Decision Trees

Lecturer:
Jiří Matas

Authors:
Jiří Matas, Ondřej Drbohlav

Centre for Machine Perception
Czech Technical University, Prague
http://cmp.felk.cvut.cz

9.1.2017

Pros and Cons of Decision Trees

Pros

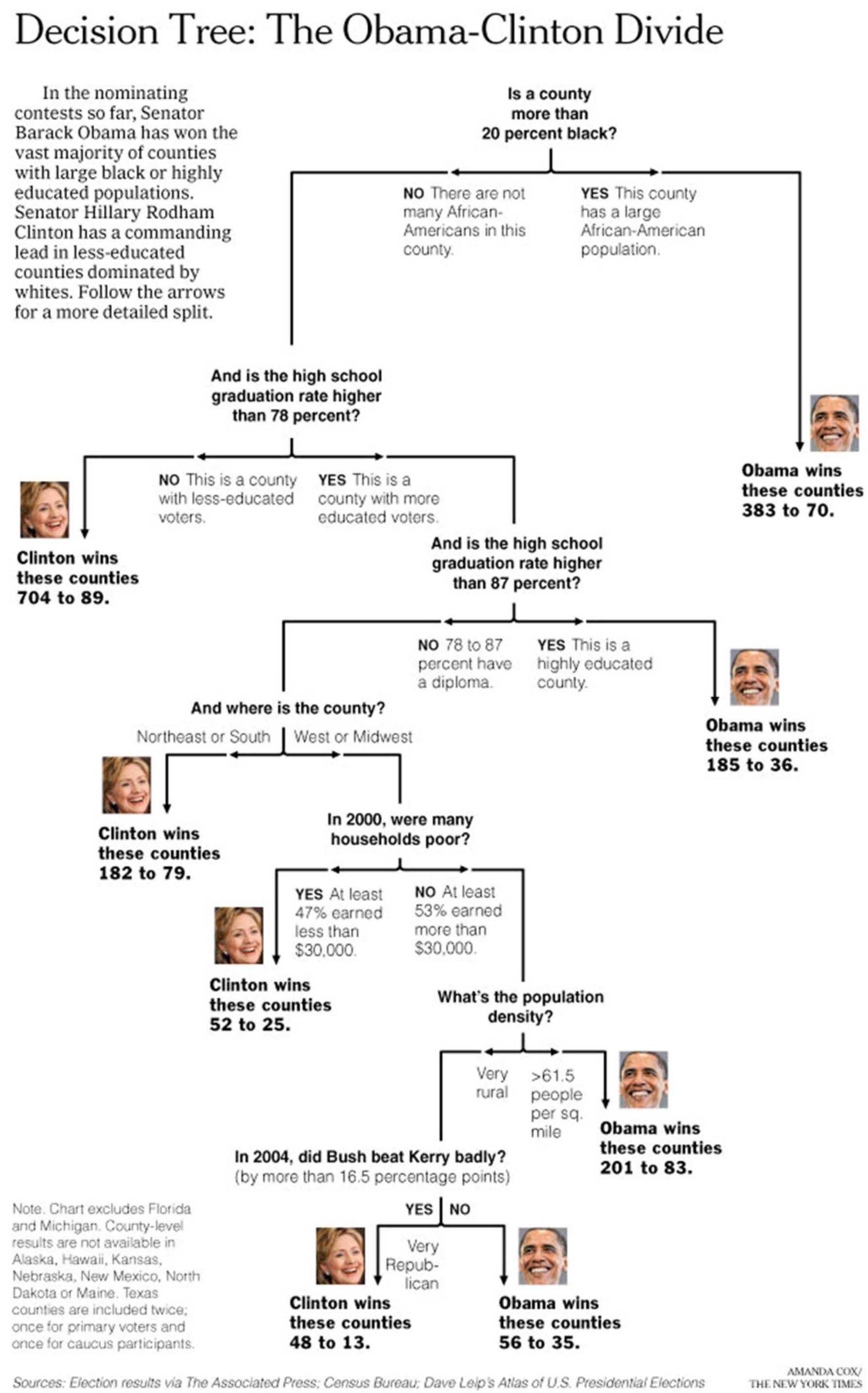
1. fast (in decision time)
2. sublinear in the number of classes K
3. intuitive, interpretable
4. no restrictions on the feature space:

$$X_1 \times X_2 \times \dots X_D \,, \qquad (D \gg 1) \tag{1}$$

   X_i can be discrete, continuous, categorical, ordinal, . . .
5. to classify, not all X_i may be actually needed
6. sequential decisions
7. high degree of robustness to outliers (unlike e.g. linear classifiers)
8. different scales between features are not a problem
9. deals with additional information
10. classified samples can be kept at tree nodes/leaves; this enables to compute class probabilities (therefore decision error), and use statistics of the samples for further analysis
11. can be used in regression problems

Cons

1. not well justified learning algorithm
2. overfitting is a problem for standard methods, but this can be fixed (randomized d. trees, decision *forests*)

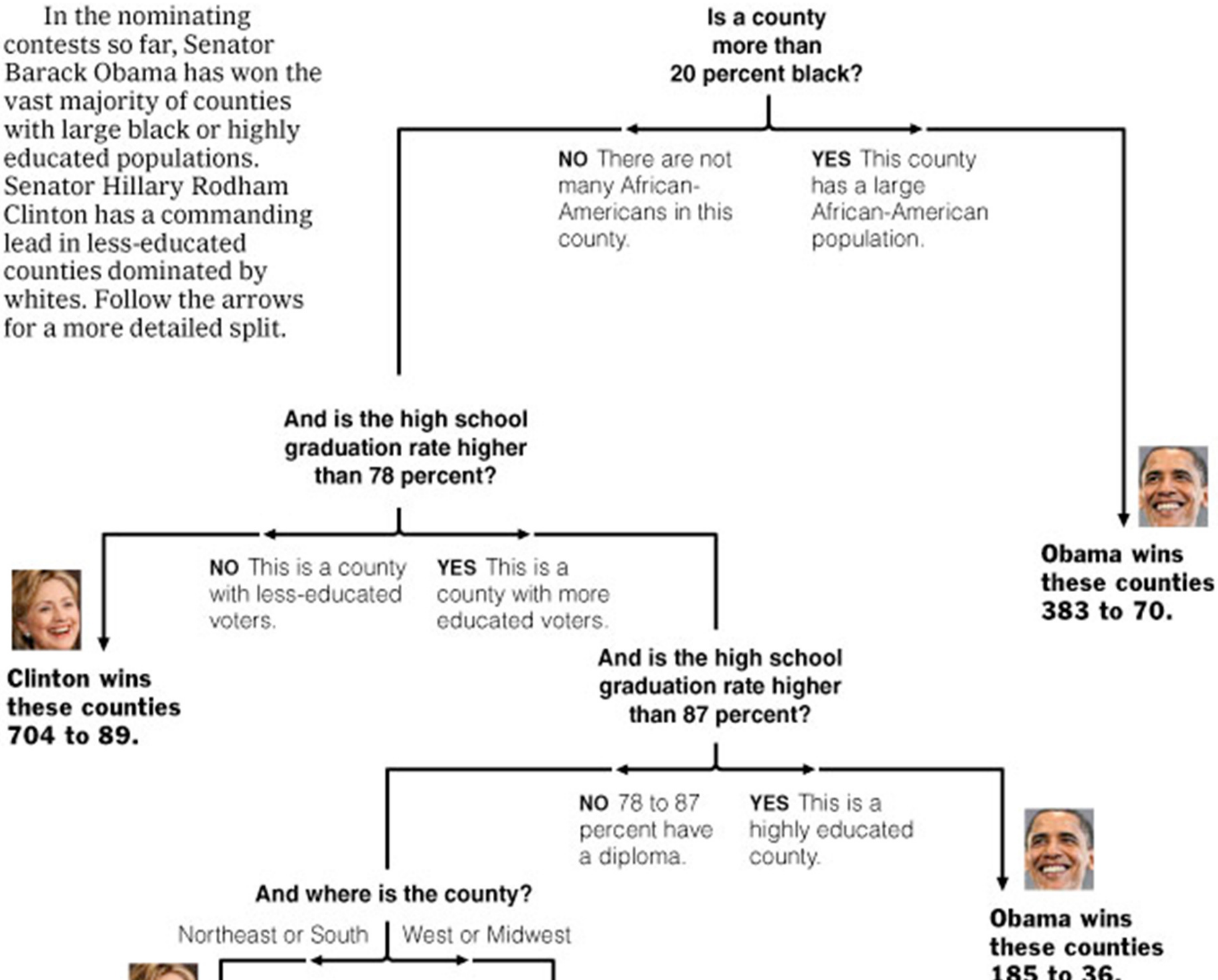


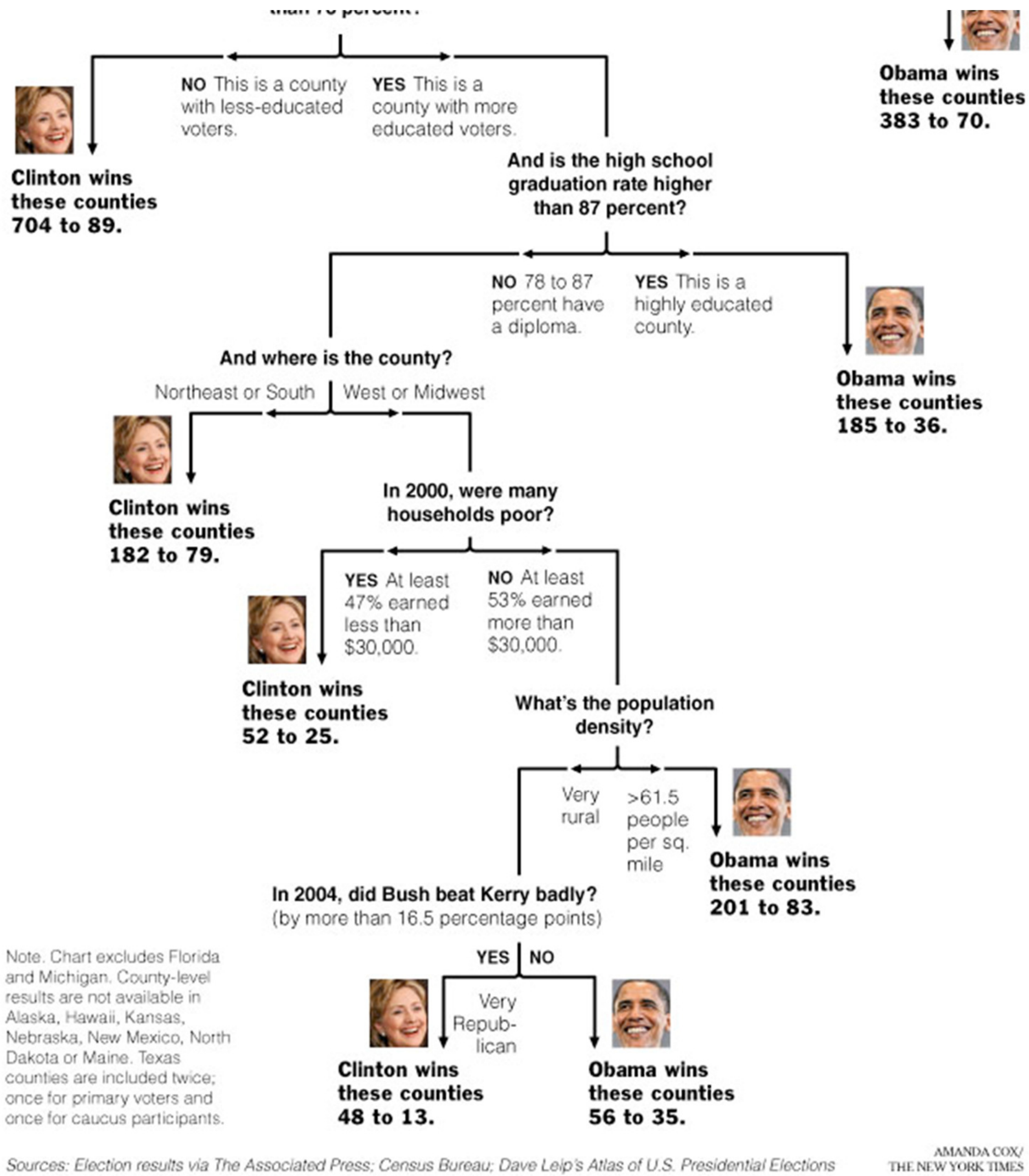
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

Decision Tree: The Obama-Clinton Divide

In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites. Follow the arrows for a more detailed split.

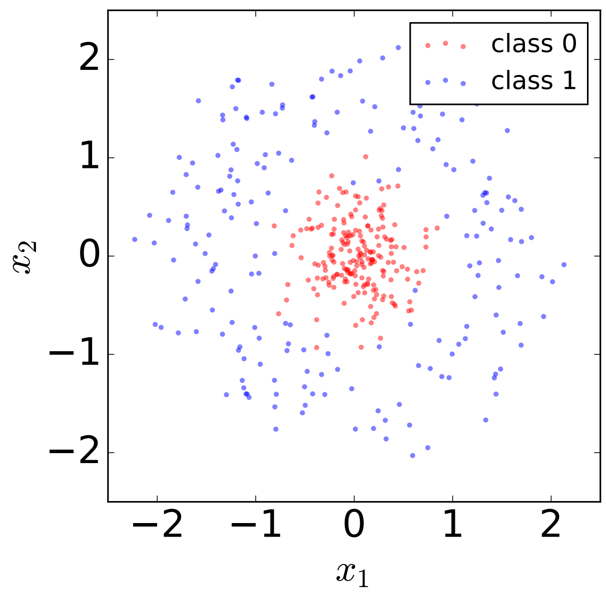


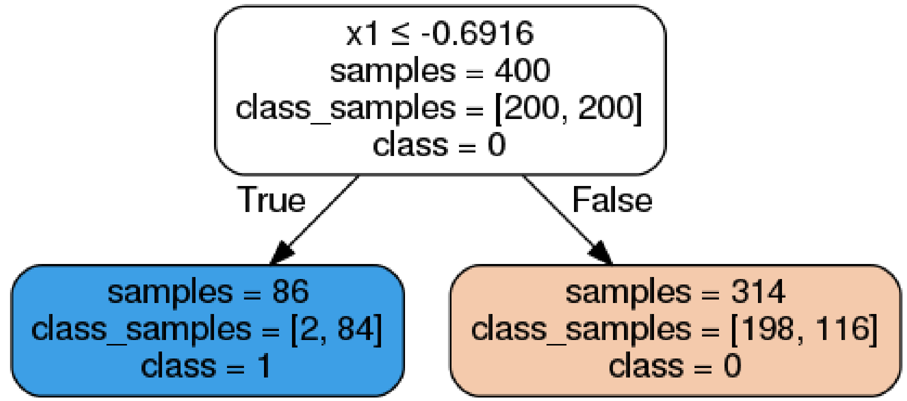


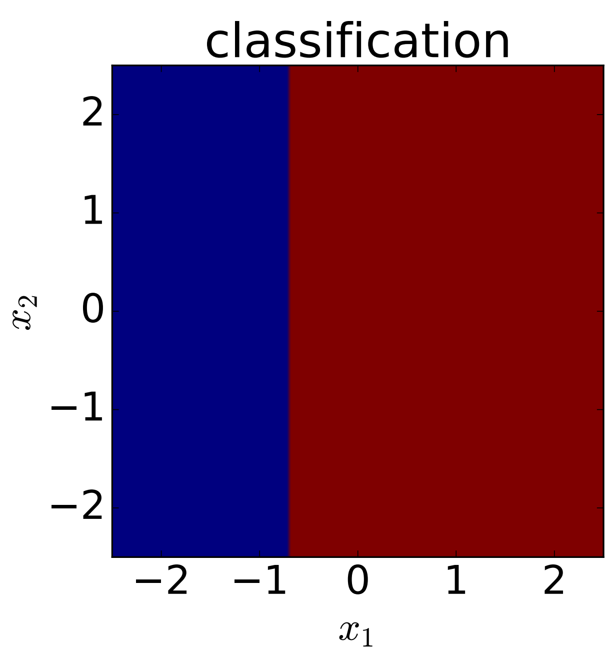
NO This is a county with less-educated voters.
YES This is a county with more educated voters.
Clinton wins these counties 704 to 89.
Obama wins these counties 383 to 70.
And is the high school graduation rate higher than 87 percent?
NO 78 to 87 percent have a diploma.
YES This is a highly educated county.
Obama wins these counties 185 to 36.
And where is the county?
Northeast or South
West or Midwest
Clinton wins these counties 182 to 79.
In 2000, were many households poor?
YES At least 47% earned less than $30,000.
NO At least 53% earned more than $30,000.
Clinton wins these counties 52 to 25.
What's the population density?
Very rural
>61.5 people per sq. mile
Obama wins these counties 201 to 83.
In 2004, did Bush beat Kerry badly? (by more than 16.5 percentage points)
YES
NO
Very Repub-lican
Clinton wins these counties 48 to 13.
Obama wins these counties 56 to 35.
Note. Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Texas counties are included twice; once for primary voters and once for caucus participants.
Sources: Election results via The Associated Press; Census Bureau; Dave Leip's Atlas of U.S. Presidential Elections
AMANDA COX/ THE NEW YORK TIMES

