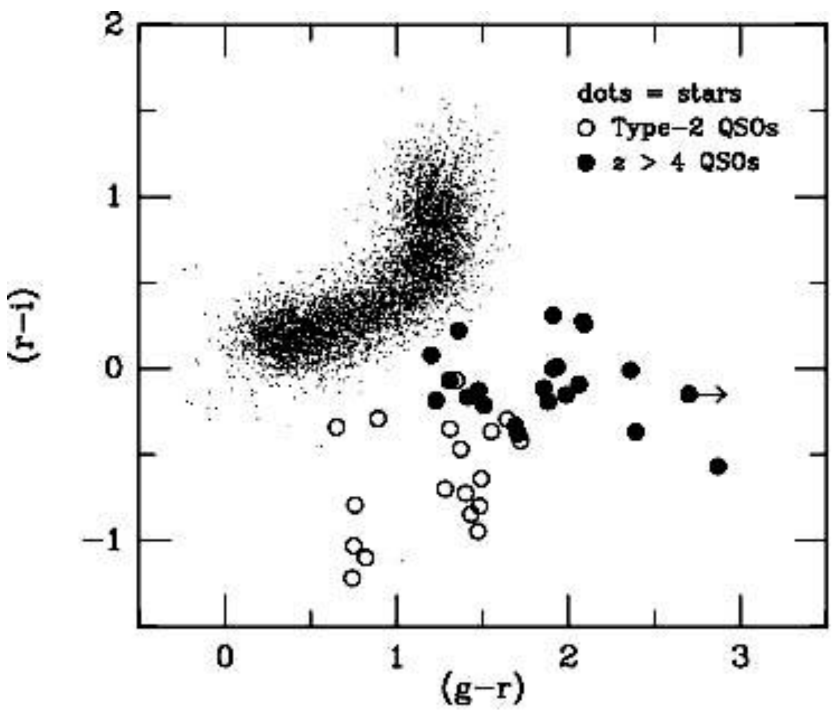


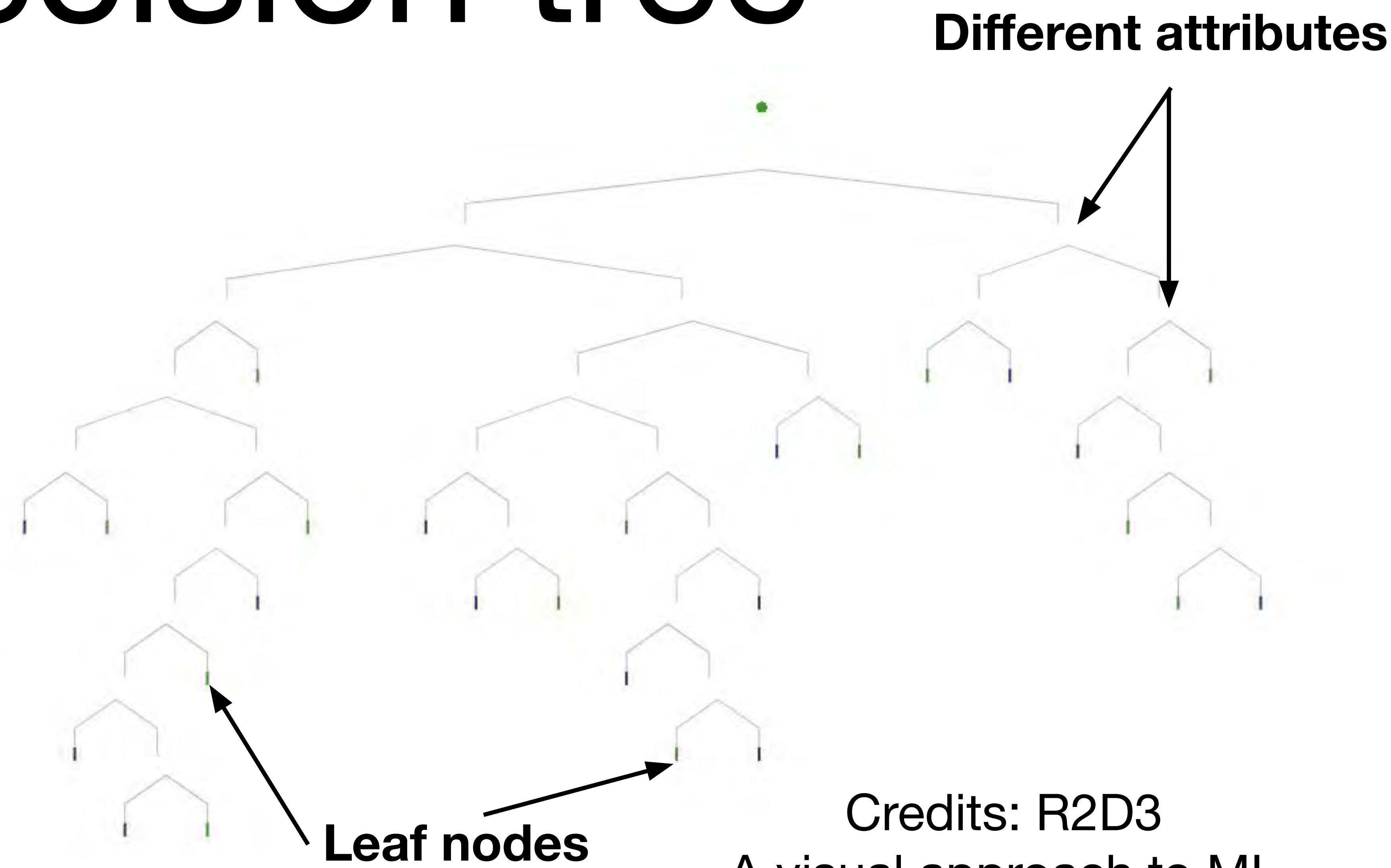
Well separated in
feature space, but
not linearly separable

For interpretable real life models, we need
- multiple decision boundaries
- locally linear

(Many slides from
Pavlos Protopapas:
La Serena Data Science)