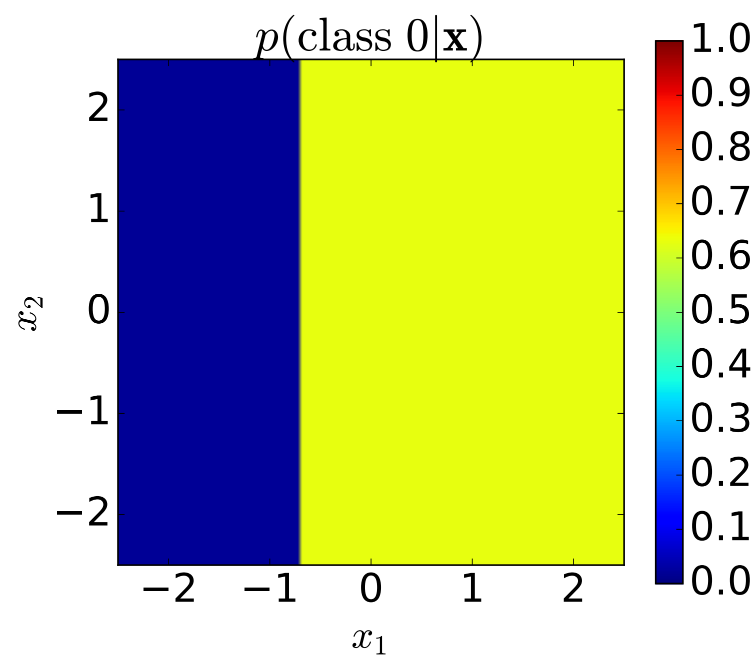
3

Decision tree: Example, 2D data

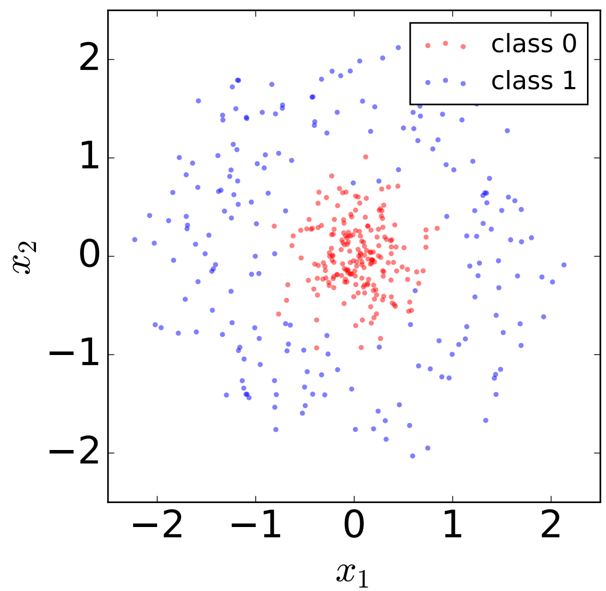


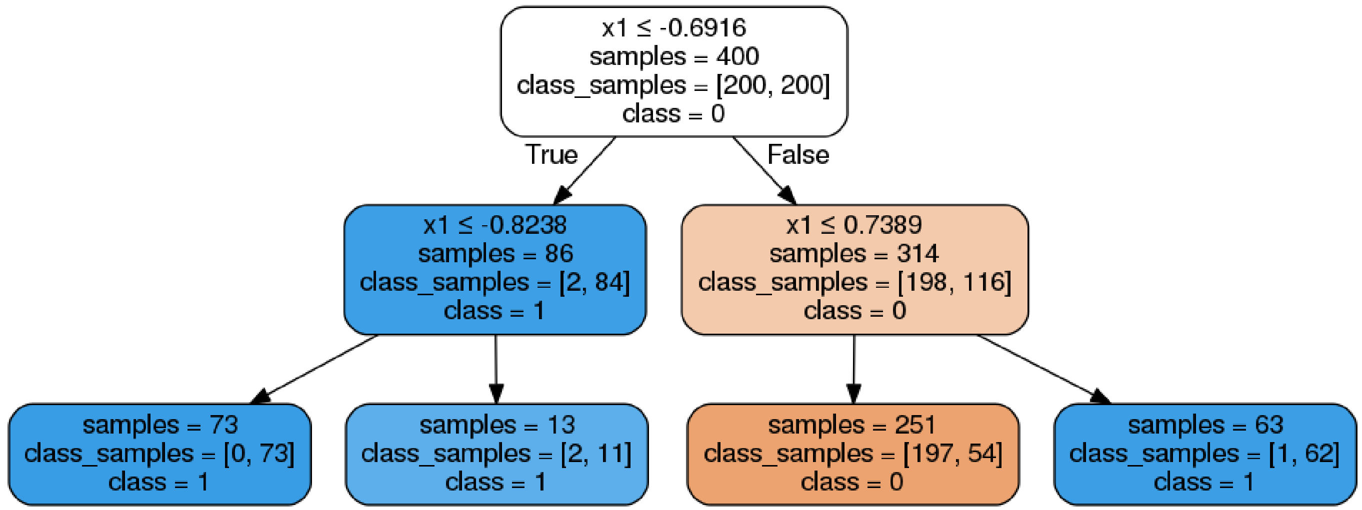


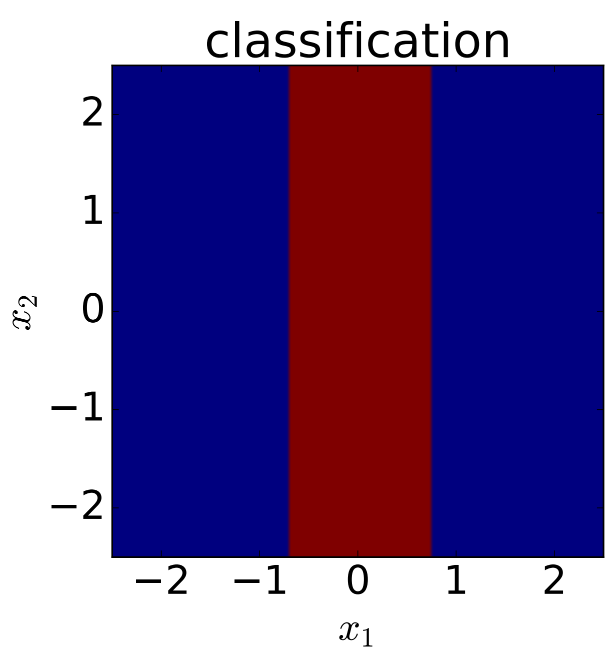


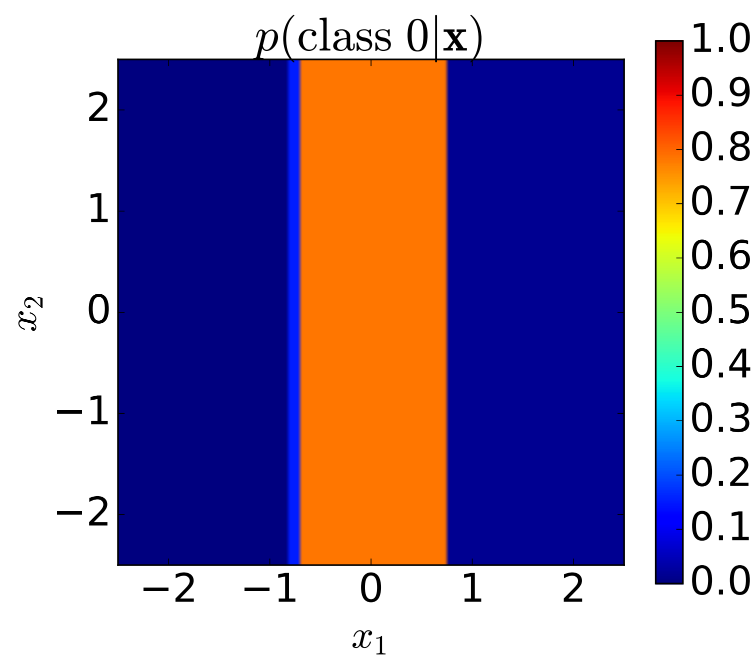


Decision tree: Example, 2D data

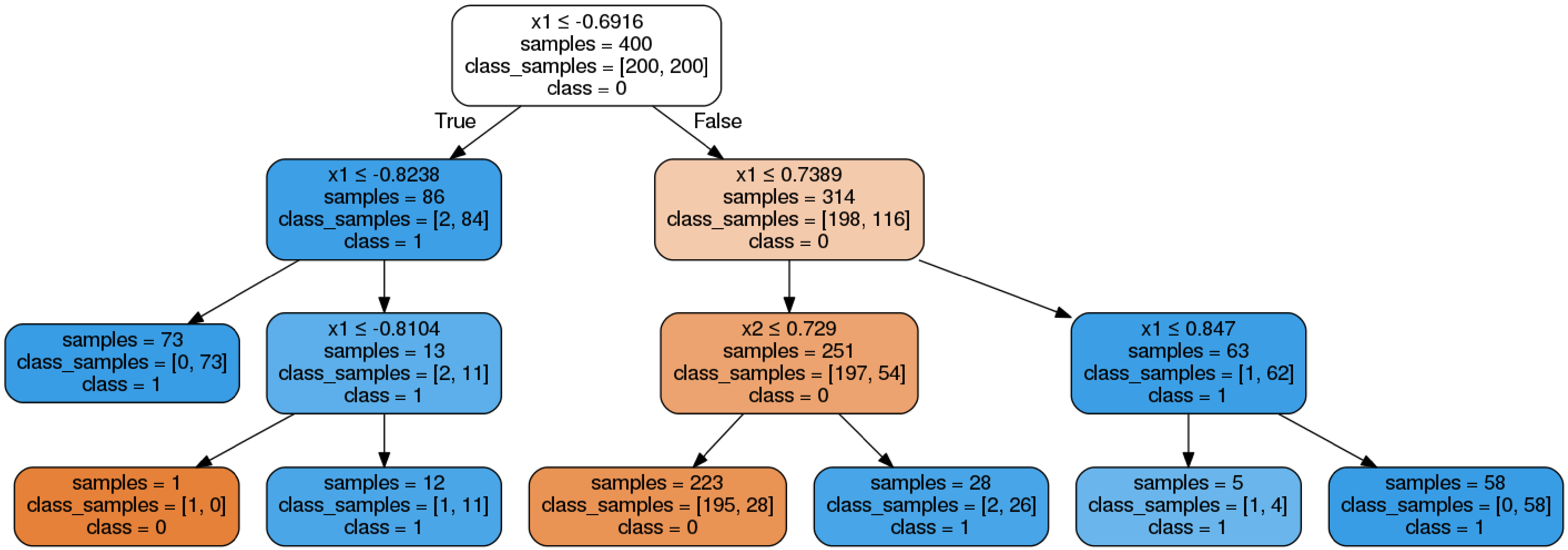


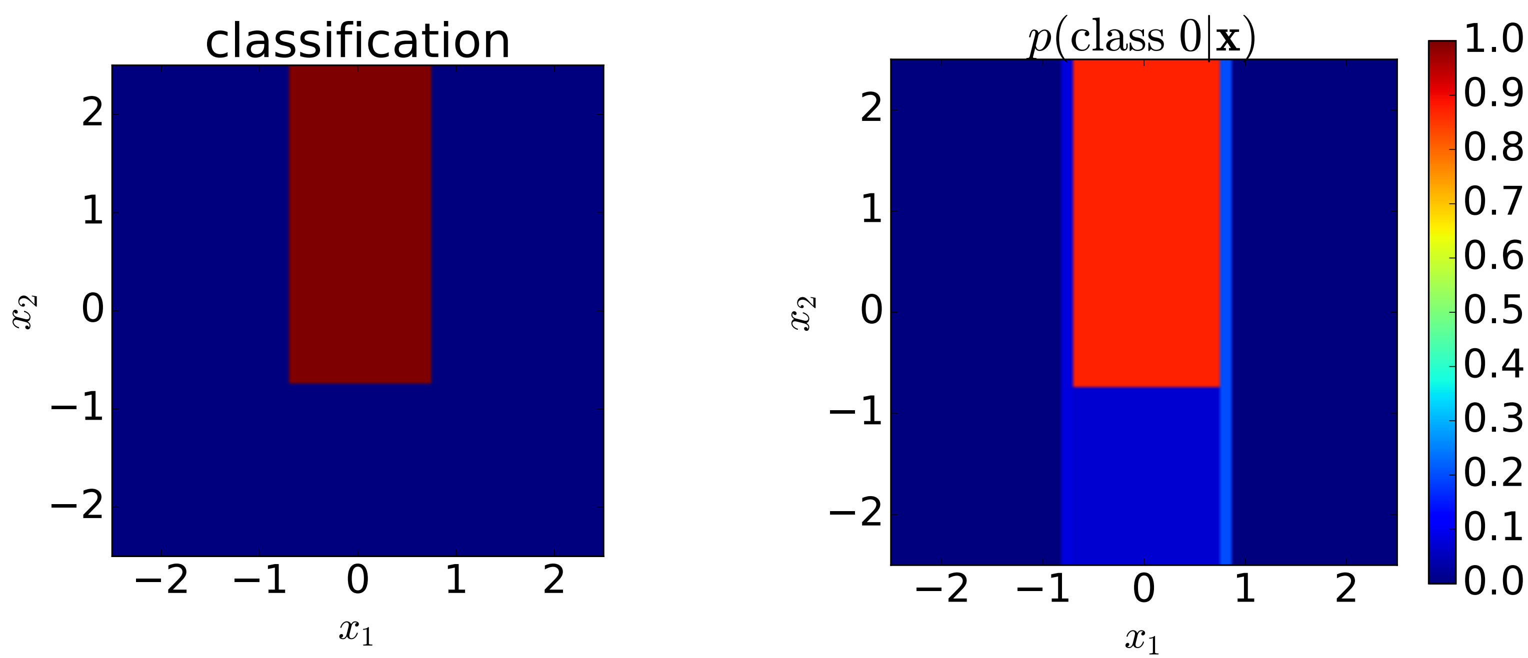


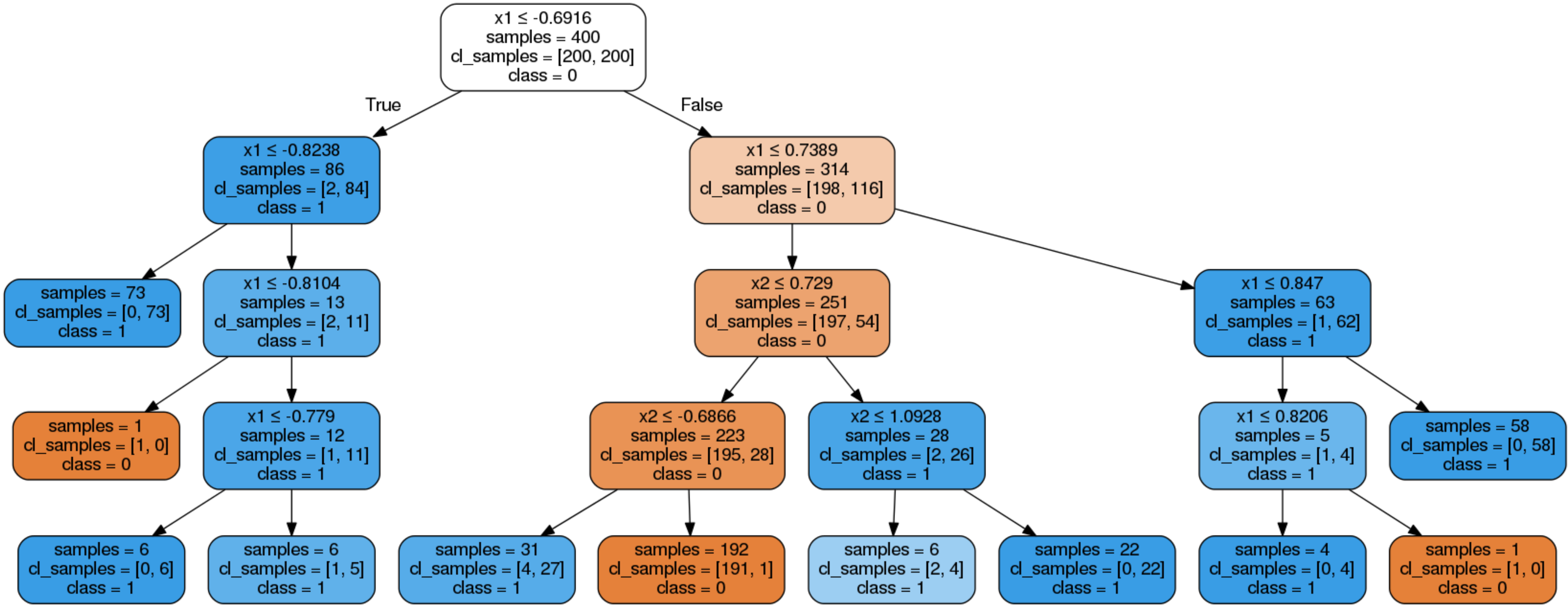


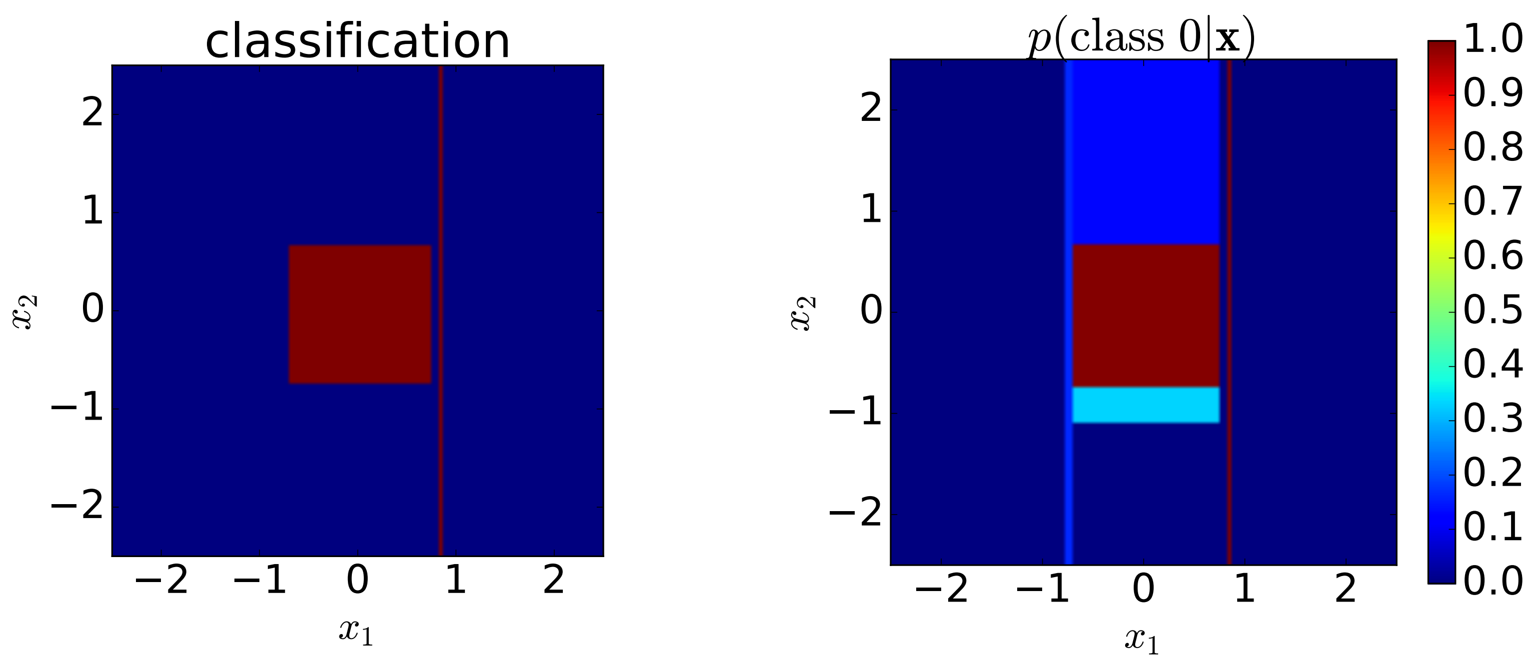


Decision tree: Example, 2D data






x1 ≤ -0.6916
samples = 400
cl_samples = [200, 200]
class = 0
True
False
x1 ≤ -0.8238
samples = 86
cl_samples = [2, 84]
class = 1
x1 ≤ 0.7389
samples = 314
cl_samples = [198, 116]
class = 0
samples = 73
cl_samples = [0, 73]
class = 1
x1 ≤ -0.8104
samples = 13
cl_samples = [2, 11]
class = 1
x2 ≤ 0.729
samples = 251
cl_samples = [197, 54]
class = 0
x1 ≤ 0.847
samples = 63
cl_samples = [1, 62]
class = 1
samples = 1
cl_samples = [1, 0]
class = 0
x1 ≤ -0.779
samples = 12
cl_samples = [1, 11]
class = 1
x2 ≤ -0.6866
samples = 223
cl_samples = [195, 28]
class = 0
x2 ≤ 1.0928
samples = 28
cl_samples = [2, 26]
class = 1
x1 ≤ 0.8206
samples = 5
cl_samples = [1, 4]
class = 1
samples = 58
cl_samples = [0, 58]
class = 1
samples = 6
cl_samples = [0, 6]
class = 1
samples = 6
cl_samples = [1, 5]
class = 1
samples = 31
cl_samples = [4, 27]
class = 1
samples = 192
cl_samples = [191, 1]
class = 0
samples = 6
cl_samples = [2, 4]
class = 1
samples = 22
cl_samples = [0, 22]
class = 1
samples = 4
cl_samples = [0, 4]
class = 1
samples = 1
cl_samples = [1, 0]
class = 0


classification
$p(\text{class } 0|\mathbf{x})$
x_2
x_1
2
1
0
−1
−2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Learning decision trees: An example

Problem: decide whether to wait for a table at a restaurant.

Attributes used

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range ($, $$, $$$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

Goal predicate: WillWait?

Attribute-based representations

Examples described by attribute values (Boolean, discrete, continuous)

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

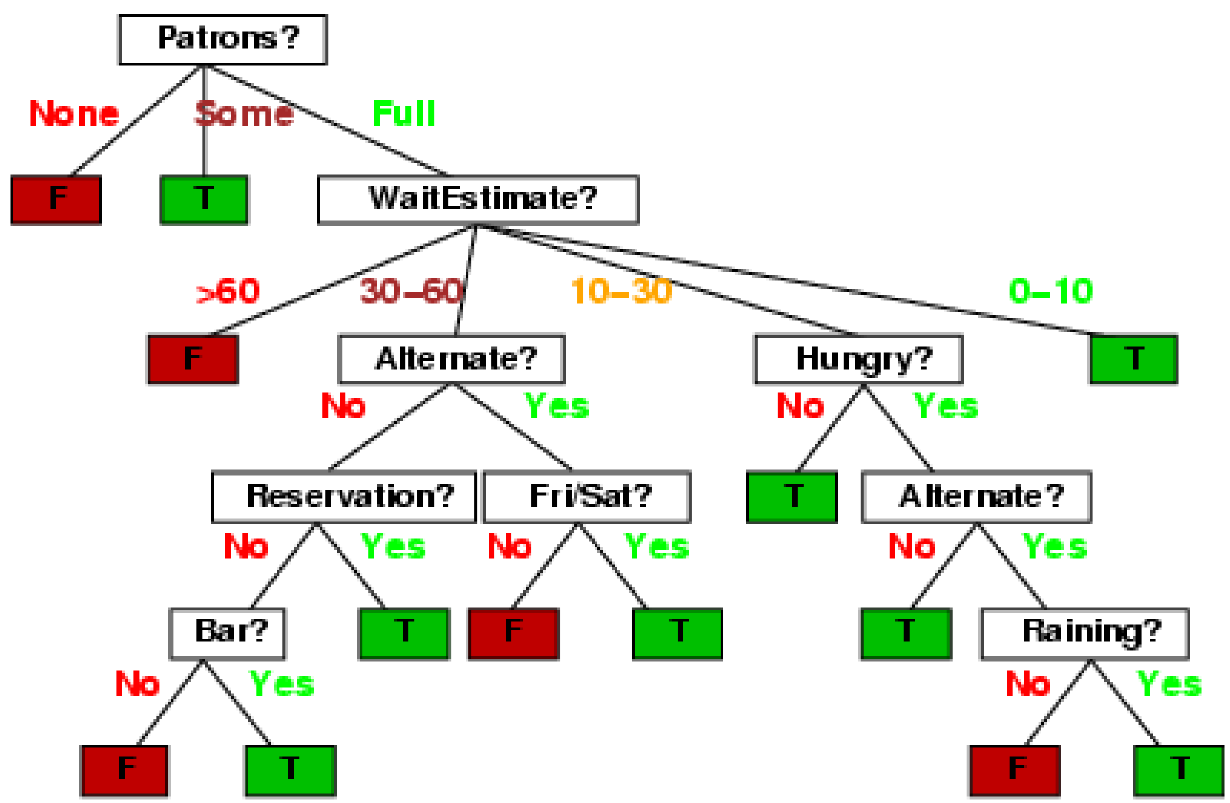
12 examples
(6 T, 6 F)

Classification of examples is positive (T) or negative (F)

Decision trees

One possible representation for hypotheses

E.g., here is a tree for deciding whether to wait:



Top-Down Induction of DT (TDIDT)

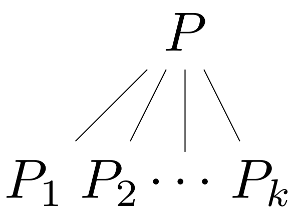
Growth Phase: The tree is constructed top-down.

- Find the “best” attribute.
- Partition examples based on the attribute’s values.
- Apply recursively to each partition.

Pruning Phase: The tree is pruned bottom-up

- For each node, keep subtree or change to leaf.
- Choose by comparing estimated error.

Splitting a Node to Branches



- Define a splitting function $S(P, \boldsymbol{\theta})$ as a mapping which takes a set P (data points at a node) and splitting parameters $\boldsymbol{\theta}$ and produces the partition of P to $\{P_1, P_2, \ldots, P_k\}$. That is,

$$S : P, \boldsymbol{\theta} \rightarrow \{P_1, P_2, \ldots, P_k\}\,, \tag{2}$$

$$P_i \cap P_j = \emptyset \Leftrightarrow i \neq j \tag{3}$$

$$\cup_{i=1}^{k} P_i = P \tag{4}$$

- $\boldsymbol{\theta}$: may contain the branching factor k, the index i of the splitting dimension X_i
- $k = 2$: binary split
- $k > 2$: multiway split
- When X_i is continuous, $\boldsymbol{\theta}$ contains the range(s) of values for individual partitions (when $k = 2$, typically a threshold. Example: [height $\leqslant 1.75m$?])

How to Split: Greedy Maximization of Information Gain

Information Gain (IG) measures the expected reduction in entropy when a set P is partitioned to subsets $P_1, P_2, ..., P_k$. The information gain $IG(A)$ (A is the attribute used for splitting P) is

$$IG(A) = H(P) - \sum_{i=1}^{k} \frac{|P_i|}{|P|} H(P_i) \tag{5}$$

$$H(P) = -\sum_{j} \frac{|P^{(j)}|}{|P|} \log \frac{|P^{(j)}|}{|P|} \quad \text{(entropy of } P\text{)} \tag{6}$$

($P^{(j)}$: number of points of class j in P)

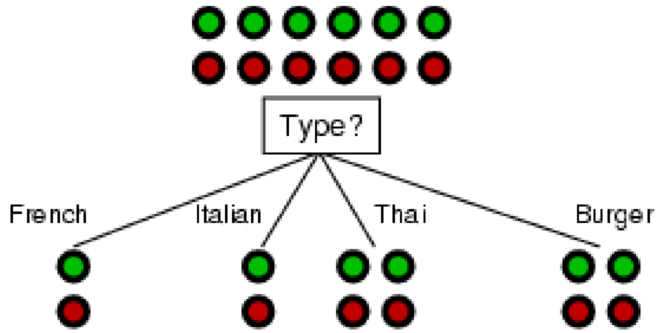
$$H(P_i) = -\sum_{j} \frac{|P_i^{(j)}|}{|P_i|} \log \frac{|P_i^{(j)}|}{|P_i|} \quad \text{(entropy of } P_i\text{)} \tag{7}$$

($P_i^{(j)}$: number of points of class j in P_i)

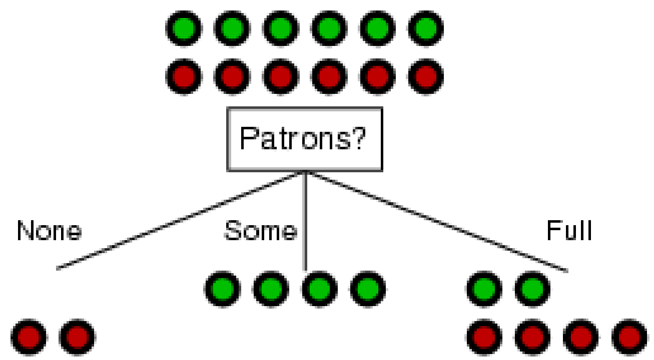
Information gain

For the training set, $p = n = 6$, *entropy* $I(6/12, 6/12) = 1$ bit

Consider the attributes *Type* and *Patrons*:



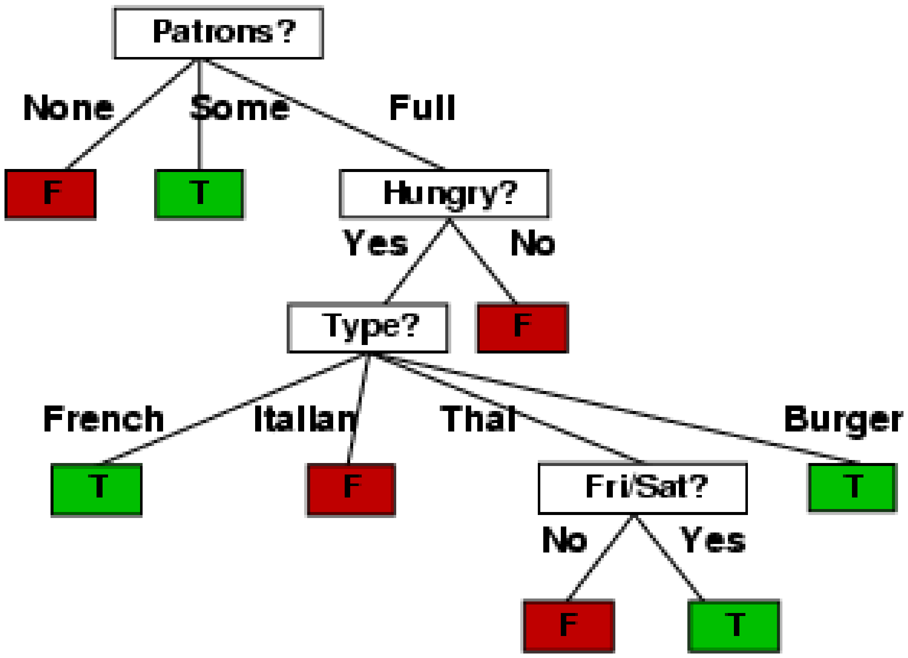
$$IG(Type) = 1 - [\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$



$$IG(Patrons) = 1 - [\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I(\frac{2}{6}, \frac{4}{6})] = .0541 \text{ bits}$$

Example contd.

Decision tree learned from the 12 examples:



How to Split: Greedy Maximization of Information Gain (reformulation)

Input:

- training set $\mathcal{T} = \{(\mathbf{x}_1, k_1), (\mathbf{x}_2, k_2), ..., (\mathbf{x}_N, k_N)\}$.
- class $\mathcal{S}$ of allowed splitting parameters

Do:

1. $P = \mathcal{T}$
2. Find the best splitting function (best is measured by **Information Gain** which is the difference of entropies H of data before and after the split. *Note*: To maximize IG means to minimize entropies after the split.)

$$S = \underset{\boldsymbol{\theta} \in \mathcal{S}}{\text{argmin}} \left(\frac{|P_1|}{|P|} H(P_1) + \frac{|P_2|}{|P|} H(P_2) + \ldots \frac{|P_k|}{|P|} H(P_k) \right) \tag{8}$$

$$\{P_1, P_2, \ldots, P_k\} = S(P, \boldsymbol{\theta}) \tag{9}$$

$$H(P_i) = -\sum_j \frac{|P_i^{(j)}|}{|P_i|} \log \frac{|P_i^{(j)}|}{|P_i|} \quad \text{(entropy of } P_i\text{)} \tag{10}$$

($P_i^{(j)}$: number of points of class j in P_i)

3. Go recursively to branches

How to Split: Greedy Minimization of Training Error

Input:

- training set $\mathcal{T} = \{(\mathbf{x}_1, k_1), (\mathbf{x}_2, k_2), ..., (\mathbf{x}_N, k_N)\}$.
- class $\mathcal{S}$ of allowed splitting parameters

Do:

1. $P = \mathcal{T}$
2. Find the best splitting function (best measured by training error ϵ):

$$S = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{S}} \left(\frac{|P_1|}{|P|} \epsilon(P_1) + \frac{|P_2|}{|P|} \epsilon(P_2) + \dots \frac{|P_k|}{|P|} \epsilon(P_k) \right) \tag{11}$$

$$\{P_1, P_2, \dots, P_k\} = S(P, \boldsymbol{\theta}) \tag{12}$$

$$\epsilon(P_i) = 1 - \max_j \frac{|P_i^{(j)}|}{|P_i|} \tag{13}$$

($P_i^{(j)}$: number of points of class j in P_i)

3. Go recursively to branches

Random Decision Forests

Decision trees tend to overfit. Combining multiple DTs offers a solution to this problem.

But:

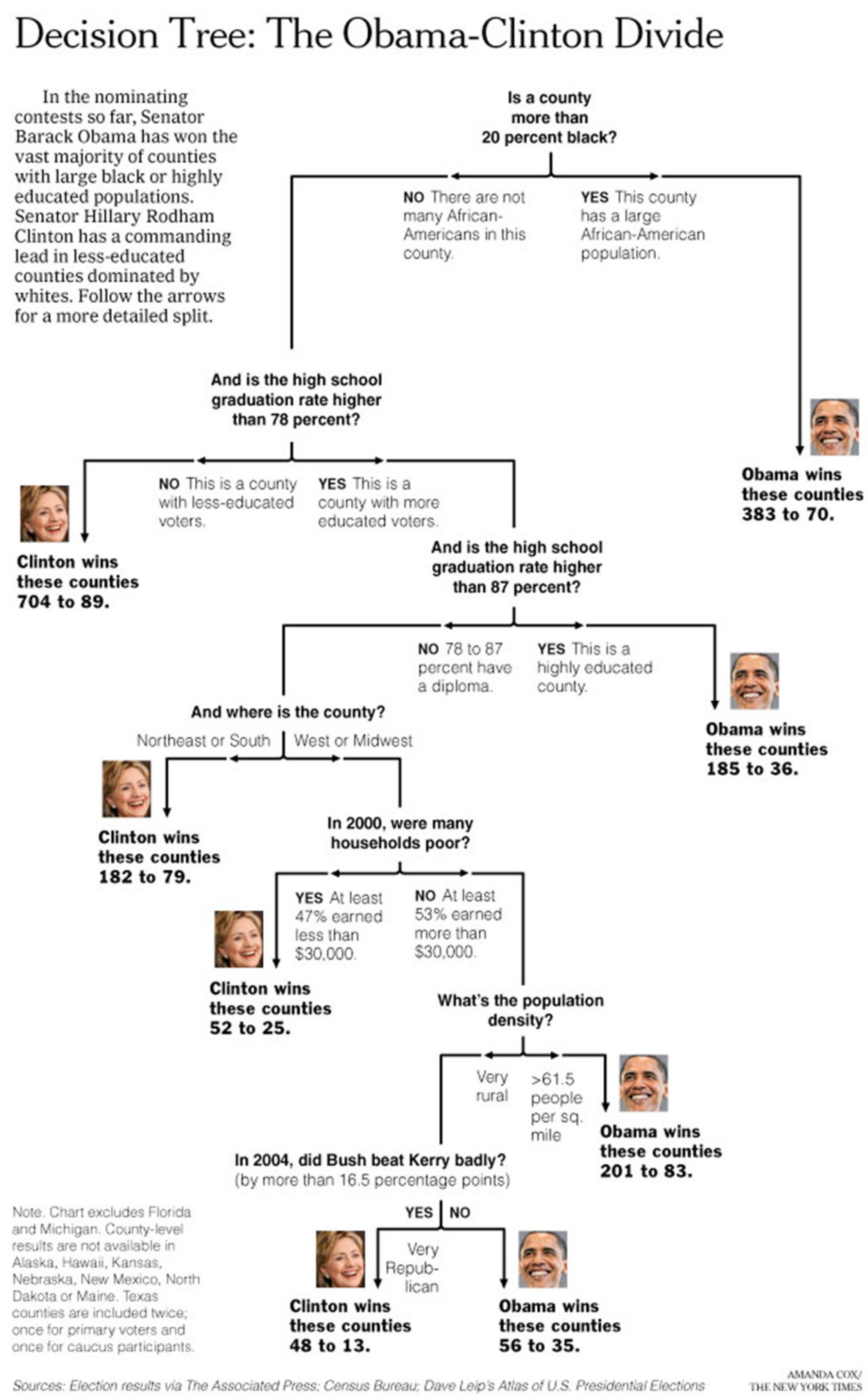
- How should multiple trees be combined?
- How should multiple trees be constructed? Note that so far, the procedure to obtain a DT has been deterministic. Repeating this procedure N times will result in N identical trees.

Combining trees: Often, majority of votes for a given class is used (like in NNs).

Constructing multiple different trees requires randomization. Randomization can be employed for:

- Training set generation for individual DTs, using **bagging**. For a training set with N_t samples, this means selecting N_t samples from this training set by sampling *with replacement*. Some training data will be selected multiple times. Different training sets will be produced due to random nature of sampling.
- **Random subset of features**. Bagging cannot produce uncorrelated trees if some features strongly dominate in the DT construction. Selecting randomly a subset of features to consider during the construction of individual DTs helps.