Decision tree

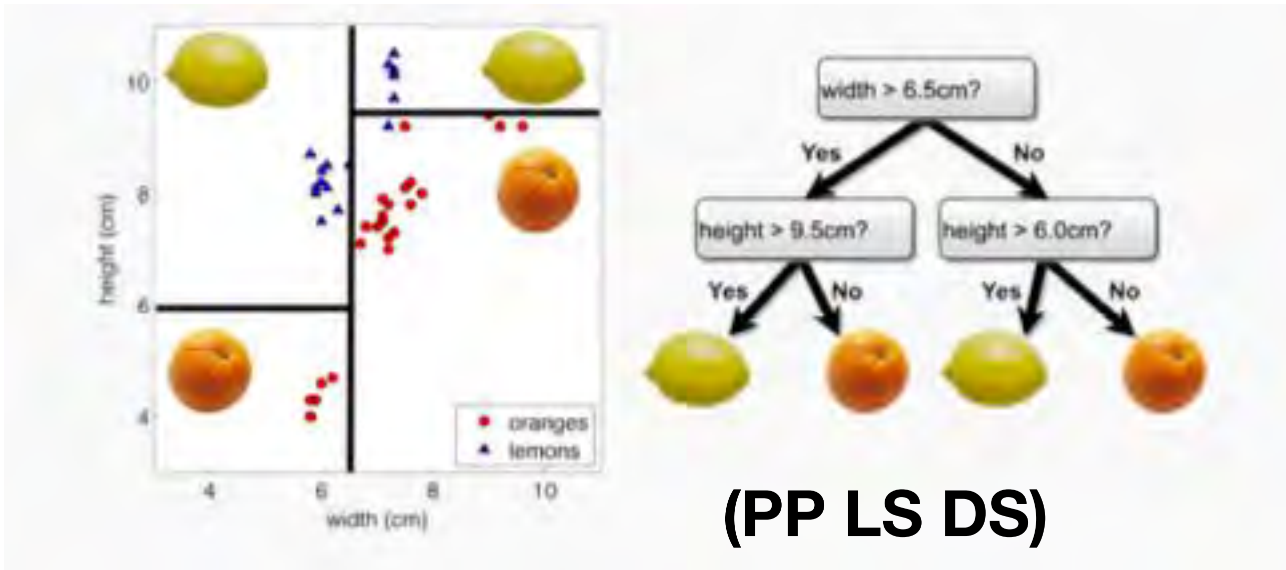




Credits: R2D3
A visual approach to ML

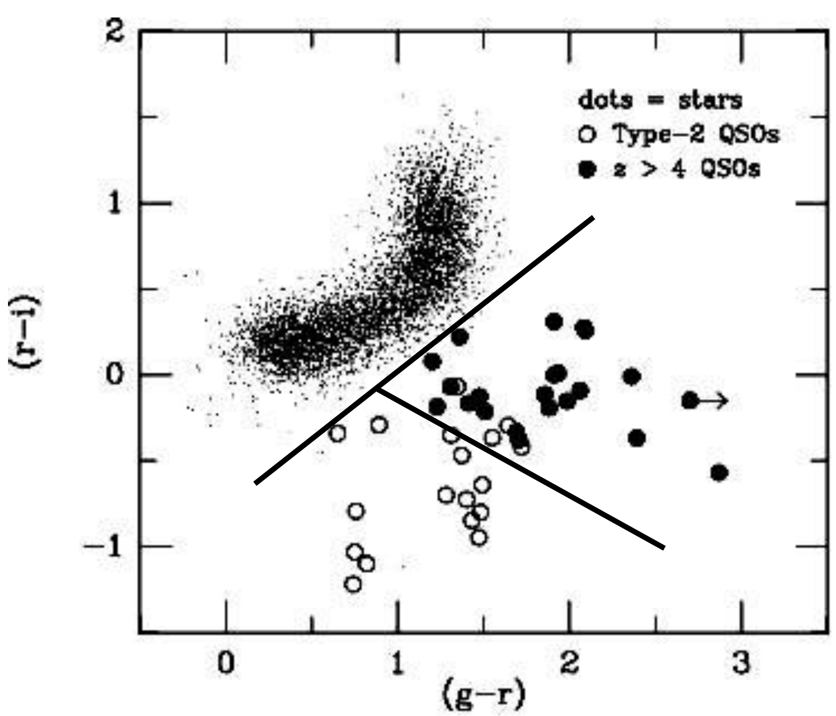
- How to choose the parameters/observables?

- How to decide on the decision boundaries?



(PP LS DS)