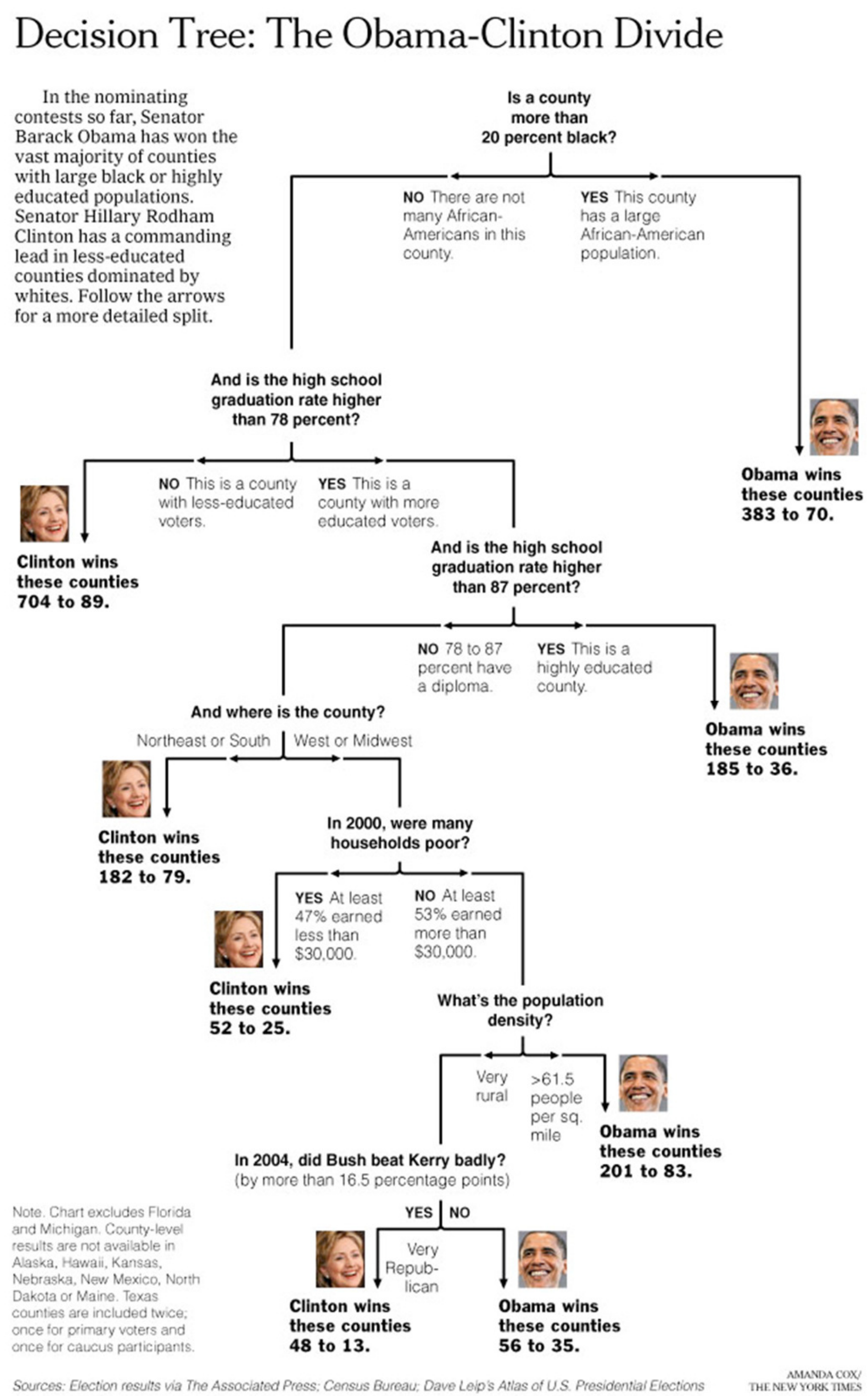
m p



New York Times
April 16, 2008

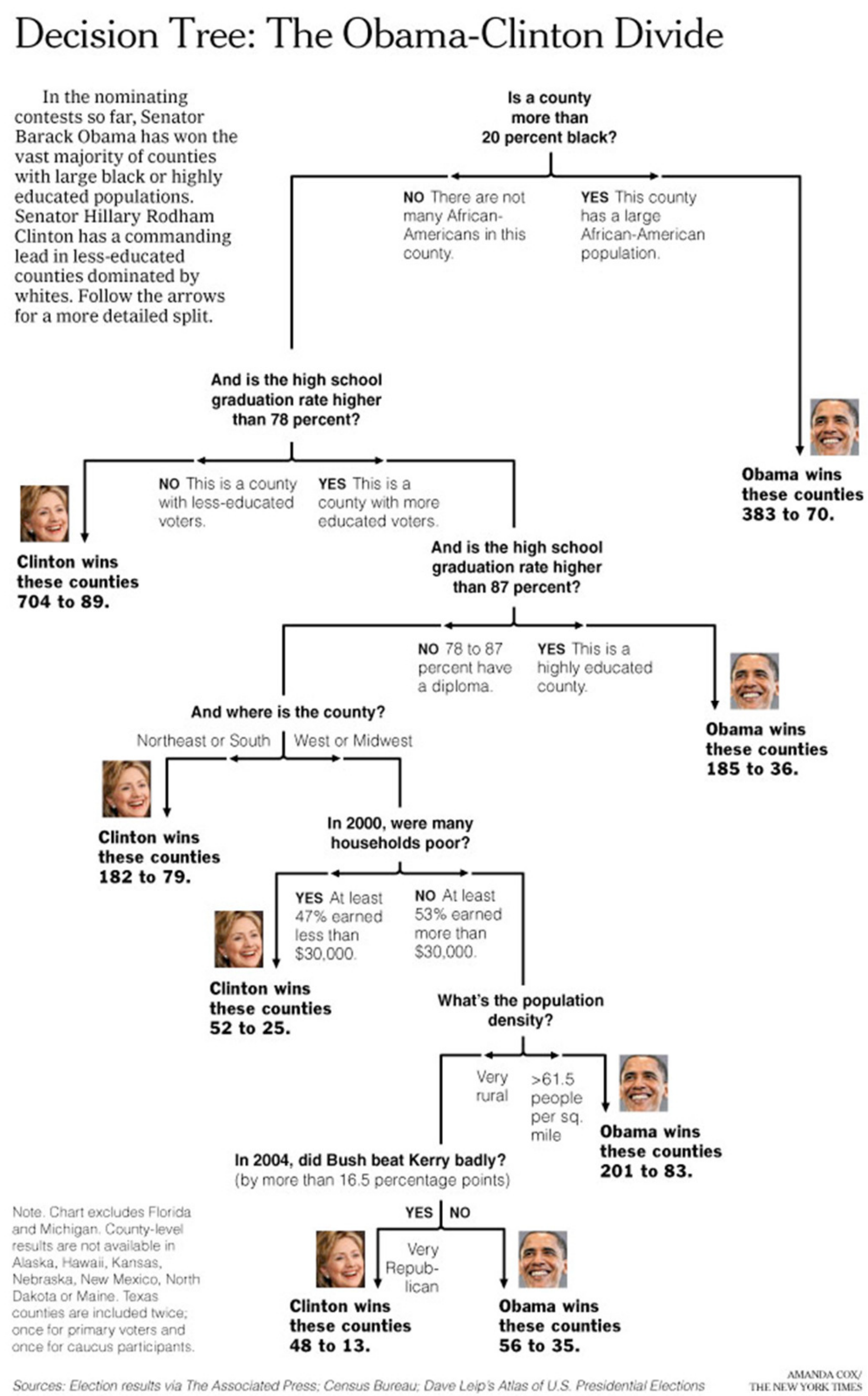
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008



Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

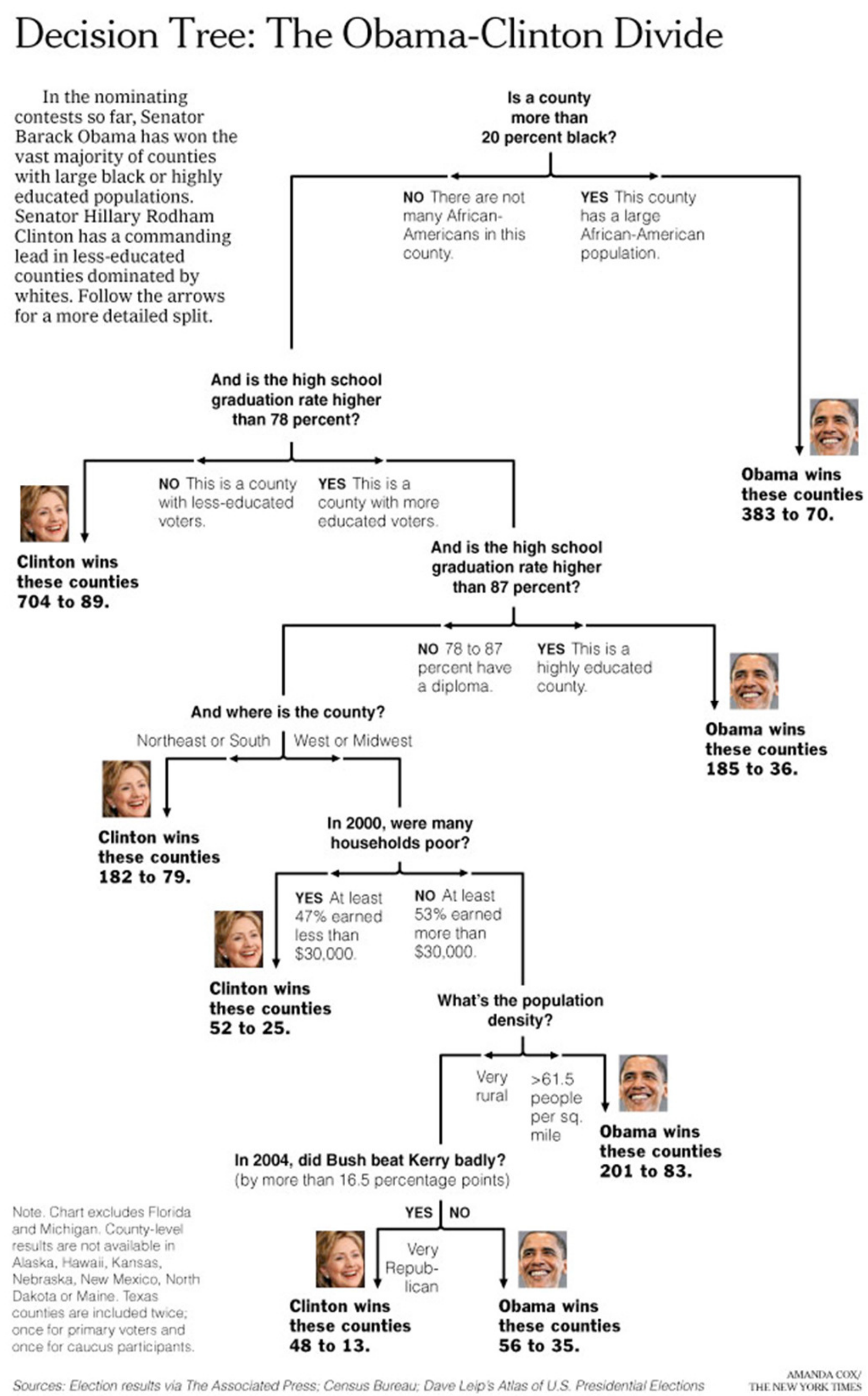
New York Times
April 16, 2008



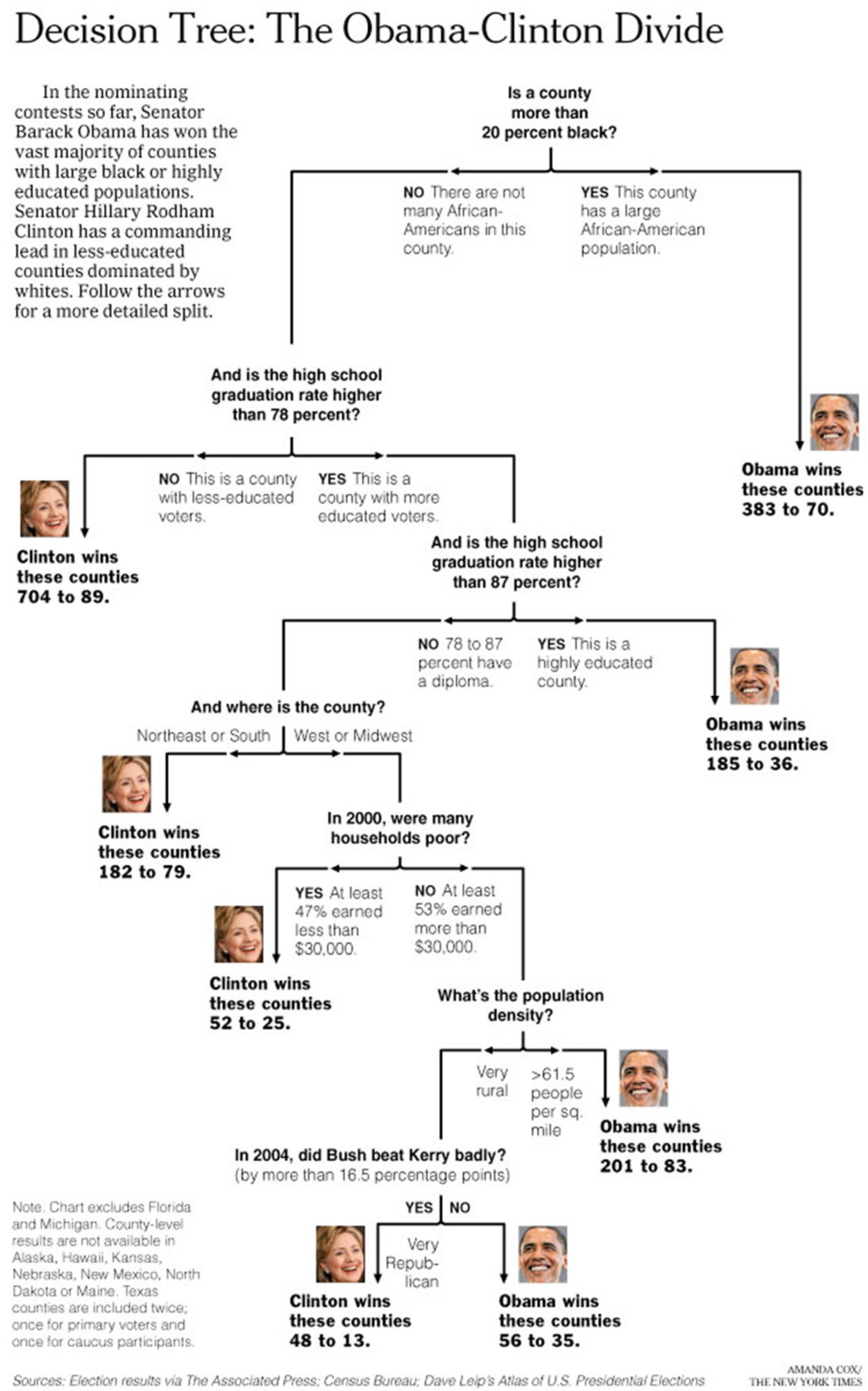
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

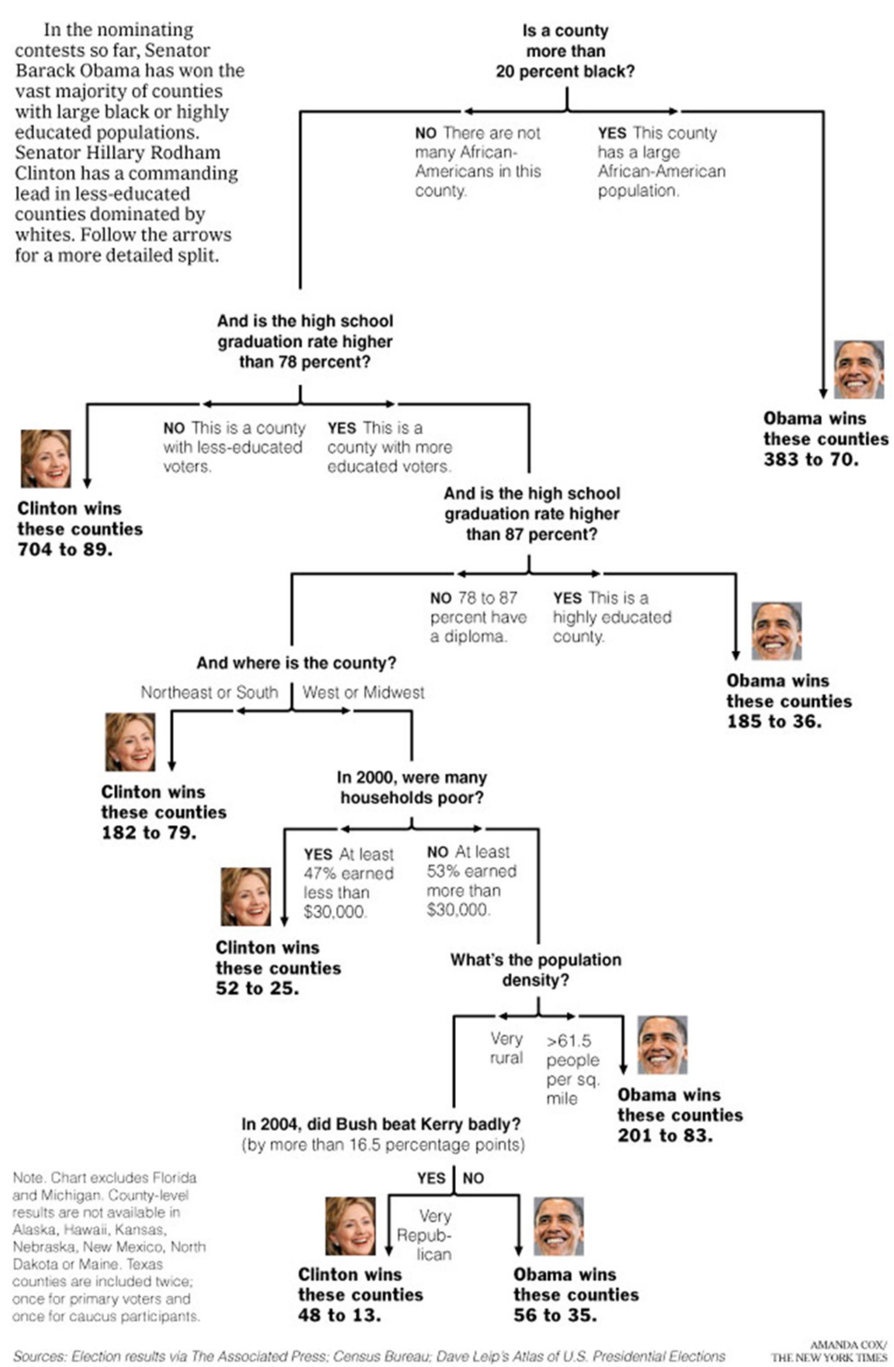


New York Times
April 16, 2008



Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

Decision Tree: The Obama-Clinton Divide

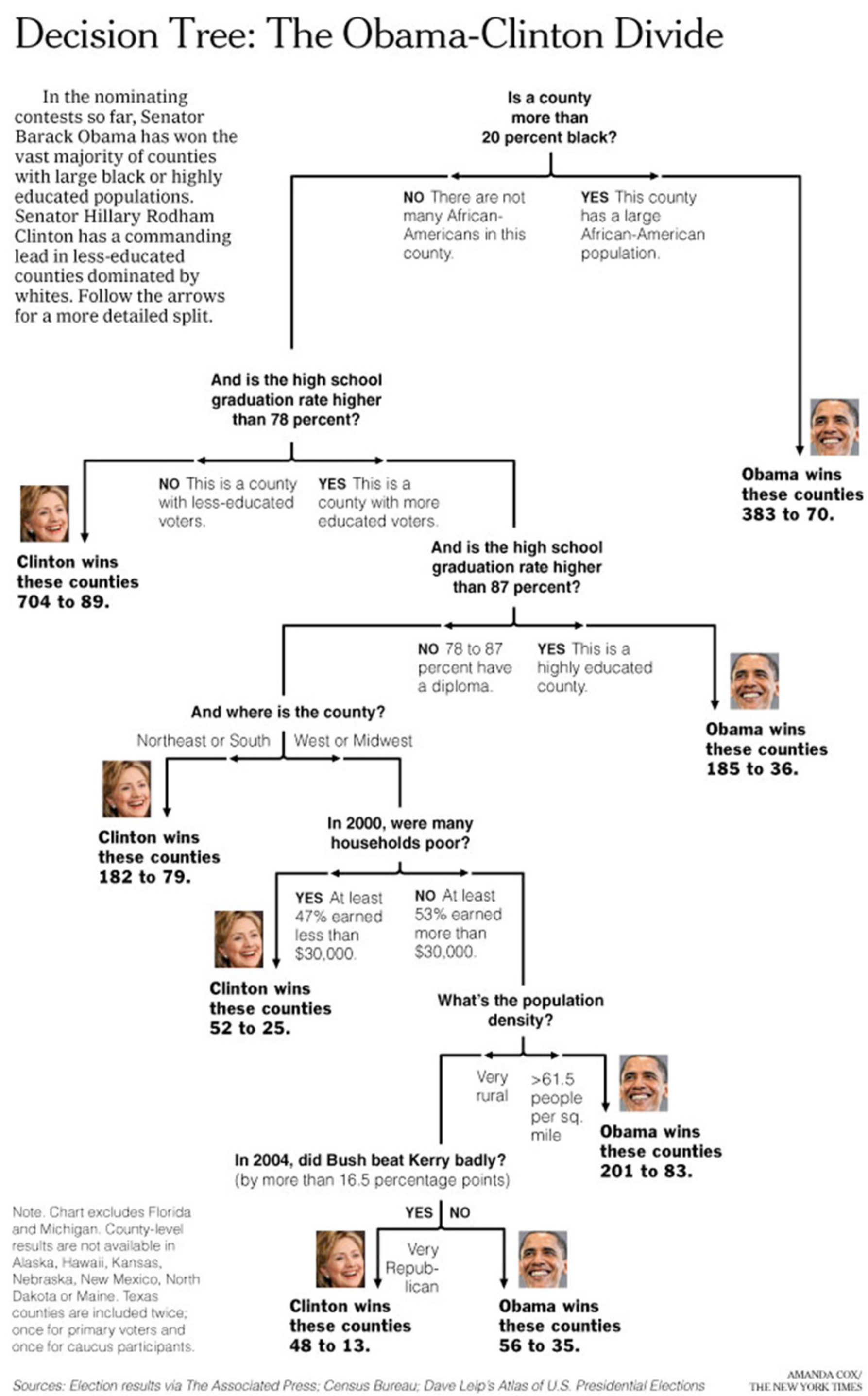


New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

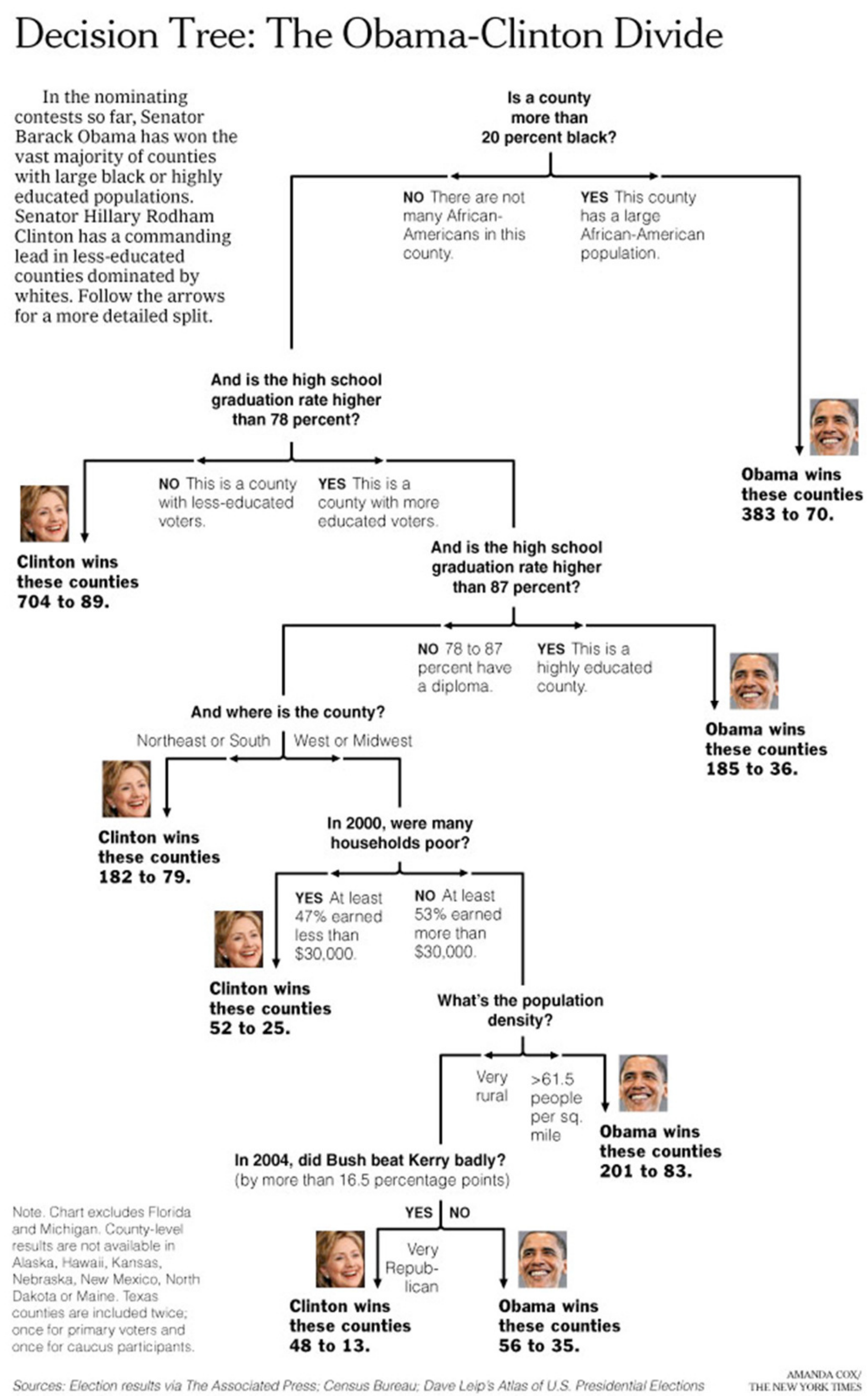
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural
for humans.
Style of many "How to"
manuals and trouble-shooting
procedures.



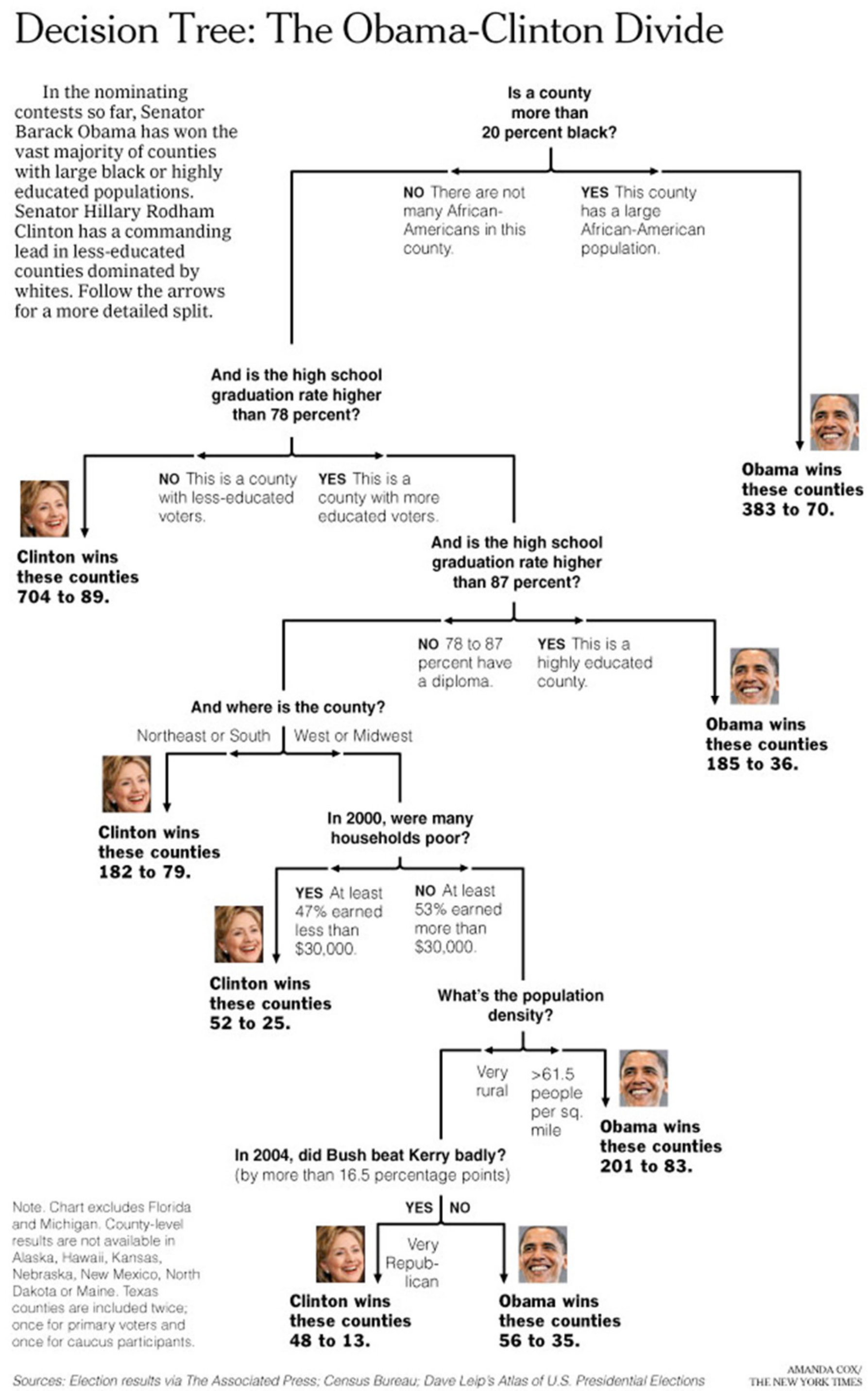
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural
for humans.
Style of many "How to"
manuals and trouble-shooting
procedures.



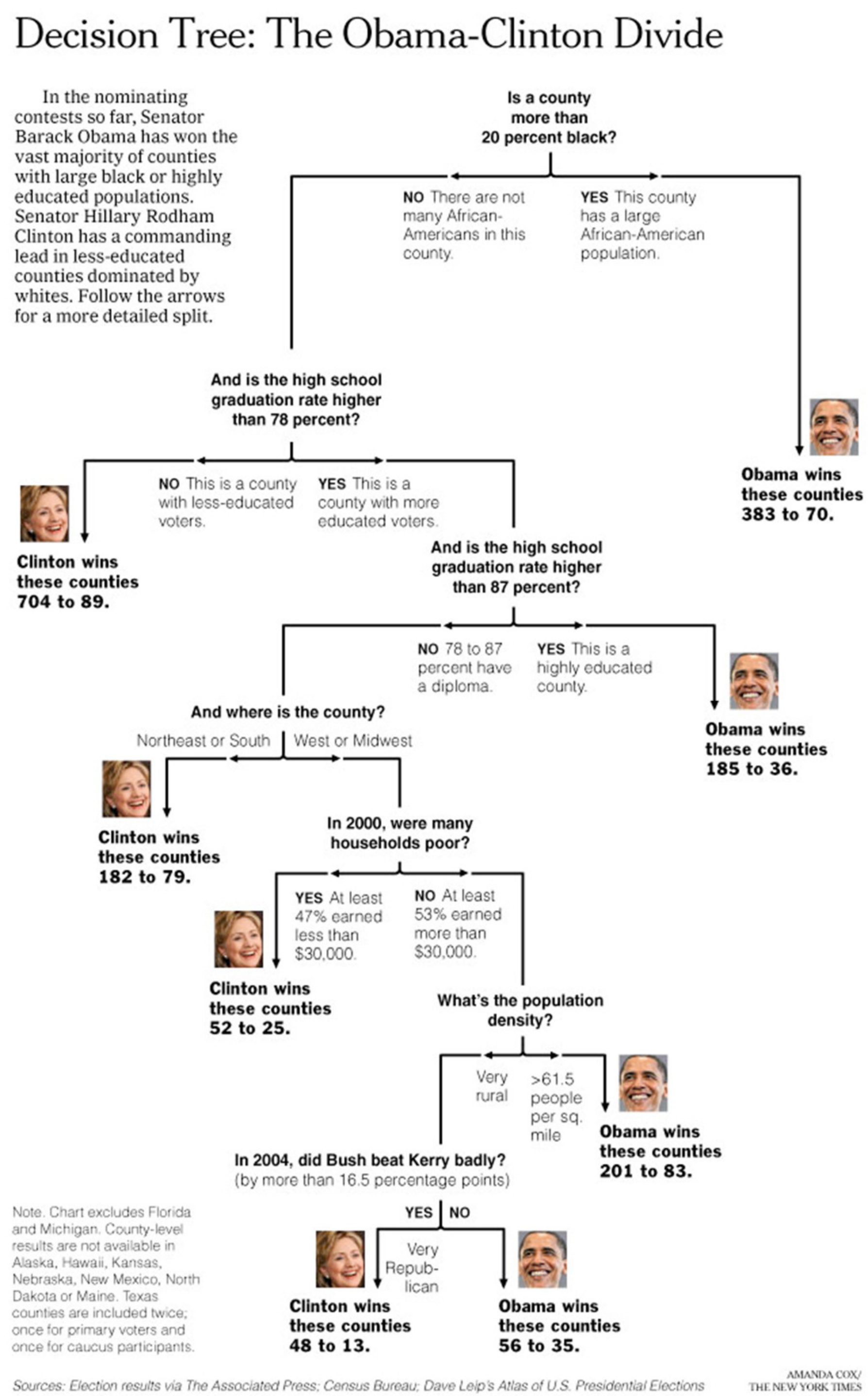
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



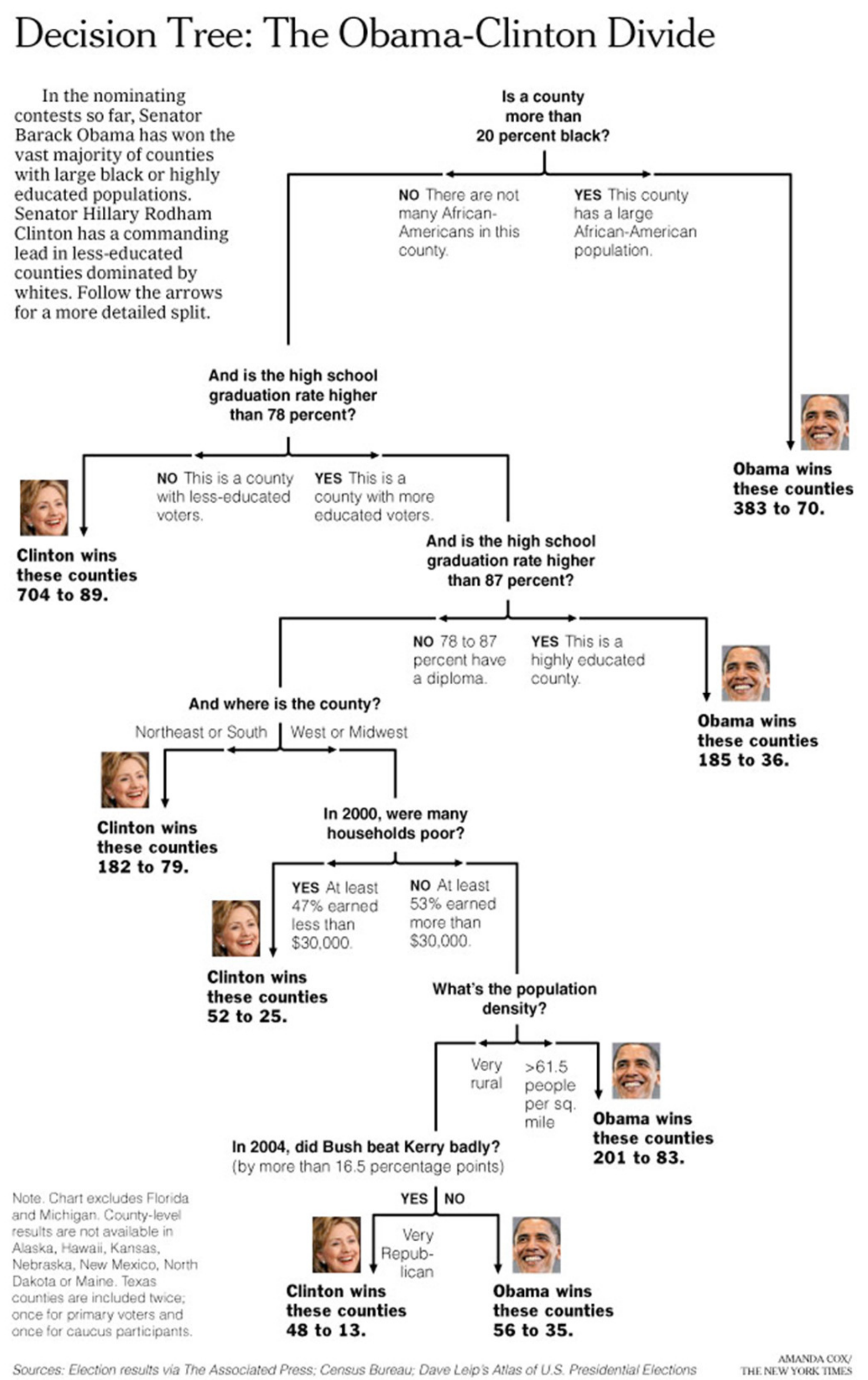
New York Times
April 16, 2008

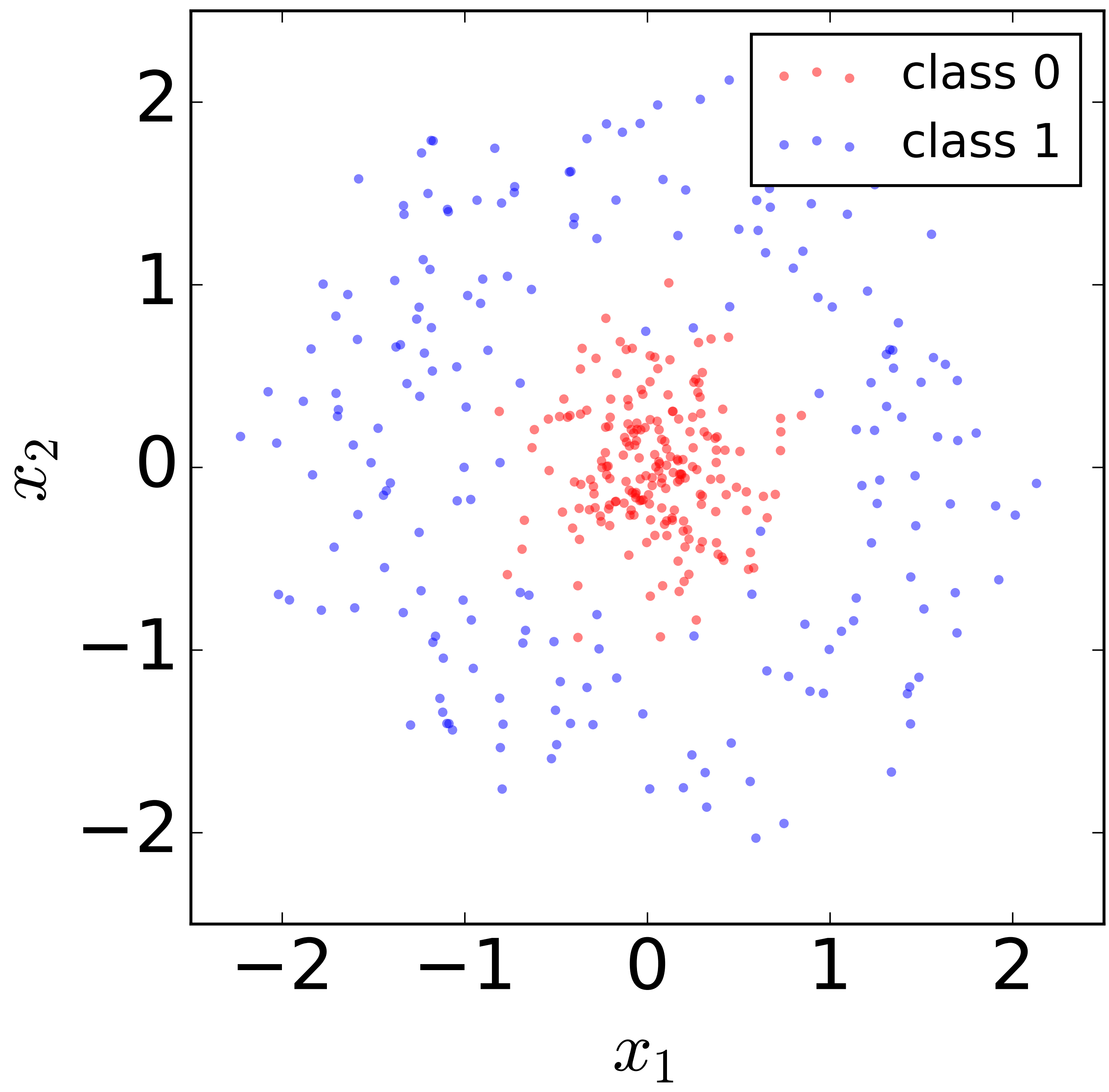
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