Boundaries parallel to measurables

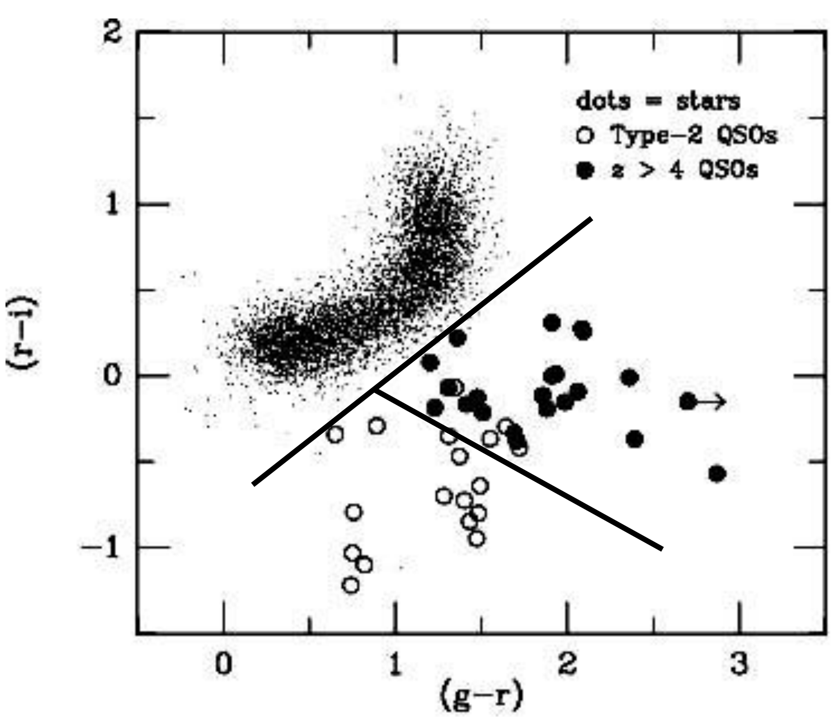


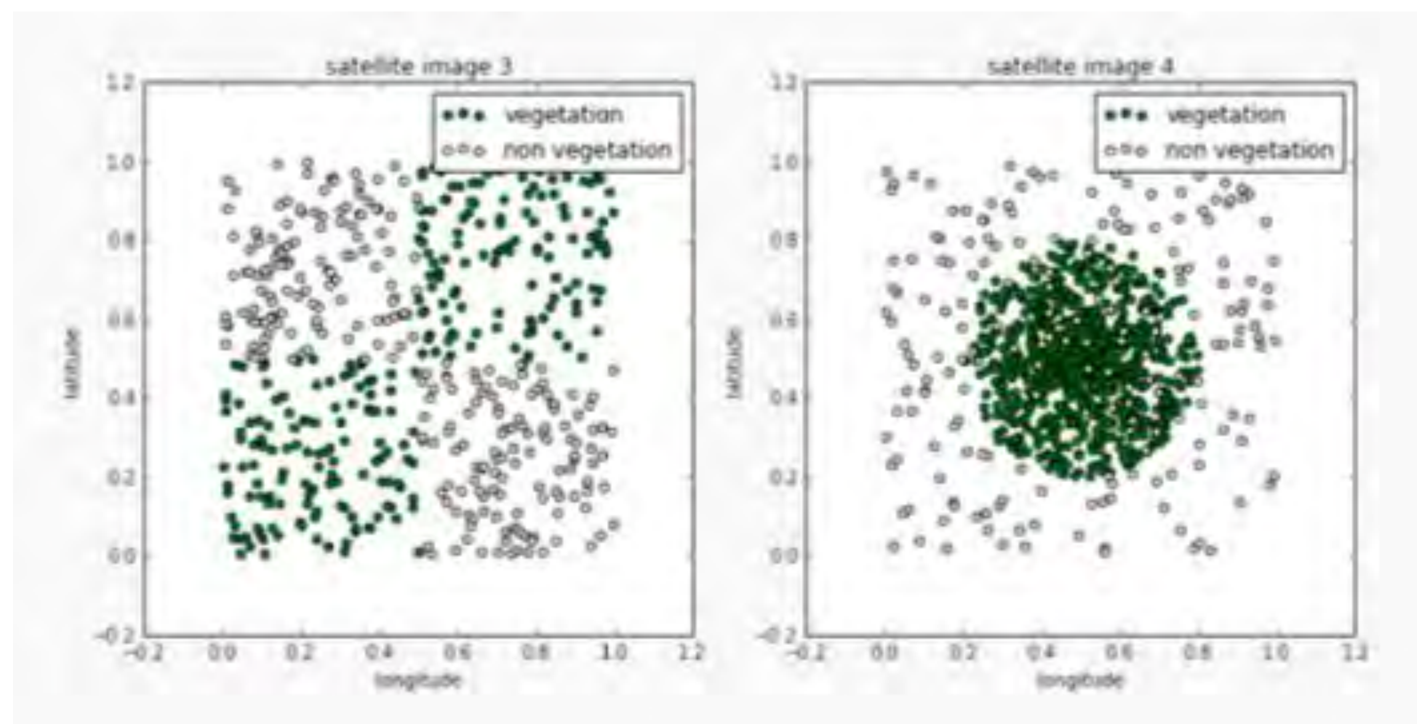
Boundaries not parallel to measurables

Exercise: determine the decision boundaries here

examples of class boundaries

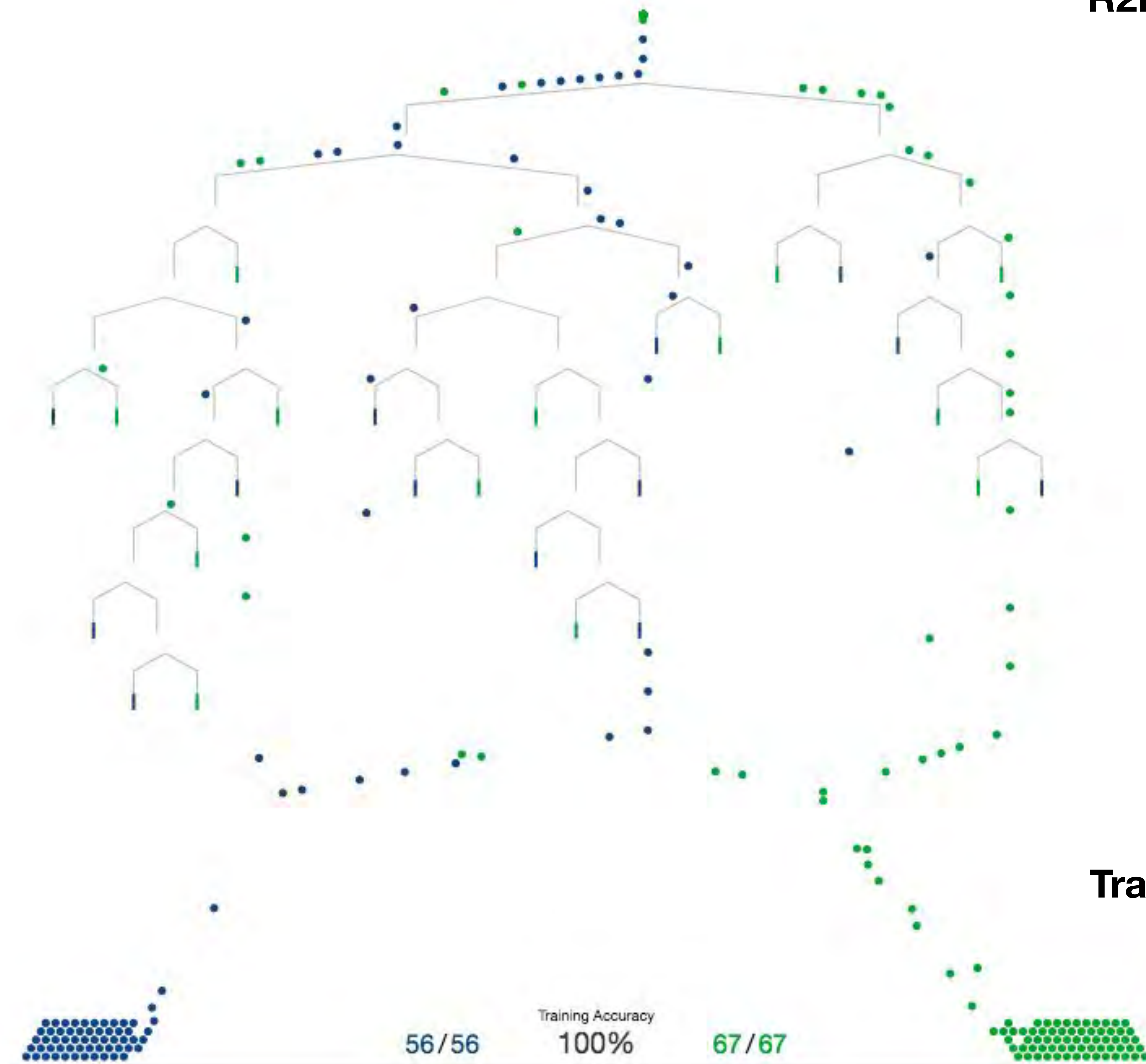
- separable by a line (quasars, dwarfs)
- separable but by a circle (radial coordinates)
- making decision boundaries (locally) linear when possible
- and human interpretable





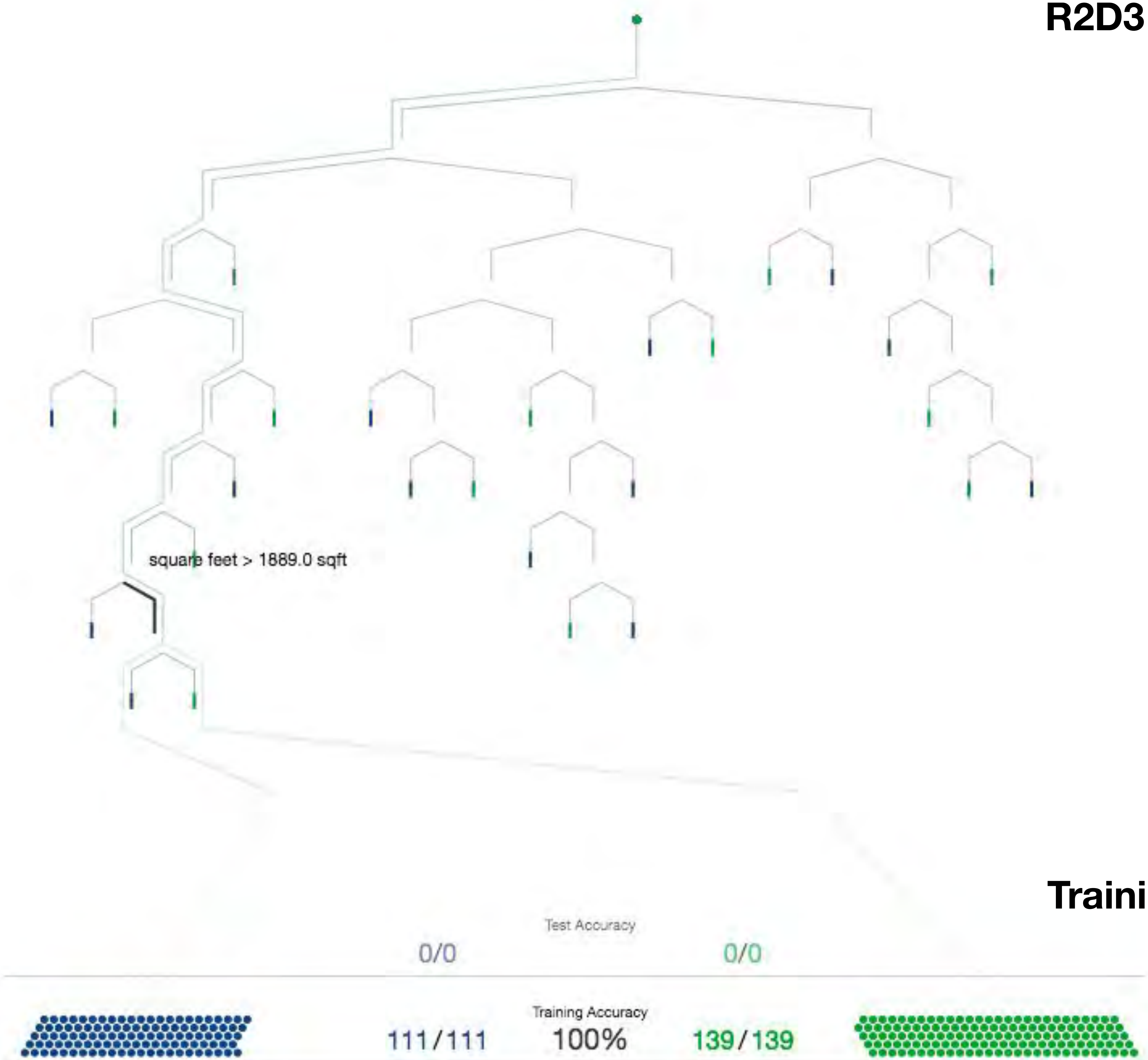
Pavlos Protopapas, La Serena data Science

R2D3 Viz



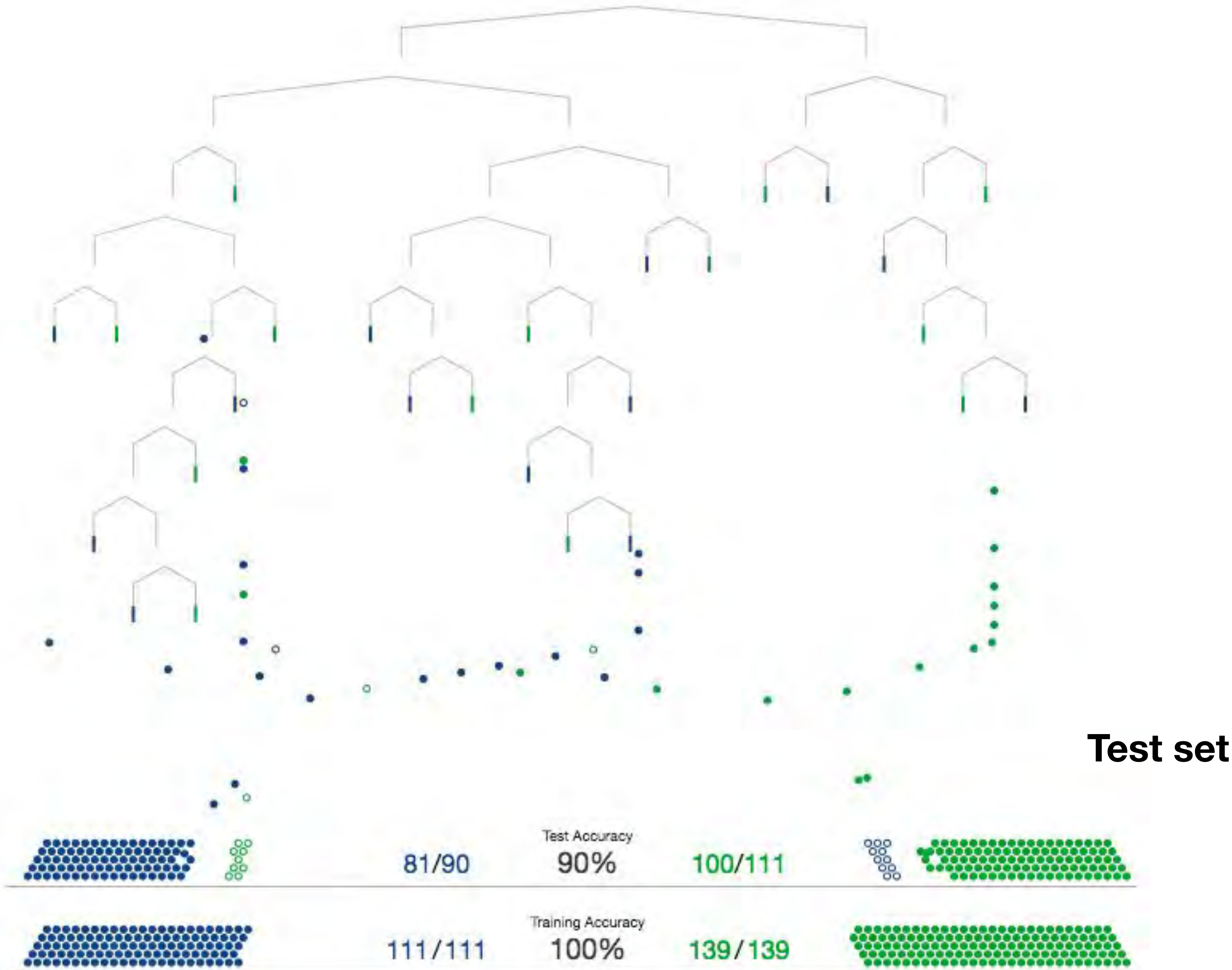
Training set

R2D3 Viz



Training set

R2D3 Viz


Test set
Test Accuracy
81/90
90%
100/111
Training Accuracy
111/111
100%
139/139

Overfitting - a pathology

- Using too many boundaries, or boundaries that distinguish inconsequential differences.

- More of these are encountered in deep learning (and also random forests)

Pop quiz: How many leaf nodes should there be?

Model Flow/Learning

- Empty decision tree = undivided feature space
- Choose optimal predictor to split, and an optimal threshold
- Recurse on each node until a stopping condition is met

Pop quiz: if your variable is categorical, how do you choose a threshold?
(e.g. 'starriness' when considering stars and galaxies)

What issues crop up if there are more than two such classes?

Splitting criteria

- feature space should grow purer
- fitness metric should be differentiable
- no empty regions should be created

Classification error

Suppose we have J number of predictors and K classes.

Suppose we select the j-th predictor and split a region containing N number of training points along the threshold $t_j \in \mathbb{R}$.

We can assess the quality of this split by measuring the classification error made by each newly created region, R_1, R_2:

$$\text{Error}(i|j, t_j) = 1 - \max_k p(k|R_i)$$

where $p(k|R_i)$ is the proportion of training points in R_i that are labeled class k.

PP LS DS

Classification error

Example

	Class 1	Class 2	Error$(i\|j, t_j)$
R_1	0	6	$1 - \max\{6/6, 0/6\} = 0$
R_2	5	8	$1 - \max\{5/13, 8/13\} = 5/13$

We can now try to find the predictor j and the threshold t_j that minimizes the average classification error over the two regions, weighted by the population of the regions:

$$\min_{j,t_j} \left\{ \frac{N_1}{N} \text{Error}(1|j, t_j) + \frac{N_2}{N} \text{Error}(2|j, t_j) \right\}$$

where N_i is the number of training points inside region R_i.

Gini index

Suppose we have J number of predictors, N number of training points and K classes.

Suppose we select the j-th predictor and split a region containing N number of training points along the threshold $t_j \in \mathbb{R}$.

We can assess the quality of this split by measuring the purity of each newly created region, R_1, R_2. This metric is called the ***Gini Index***:

$$\text{Gini}(i|j, t_j) = 1 - \sum_k p(k|R_i)^2$$

Gini index

Example

	Class 1	Class 2	Gini$(i\|j,t_j)$
R_1	0	6	$1-(6/6^2+0/6^2)=0$
R_2	5	8	$1-[(5/13)^2+(8/13)^2]=80/169$

We can now try to find the predictor j and the threshold t_j that minimizes the average Gini Index over the two regions, weighted by the population of the regions:

$$\min_{j,t_j}\left\{\frac{N_1}{N}\text{Gini}(1|j,t_j)+\frac{N_2}{N}\text{Gini}(2|j,t_j)\right\}$$

where N_i is the number of training points inside region R_i.