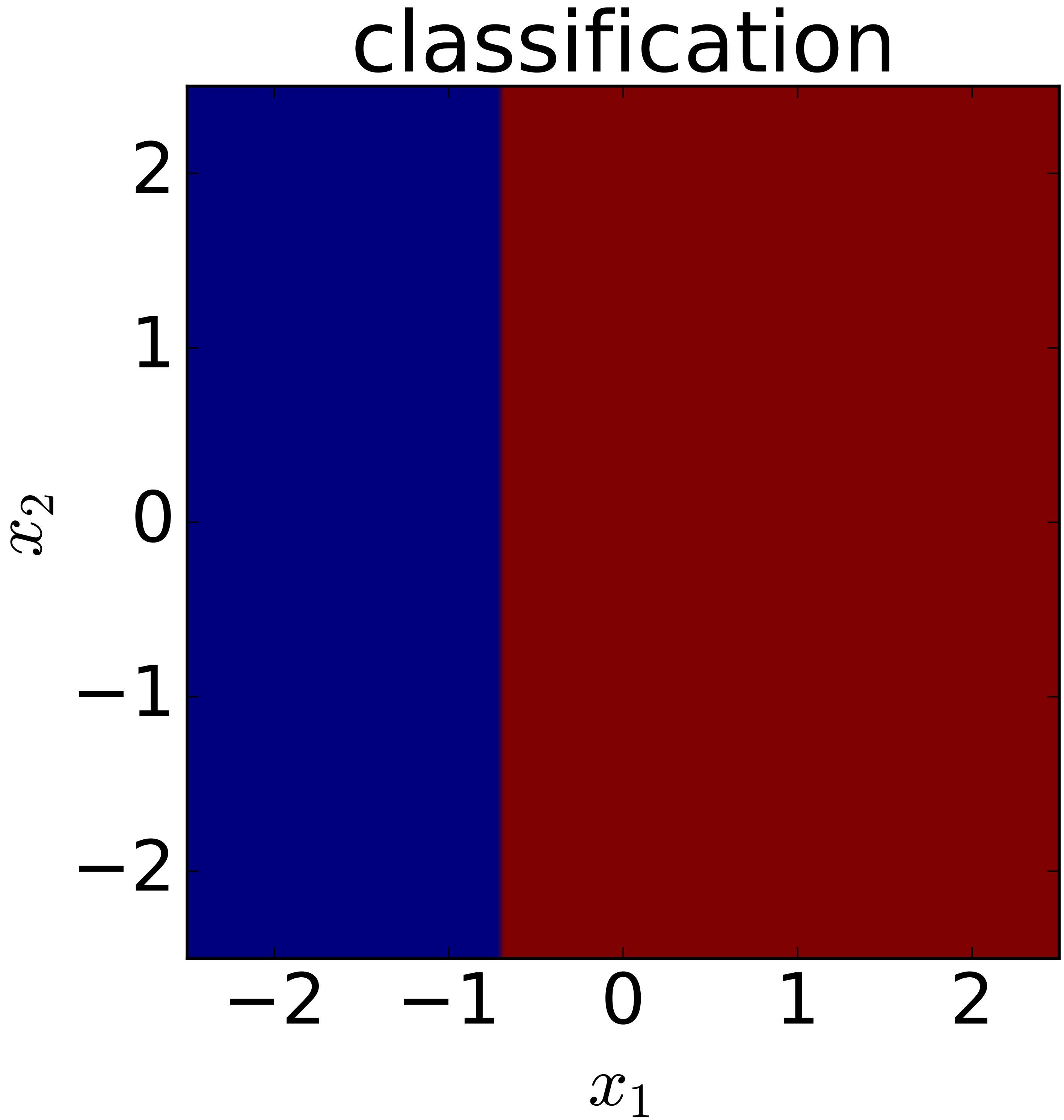


New York Times
April 16, 2008

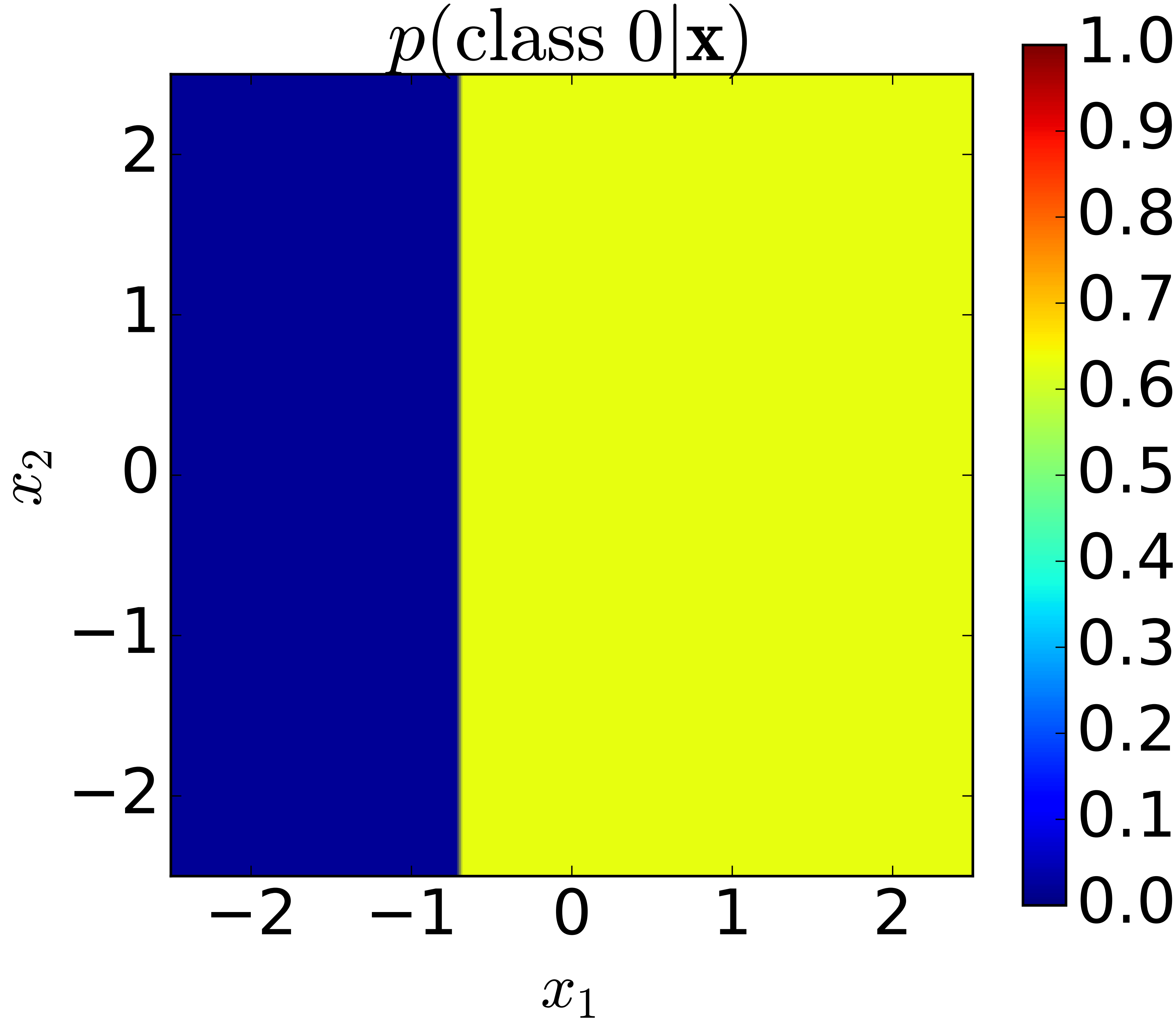
Decision Trees:
a sequence of tests.
Representation very natural
for humans.
Style of many "How to"
manuals and trouble-shooting
procedures.



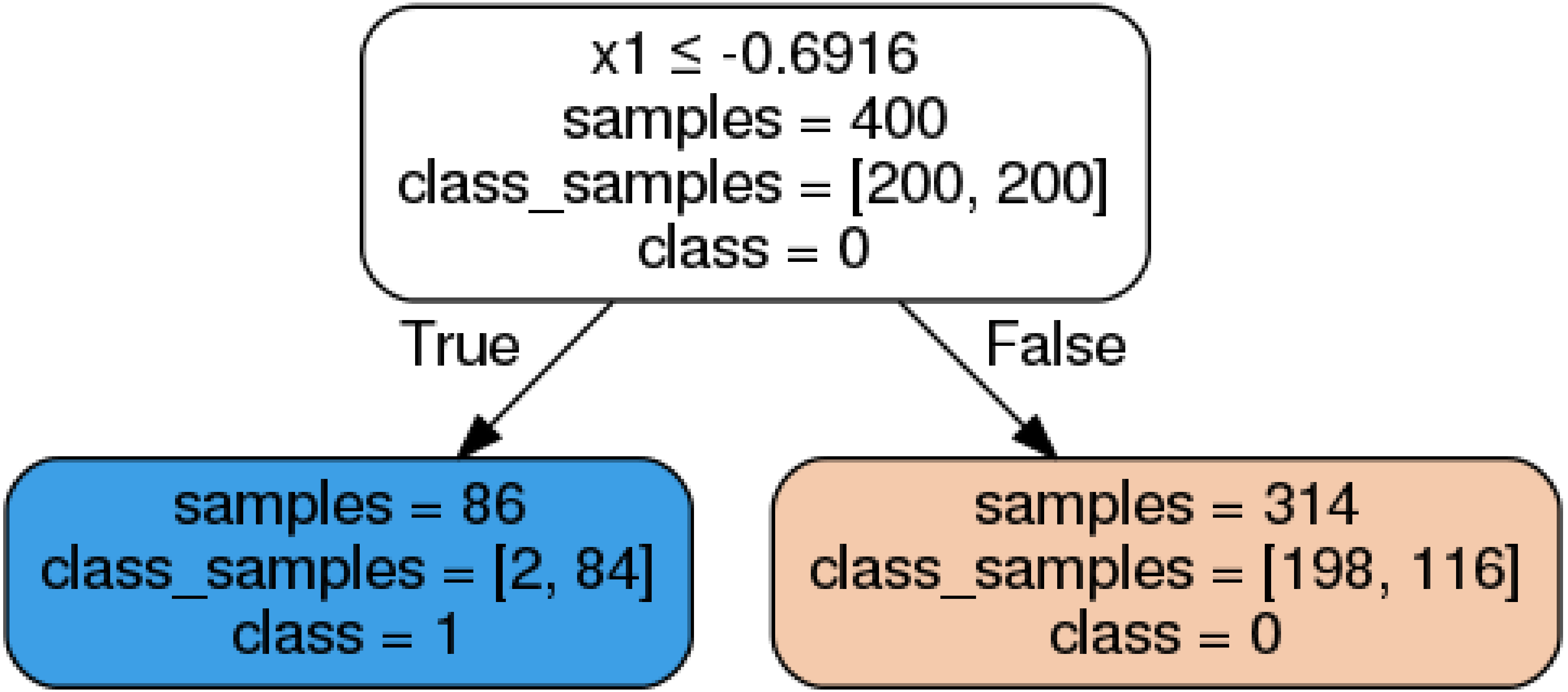

class 0
class 1
2
1
0
−1
−2
−2
−1
0
1
2
x_1
x_2

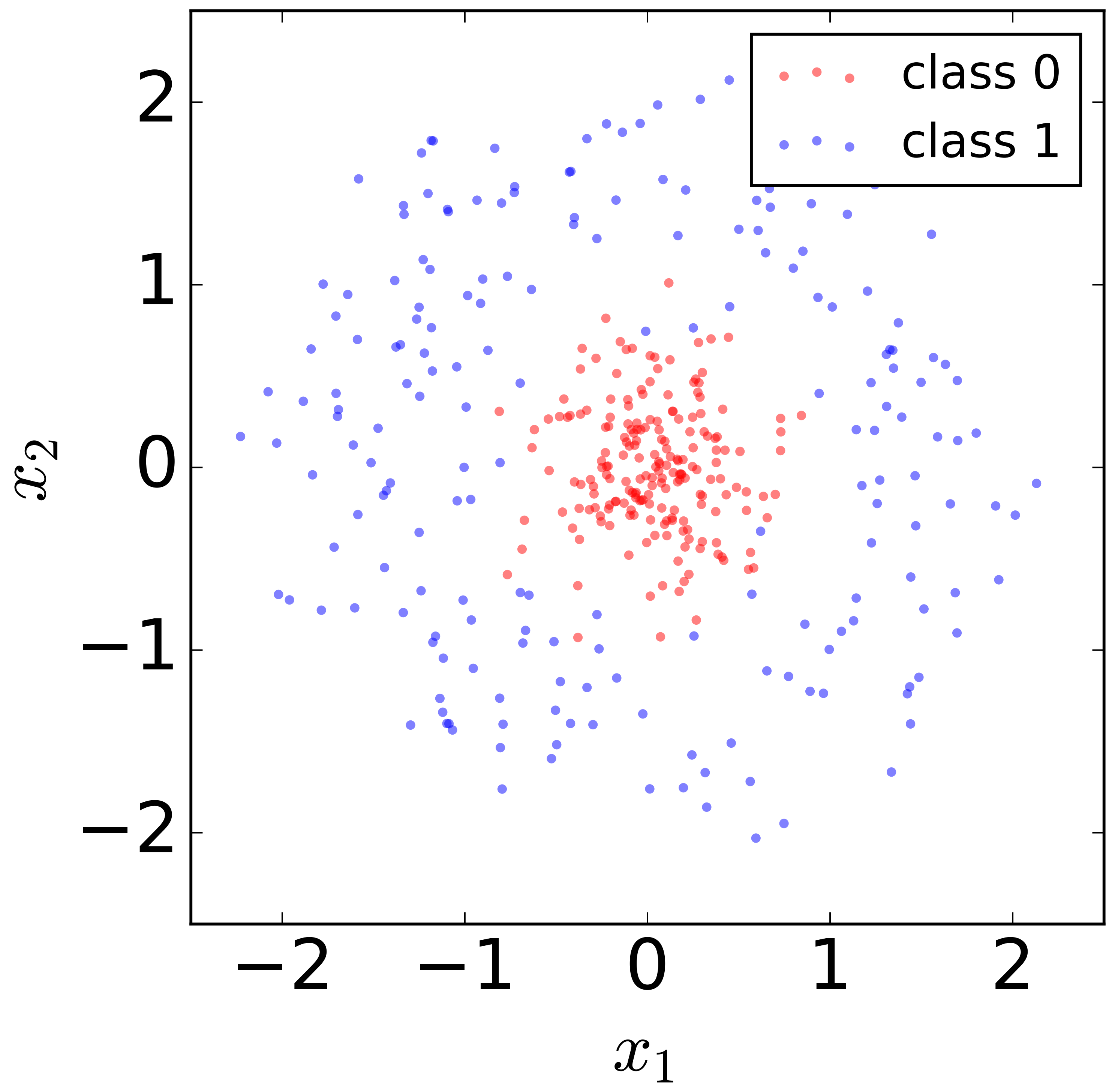


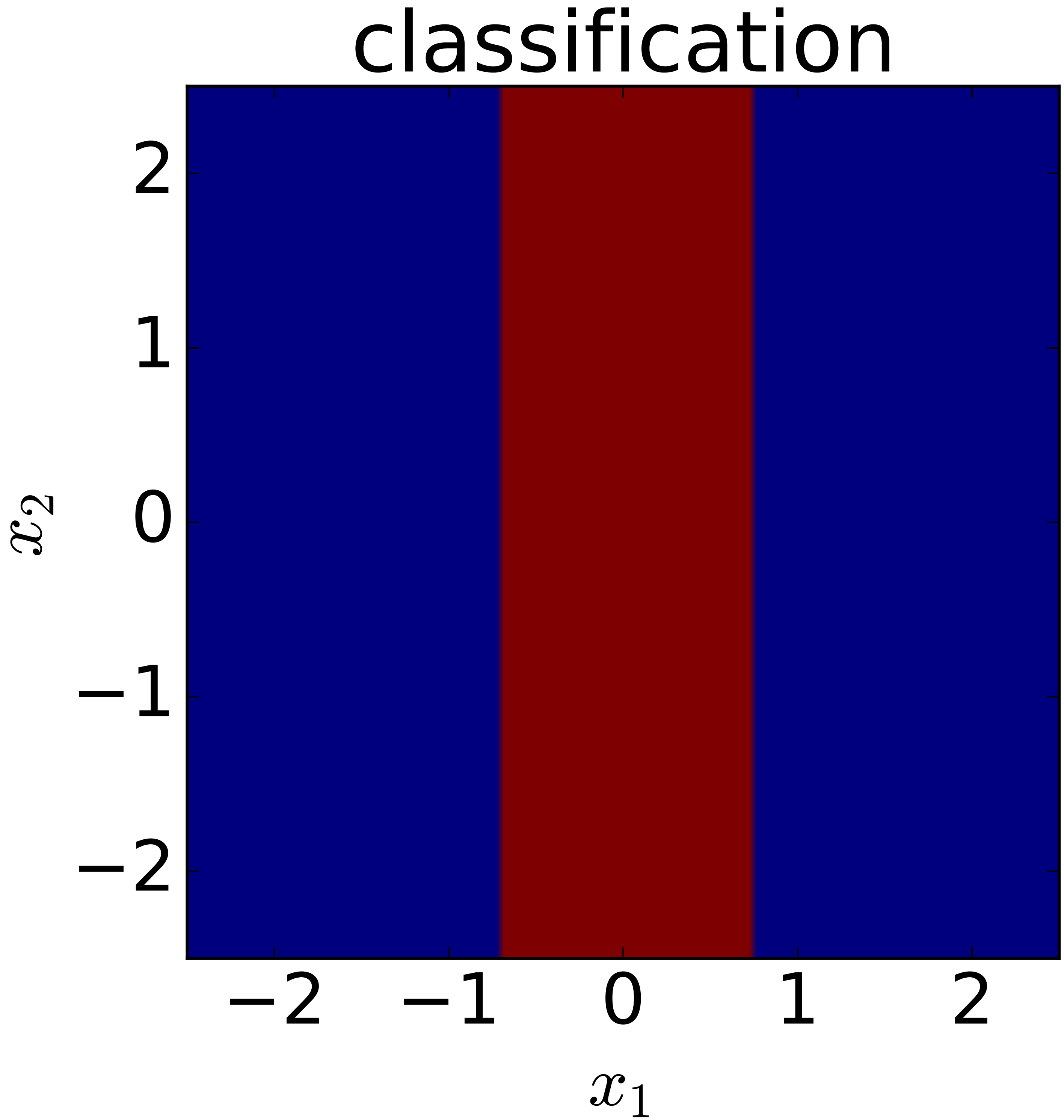
classification
2
1
0
−1
−2
x_2
−2
−1
0
1
2
x_1