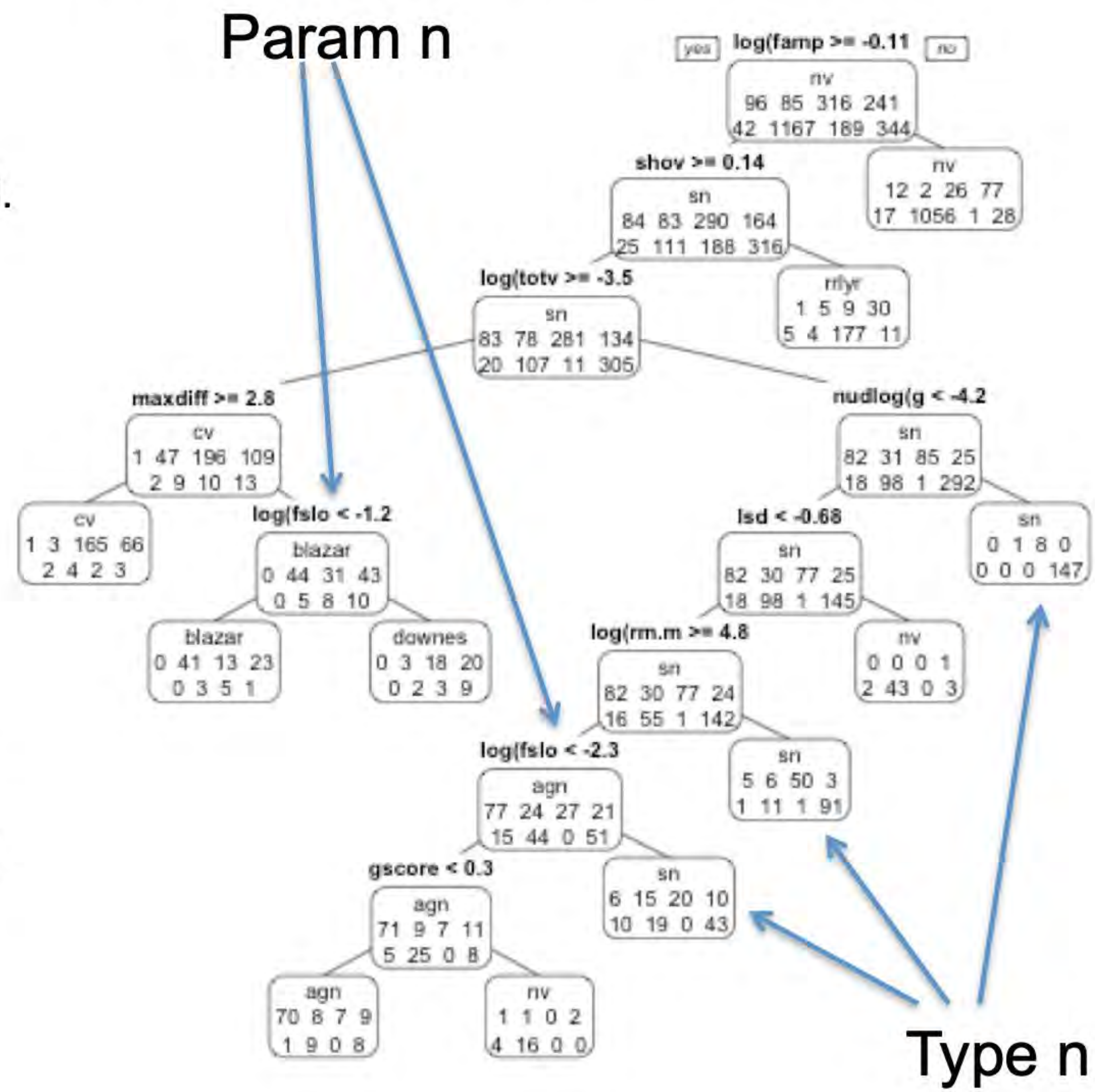
Stopping conditions

- don’t split if all examples are of one class
- don’t split if number of examples falls below pre-defined splitting threshold
- don’t split if number of leaves exceeds pre-defined threshold
- don’t split if class distribution is independent of predictors
- don’t split unless gain in purity based on some index like Gini is above pre-defined threshold

Recursive Partitioning

J Faraway, Mahabal et al.

arXiv:1401.3211



Numbers/ names not for reading

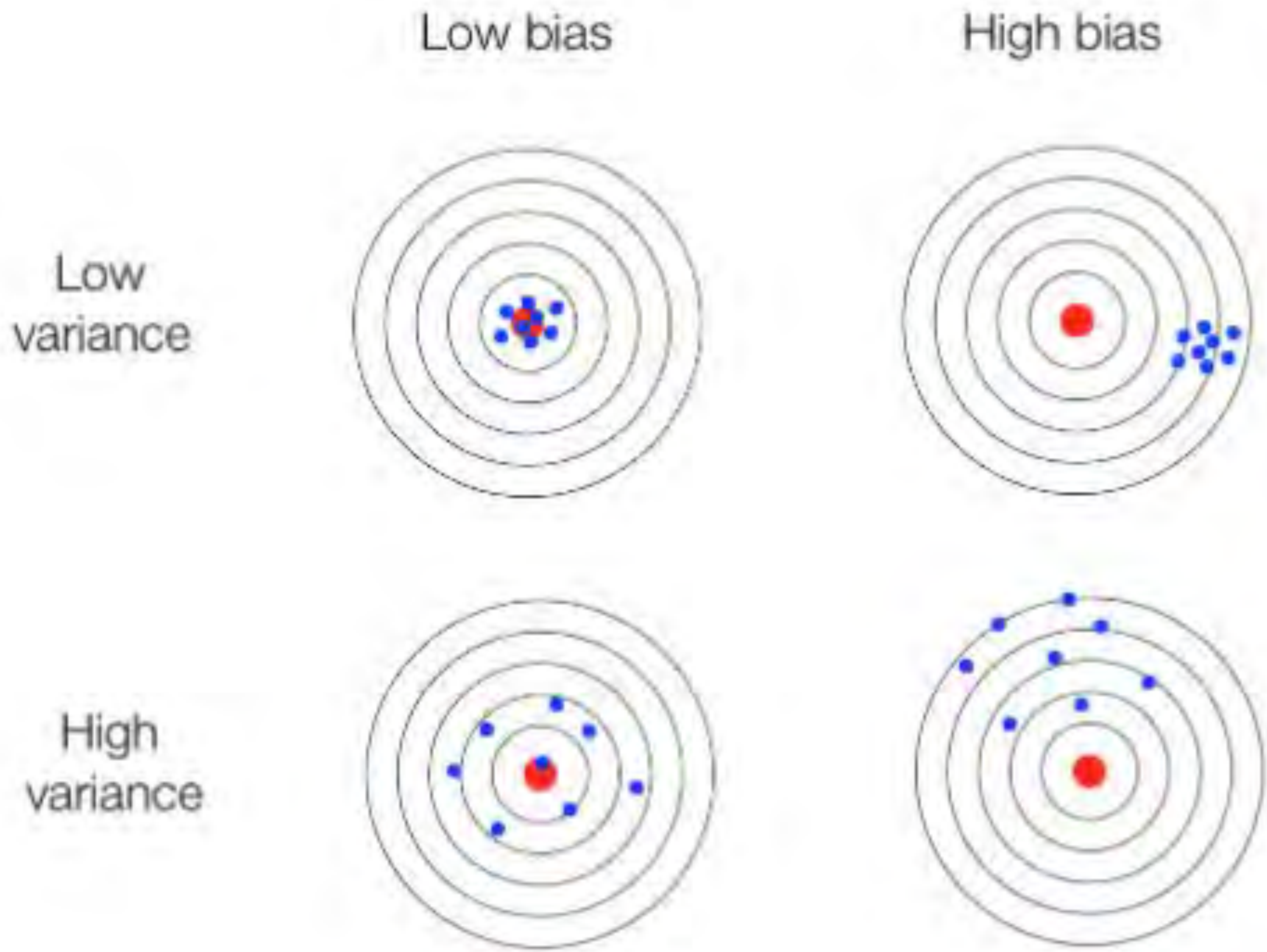
Problem with decision trees

- Bias versus variance
- Overfitting with large set of features and/or few examples

Solution

Averaging over partial sets
Randomly subsetting features

… broadly speaking



https://www.quora.com/What-does-Bagging-reduces-the-variance-while-retaining-the-bias-mean

Bagging

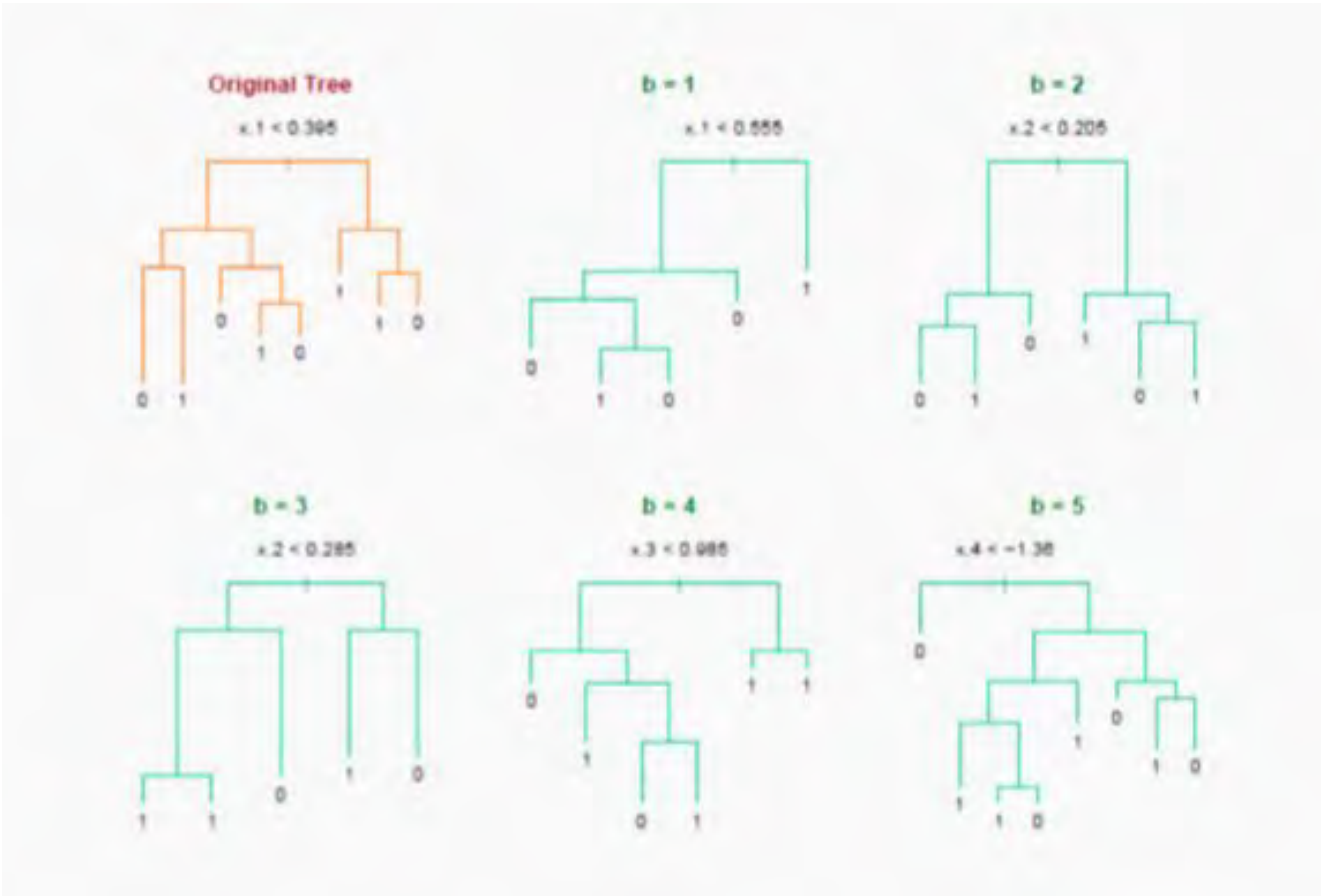
- Bootstrap to generate multiple samples, train the tree on each
- Aggregate the result of all trees for any given input

Bootstrap AGGrigatING - Breiman, 1996

Trees are fully expressive
Have reduced variance

Less interpretable (because mix of multiple trees)

Bagging



PP LS DS

Out of bag error

- For each point in training set, average predicted output over models whose training excluded this point (point-wise out-of-bag error)

- average the point-wise out-of-bag errors over the entire training set

minimize it

Improving on bagging

- Bagging is not optimal in the presence of strong predictors - all models will use it to split during early iterations giving rise to correlated trees.
- Thus trees will be identically distributed
- For B number of identically but not independently distributed variables with pairwise correlation ρ and variance σ^2, and variance of their mean is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Thus variance reduction is minimal (as B increases, only the second term vanishes)

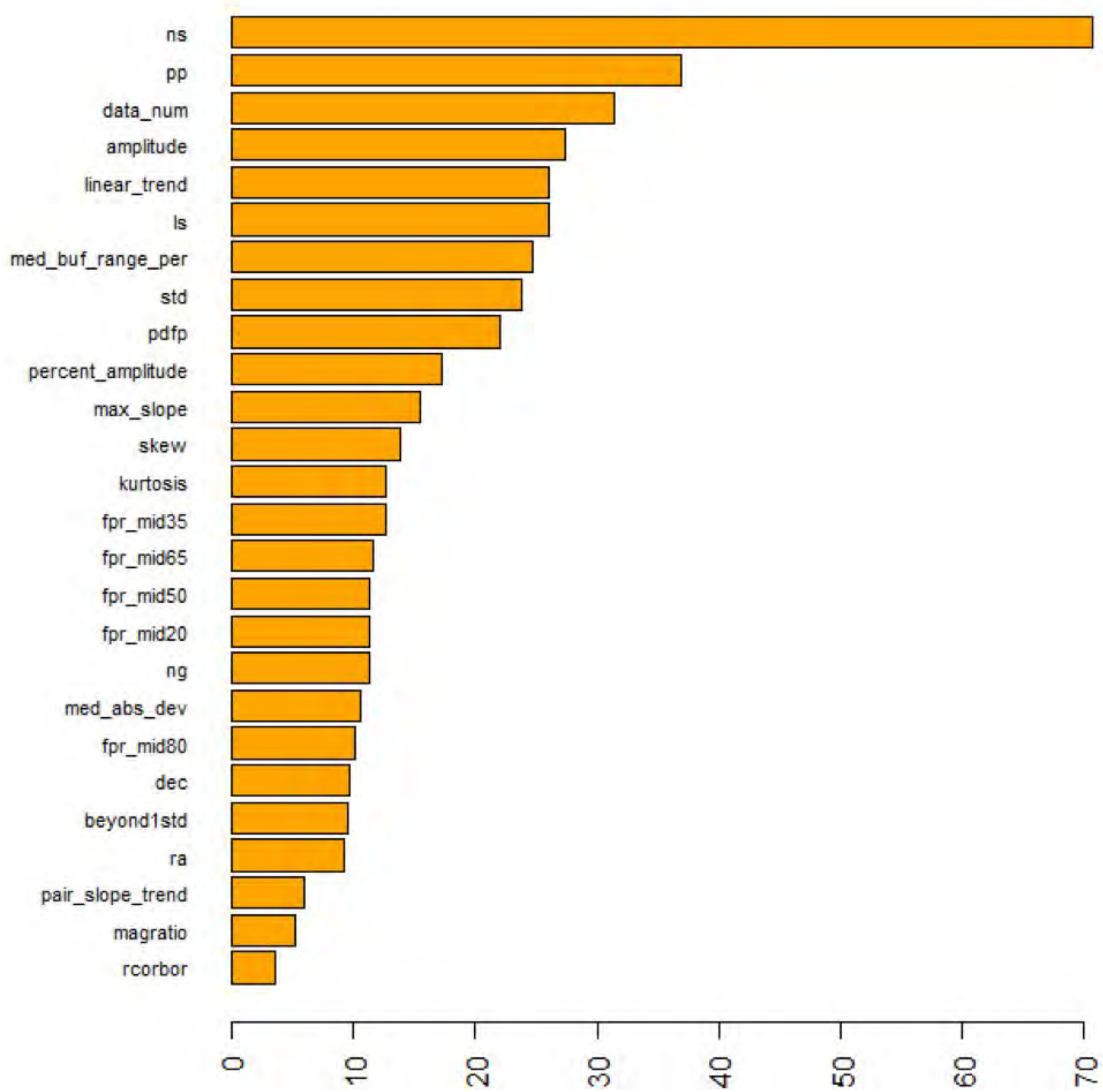
Random Forests

- Bagging, but with independent trees
- For each tree, randomly select a set of features/predictors from the full set at each split, then select the optimal predictor with corresponding optimal threshold
- train each tree with separate bootstrapping as with bagging

Hyperparameters for tuning

- number of predictors to randomly select at each split
- total number of trees
- minimum leaf node size (this keeps the tree from becoming full, and reduces computation)
- Use cross-validation to choose values
- iterate till out-of-bag error stabilizes

Variable importance



- Calculate the decrease in Gini index (or Mean Square Error, or some similar parameter) due to splits over a predictor averaged over all trees.
- Having too many predictors implies lower chance of being randomly picked. Reduce dimensionality
- Increasing number of trees does not lead to overfitting, but at very large numbers, trees can become correlated