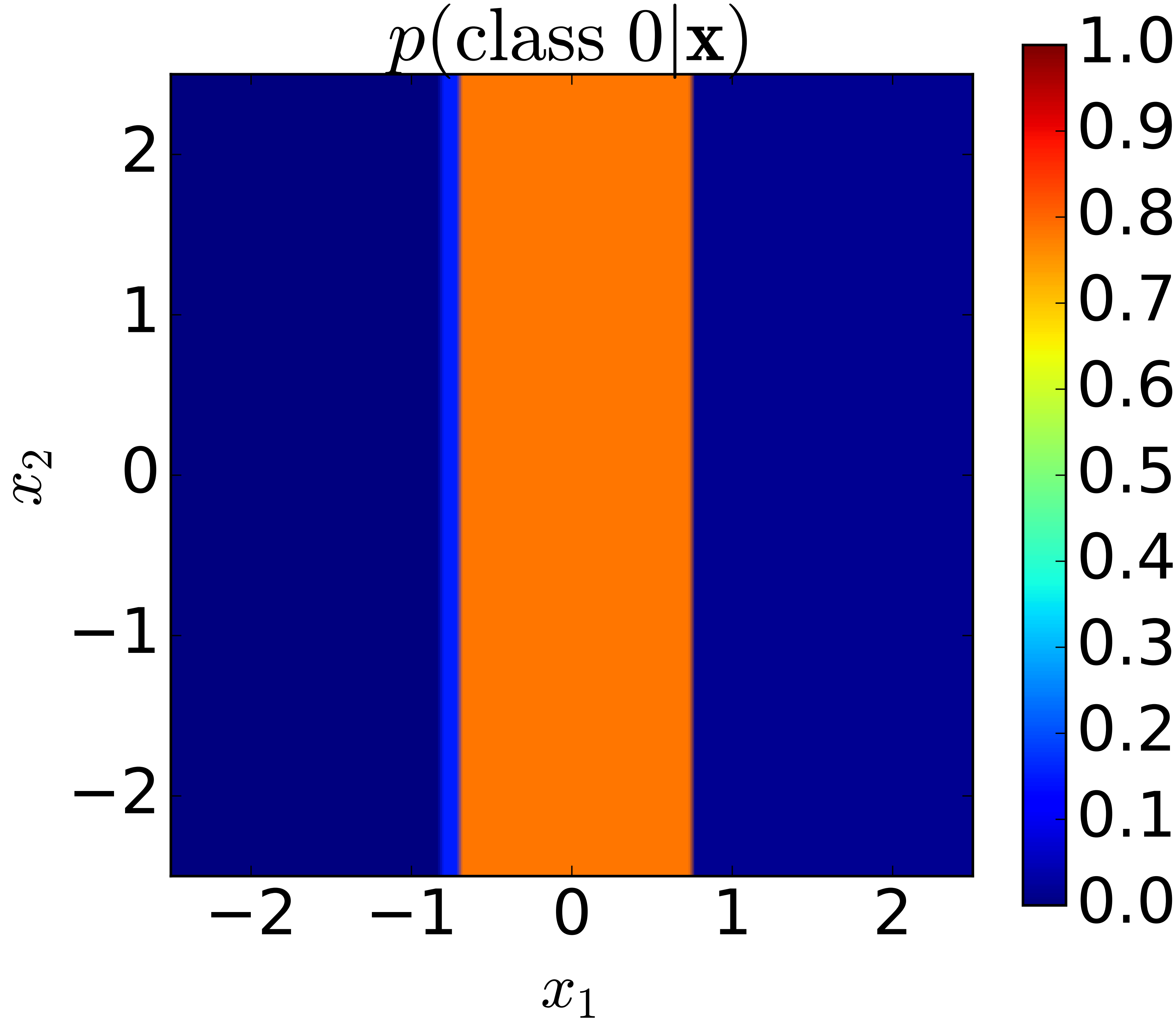


$p(\text{class } 0|\mathbf{x})$
x_2
x_1
2
1
0
−1
−2
−2
−1
0
1
2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

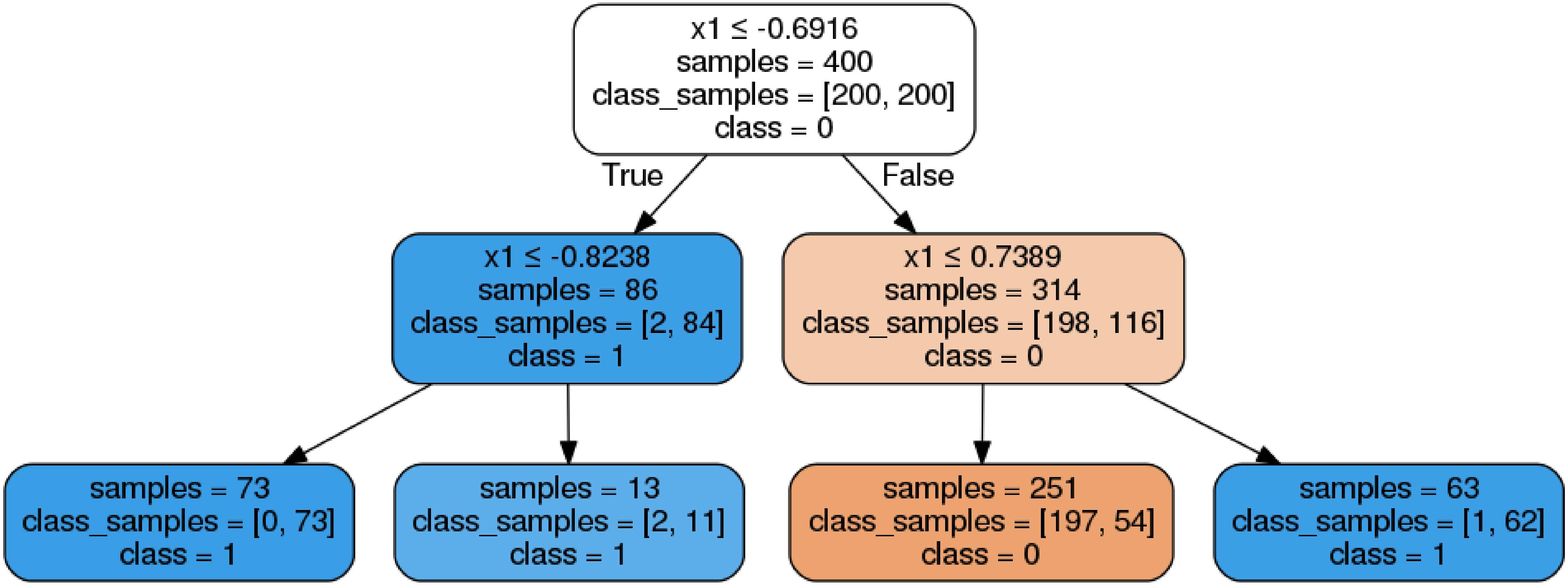

x1 ≤ -0.6916
samples = 400
class_samples = [200, 200]
class = 0
True
False
samples = 86
class_samples = [2, 84]
class = 1
samples = 314
class_samples = [198, 116]
class = 0

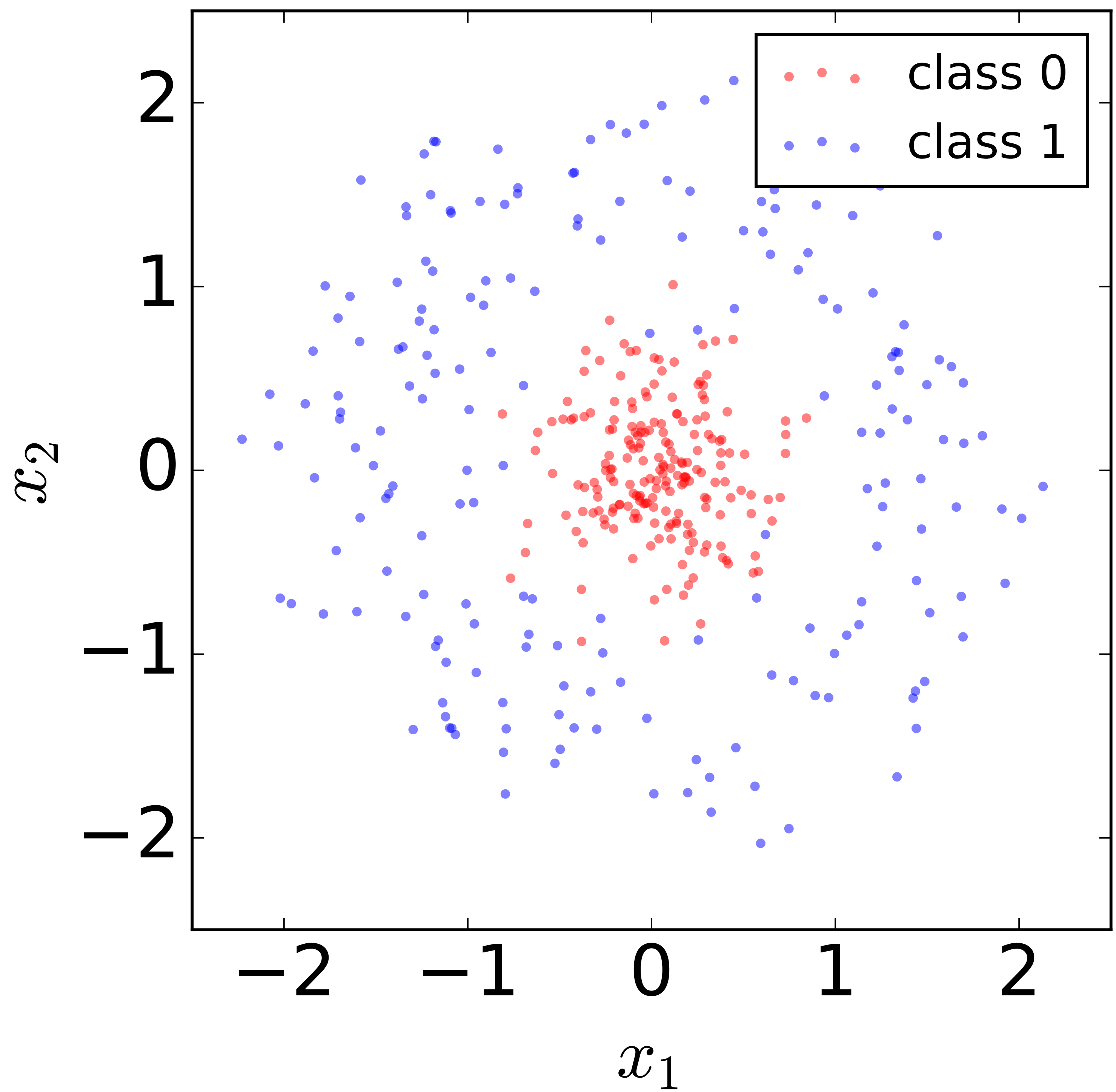

class 0
class 1
x_1
x_2
−2
−1
0
1
2

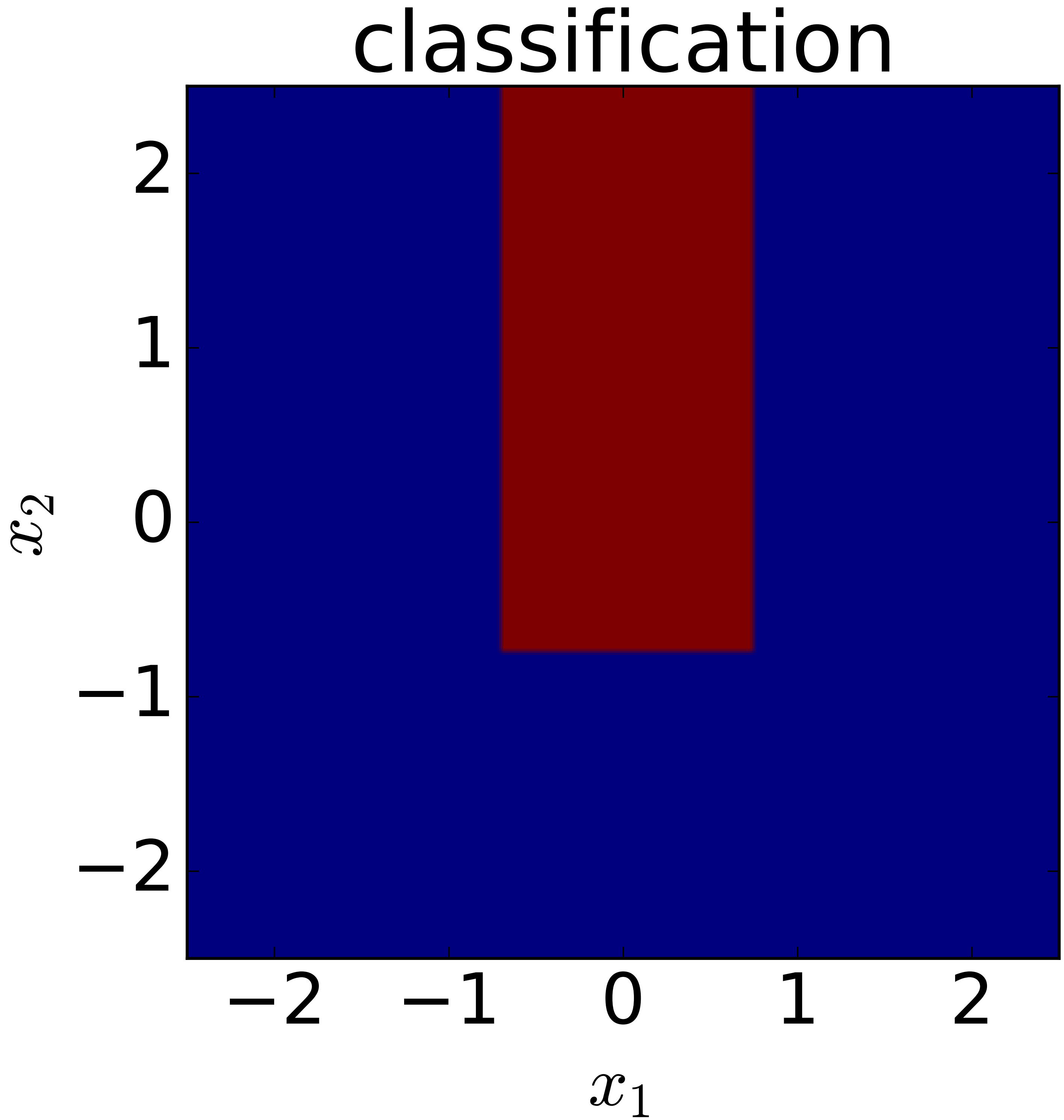

classification
2
1
0
−1
−2
x_2
−2
−1
0
1
2
x_1

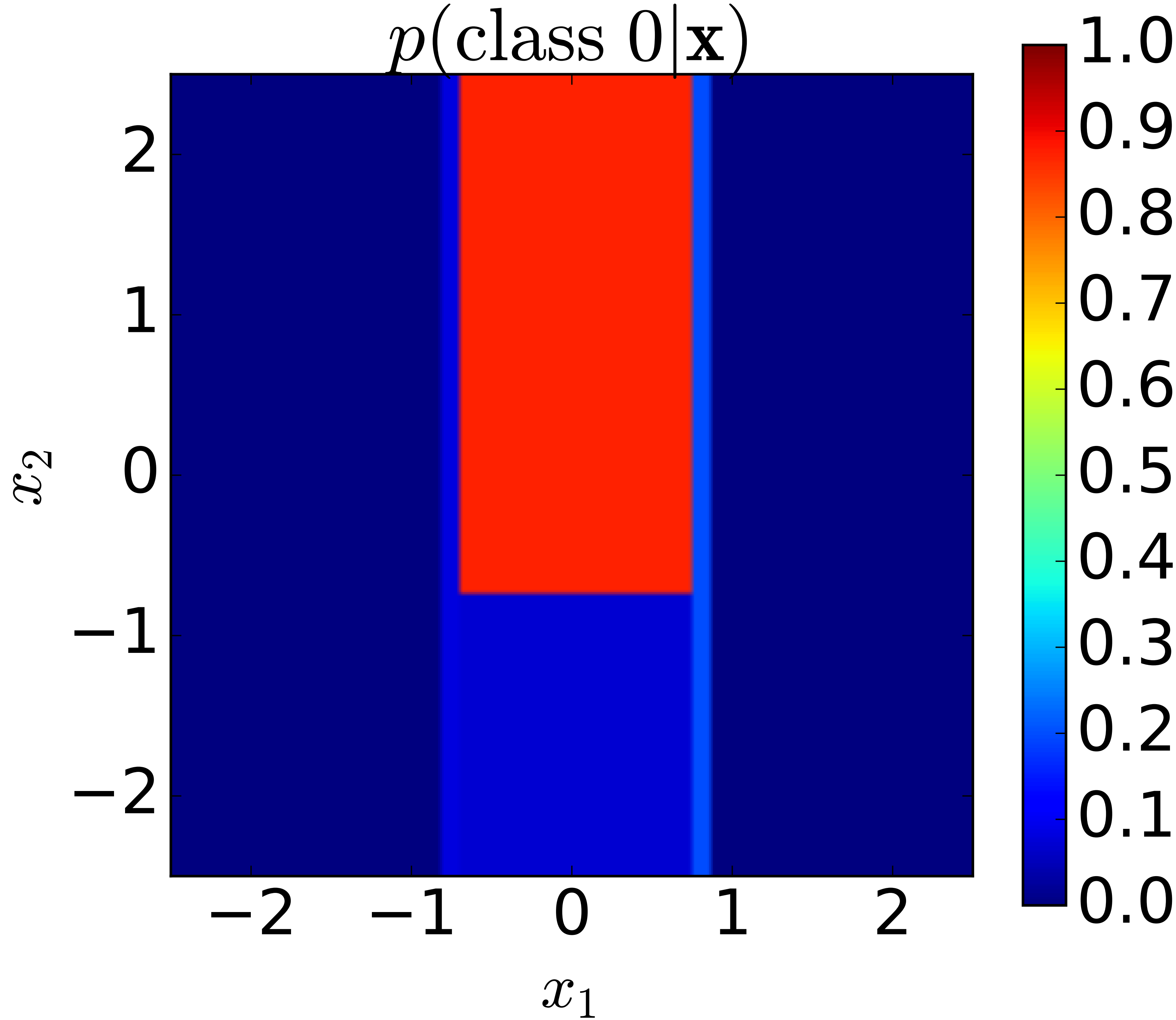


$p(\text{class } 0|\mathbf{x})$
x_1
x_2
−2
−1
0
1
2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