XGBoost

Gradient boosted tree classifiers

Real-valued tree outputs that can be added together

Trees are gradually grown

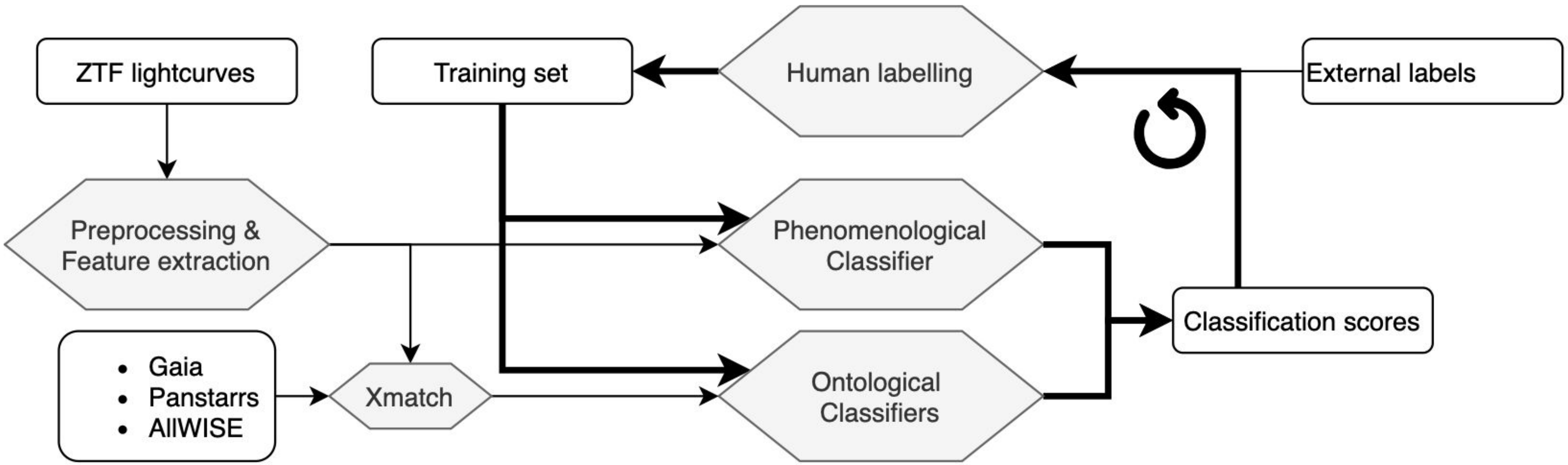
Greedy growth based on purity and loss minimization

Random subsets of data and features used per iteration

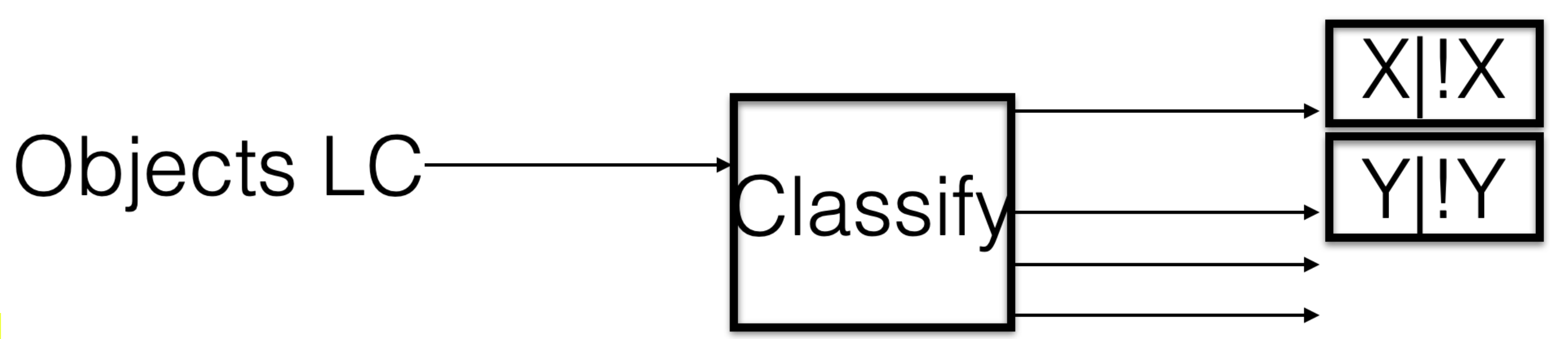
Hyperparameters

Max depth (complexity)
Min child wt (partitions)
Subsample (fraction)
Colsample (fraction)
Eta (learning rate)
Scale pos wt (balance)

SCoPe: **ZTF Variable Source Classification Project**



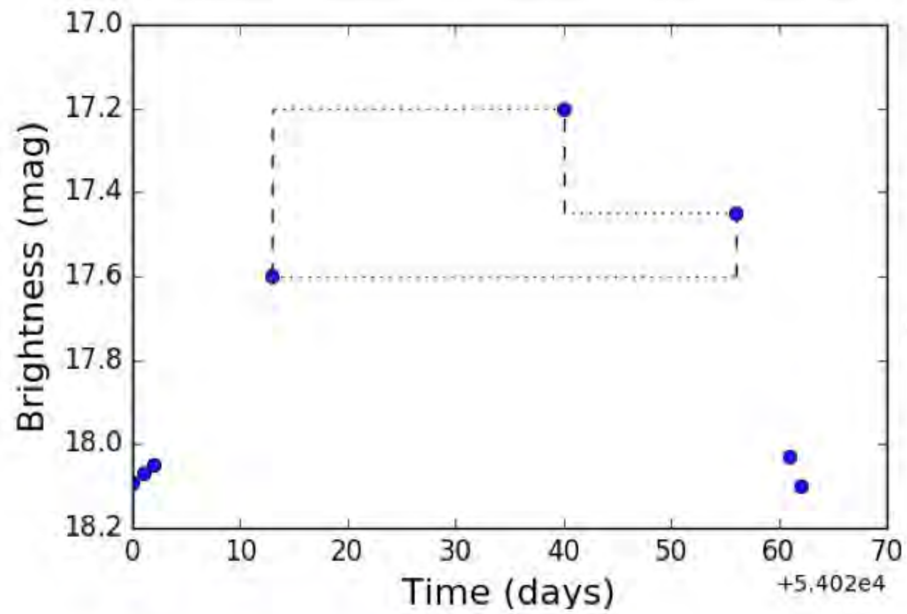
20-fields study



Van Roestel, Duev, Mahabal et al. 2020

(dmdt) Image representation

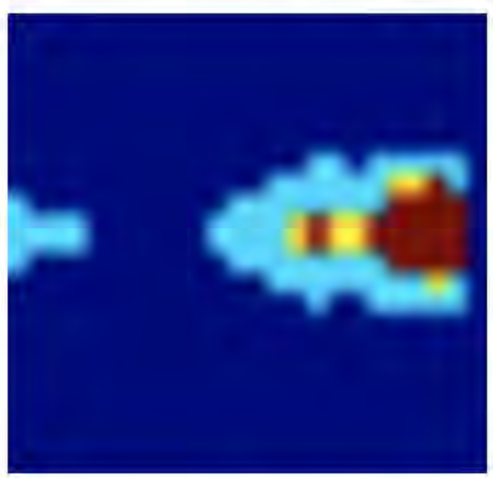
Mahabal, Sheth et al., 2017
arXiv 1709.06257

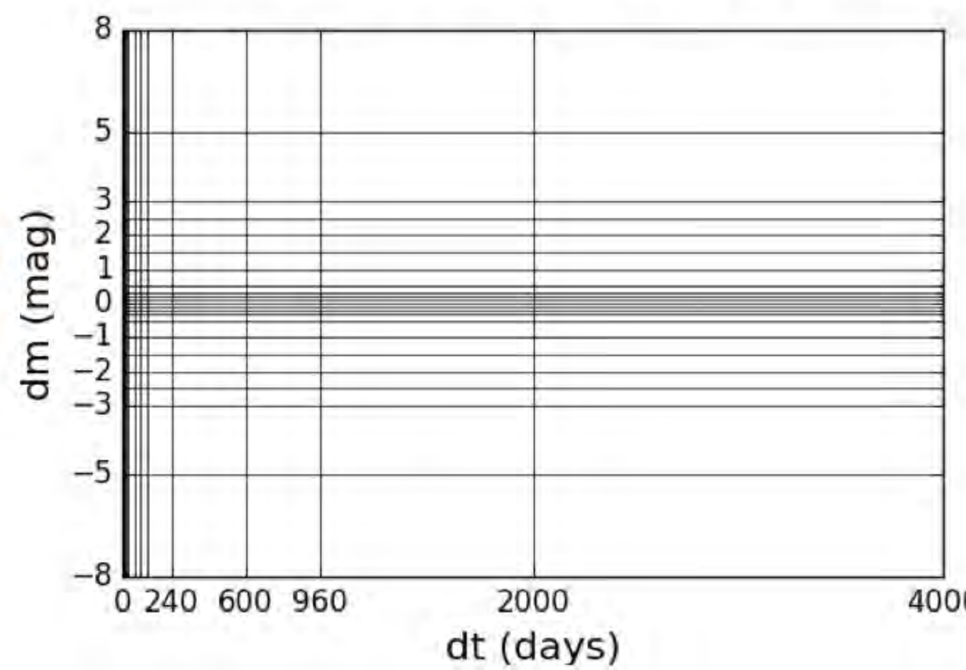


light curve with n points

23 x 24 output grid

n * (n-1)/2 points



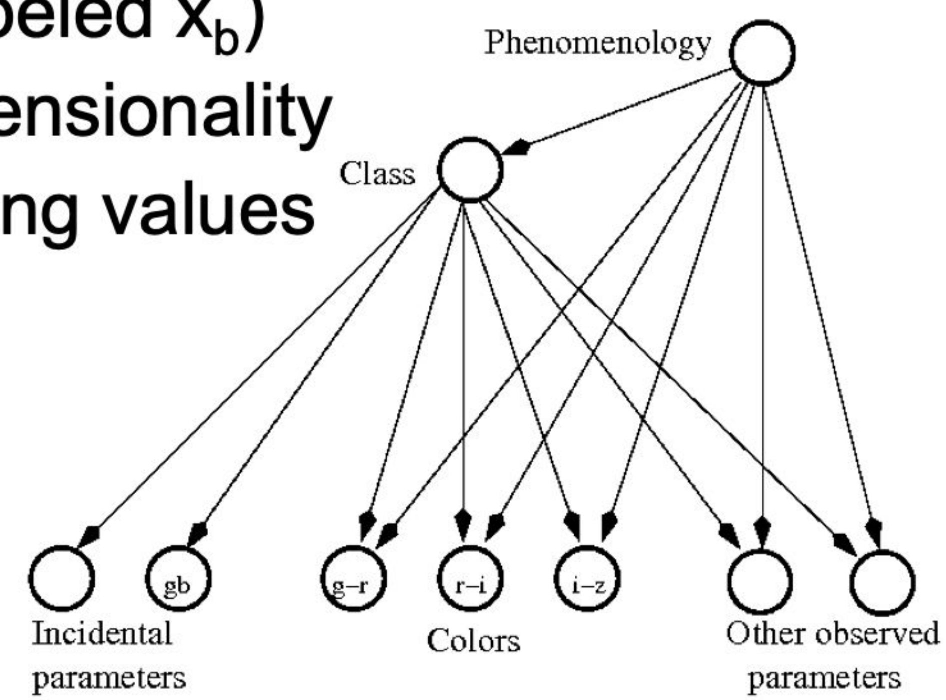


Area equalized pixels

Naïve Bayes

$$P(y=k \mid x) = P(x \mid y=k)P(k)/P(x) \propto P(k)P(x \mid y=k) \approx P(k)\prod_{b=1}^{B} P(x_b \mid y=k)$$

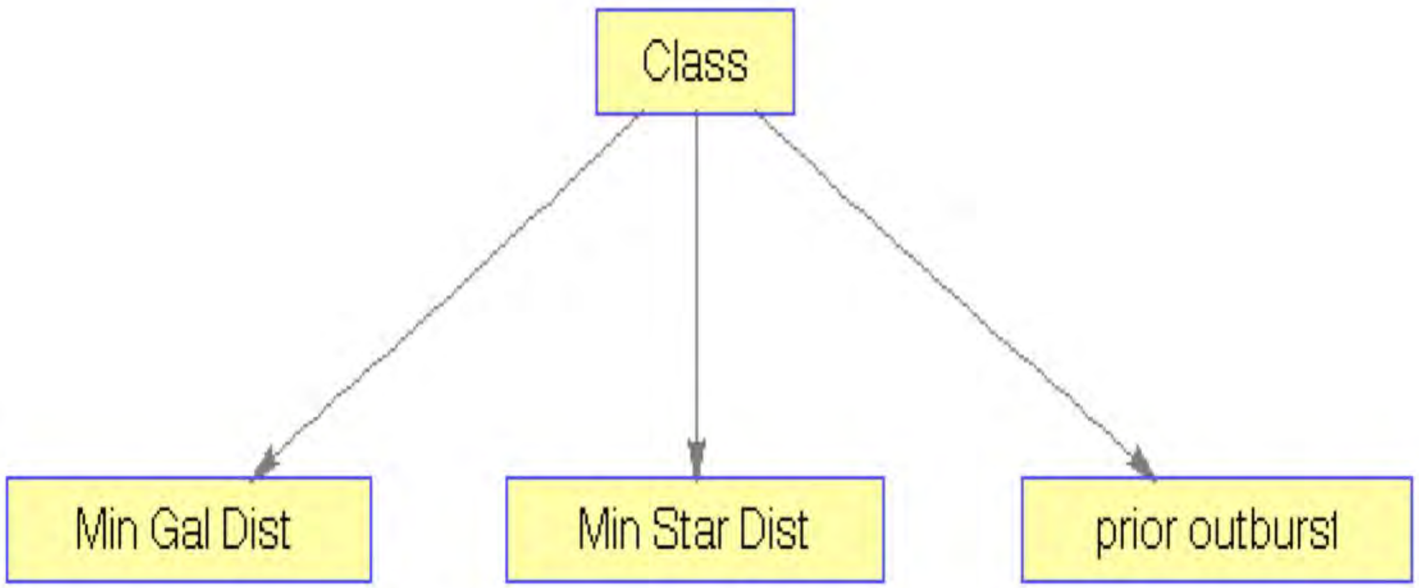
- x: feature vector of event parameters
- y: object class that gives rise to x (1<y<k)
- Certain features of x known: (position, flux)
- Others will be unknown: (color, delta-mag)
- Assumption: based on y, x is decomposable into B distinct independent classes (labeled x_b)
- This helps with the curse of dimensionality
- Also allows us to deal with missing values



BN with P60 follow-up colors:
CV/SN classification ~80% with single epoch

Radio, for instance

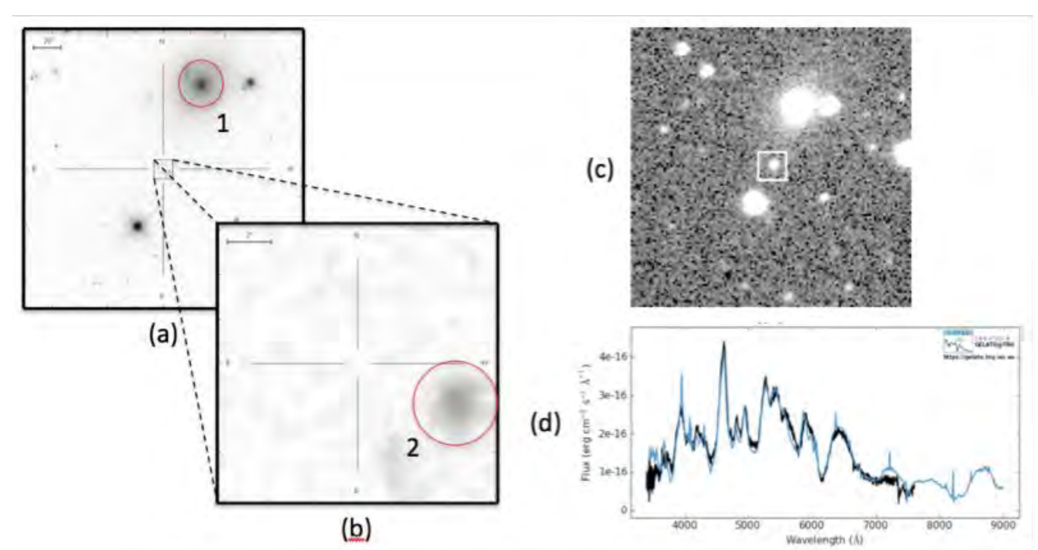
SN v. non-SN



normalized

based on peaks

$$(\frac{1}{t_{span}}(\frac{1}{N}\Sigma_i w_i(p_i - p_m)^2)^{1/2}$$



Practical problems

- ZTF
- Problem definition
- available metadata (irrelevance, missing values, …)
- label contamination