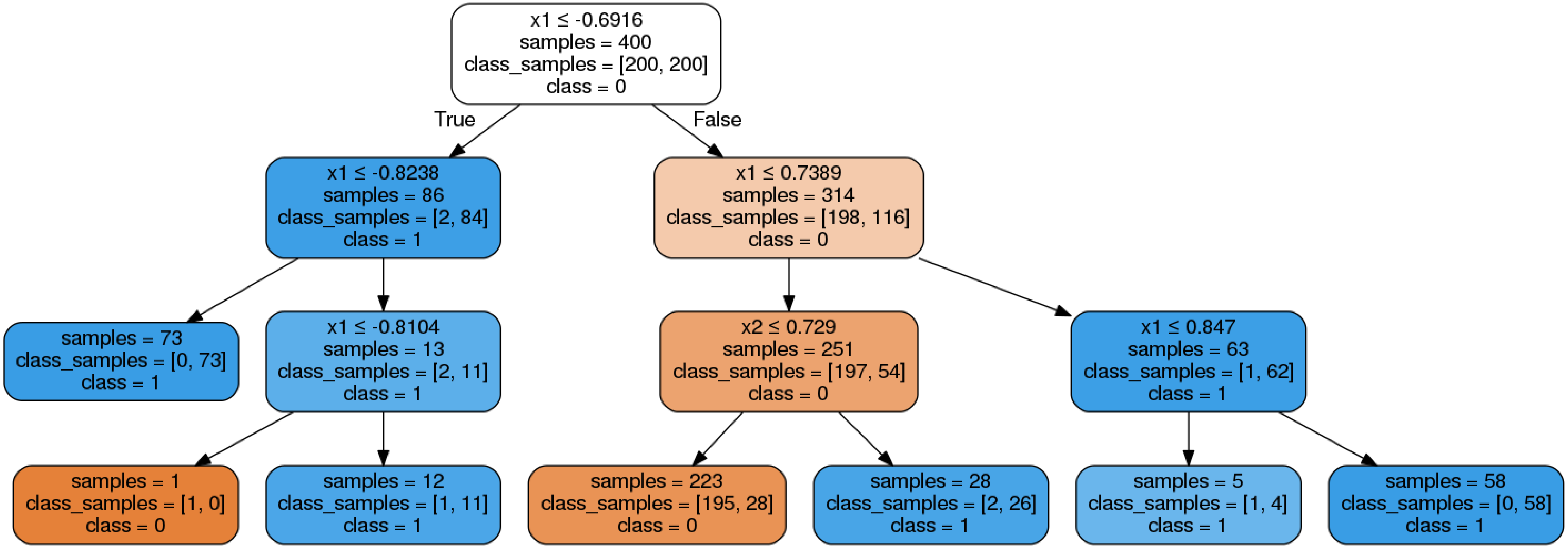

x1 ≤ -0.6916
samples = 400
class_samples = [200, 200]
class = 0
True
False
x1 ≤ -0.8238
samples = 86
class_samples = [2, 84]
class = 1
x1 ≤ 0.7389
samples = 314
class_samples = [198, 116]
class = 0
samples = 73
class_samples = [0, 73]
class = 1
samples = 13
class_samples = [2, 11]
class = 1
samples = 251
class_samples = [197, 54]
class = 0
samples = 63
class_samples = [1, 62]
class = 1

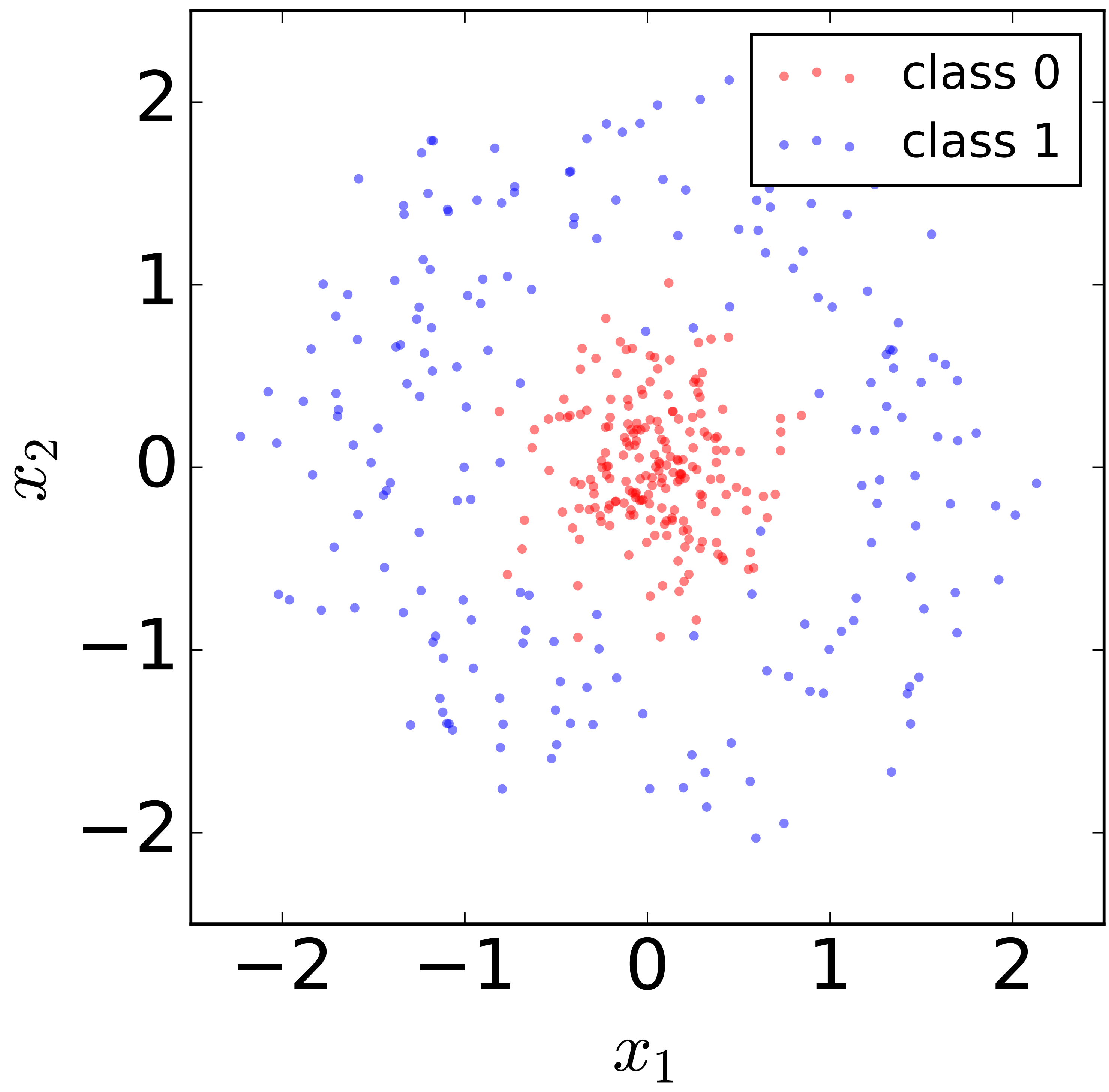

class 0
class 1
2
1
0
−1
−2
−2
−1
0
1
2
x_2
x_1

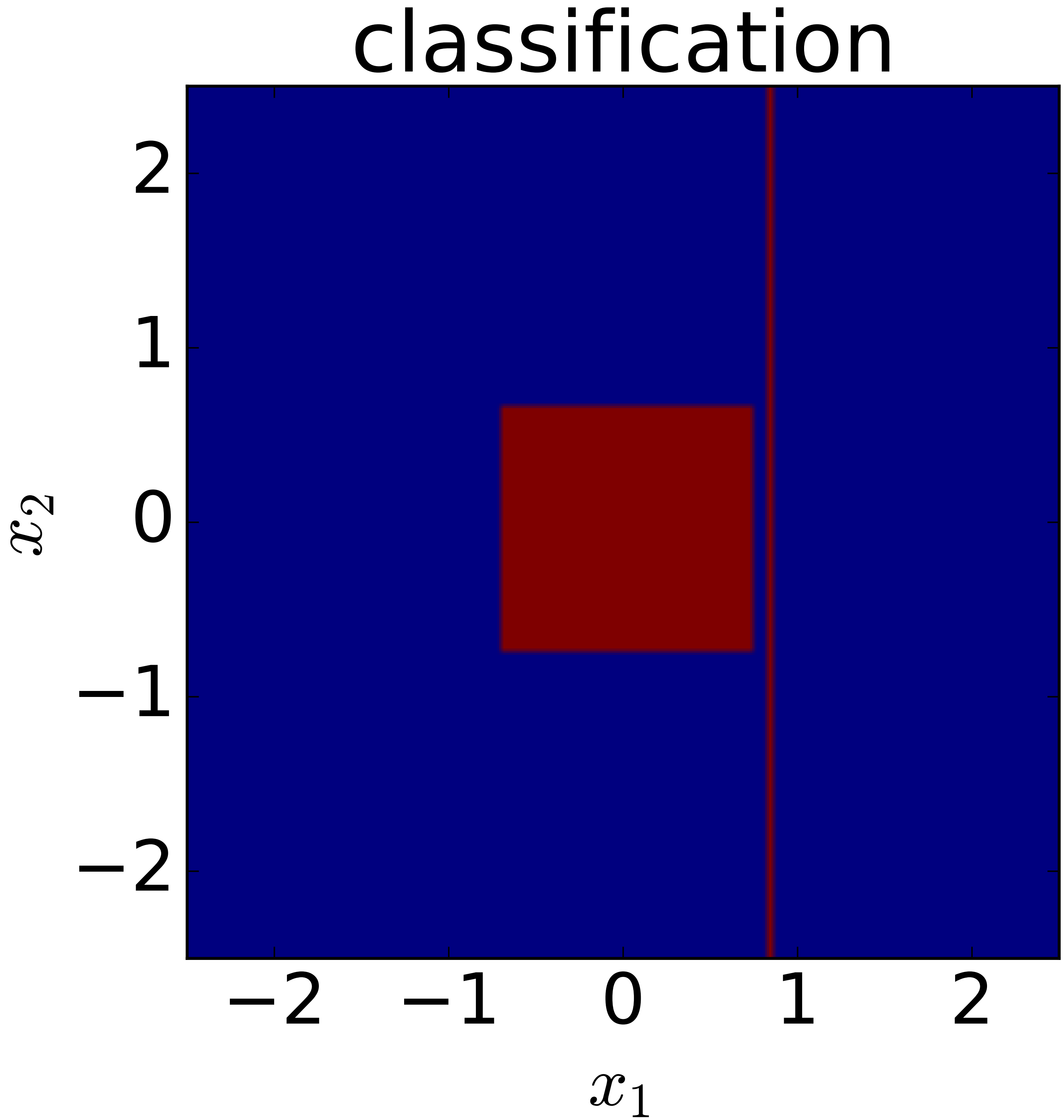

classification
2
1
0
−1
−2
x_2
−2
−1
0
1
2
x_1



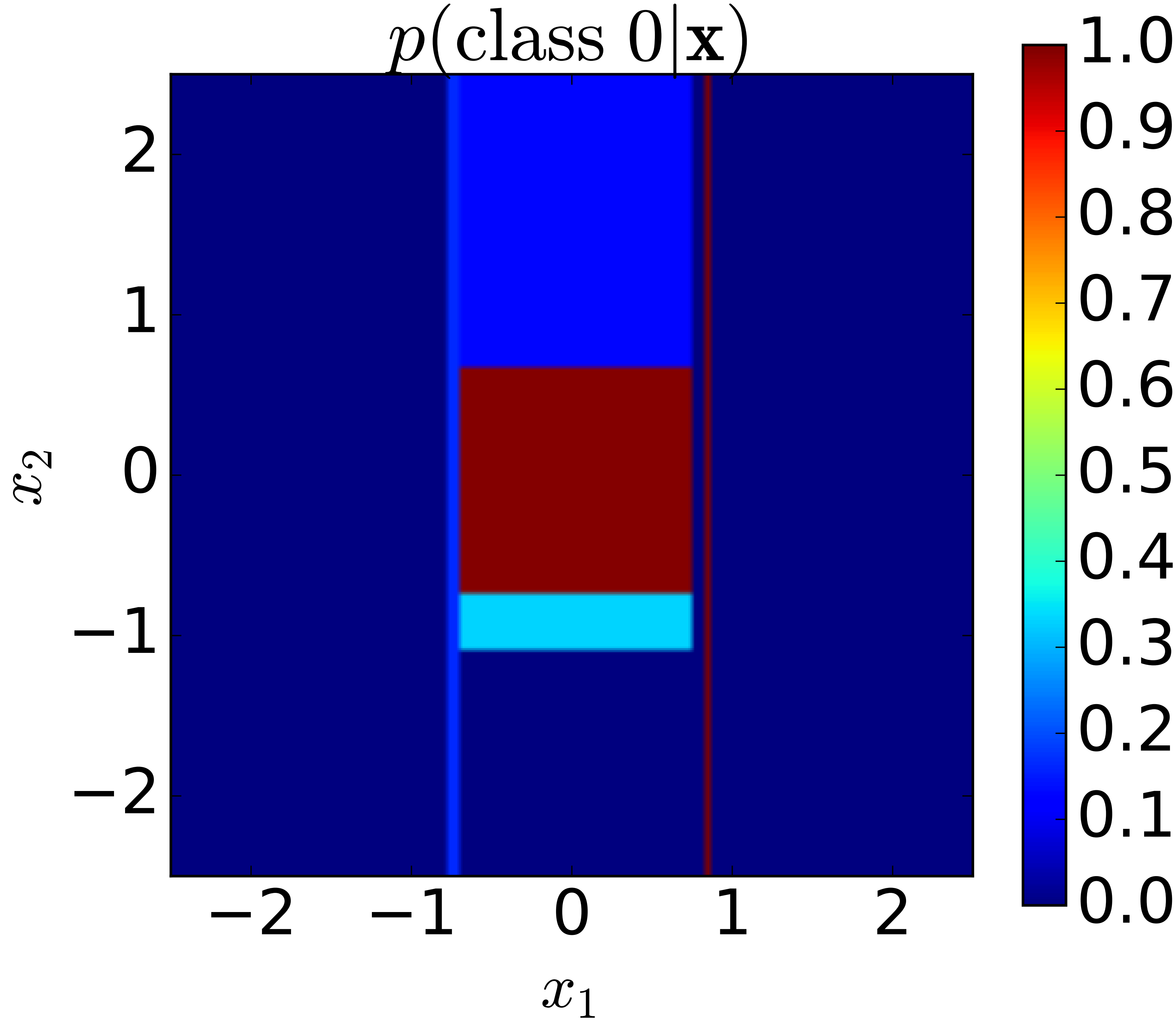
$p(\text{class } 0|\mathbf{x})$
x_1
x_2
−2
−1
0
1
2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


x1 ≤ -0.6916
samples = 400
class_samples = [200, 200]
class = 0
True
False
x1 ≤ -0.8238
samples = 86
class_samples = [2, 84]
class = 1
x1 ≤ 0.7389
samples = 314
class_samples = [198, 116]
class = 0
samples = 73
class_samples = [0, 73]
class = 1
x1 ≤ -0.8104
samples = 13
class_samples = [2, 11]
class = 1
x2 ≤ 0.729
samples = 251
class_samples = [197, 54]
class = 0
x1 ≤ 0.847
samples = 63
class_samples = [1, 62]
class = 1
samples = 1
class_samples = [1, 0]
class = 0
samples = 12
class_samples = [1, 11]
class = 1
samples = 223
class_samples = [195, 28]
class = 0
samples = 28
class_samples = [2, 26]
class = 1
samples = 5
class_samples = [1, 4]
class = 1
samples = 58
class_samples = [0, 58]
class = 1

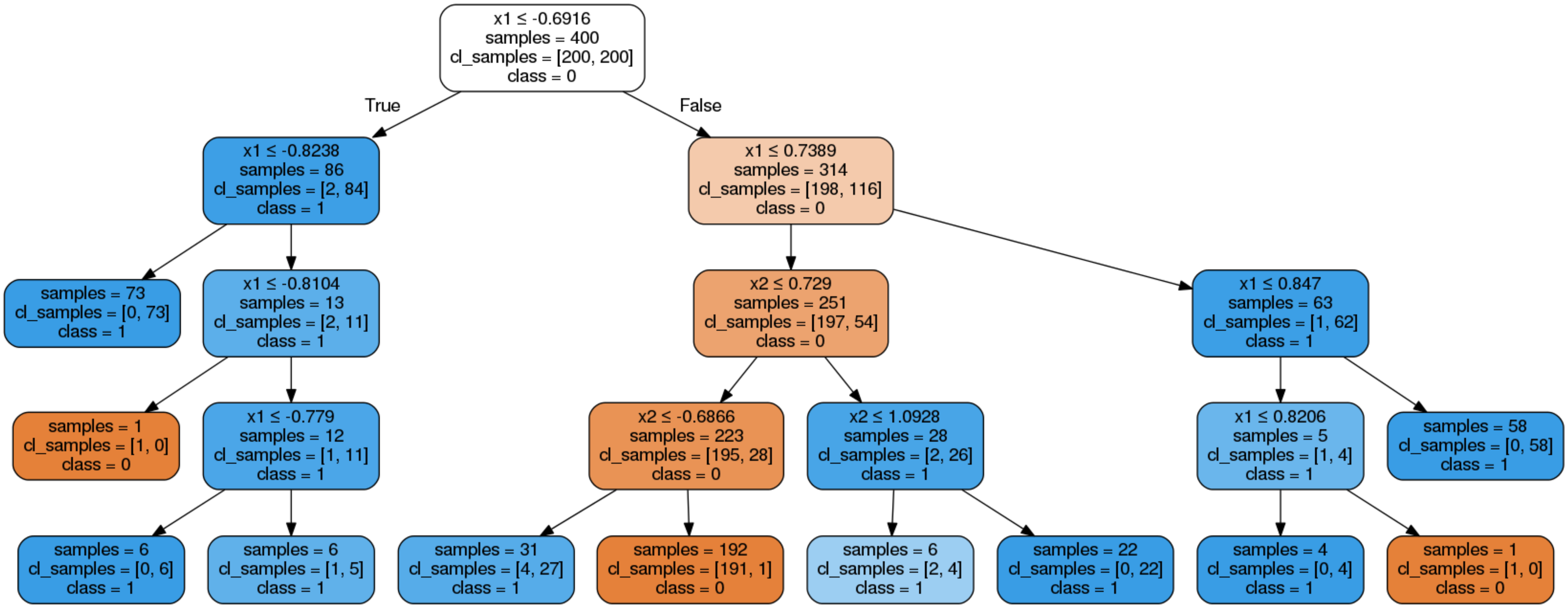

class 0
class 1
x_1
x_2
−2
−1
0
1
2



classification
2
1
0
−1
−2
x_2
−2
−1
0
1
2
x_1

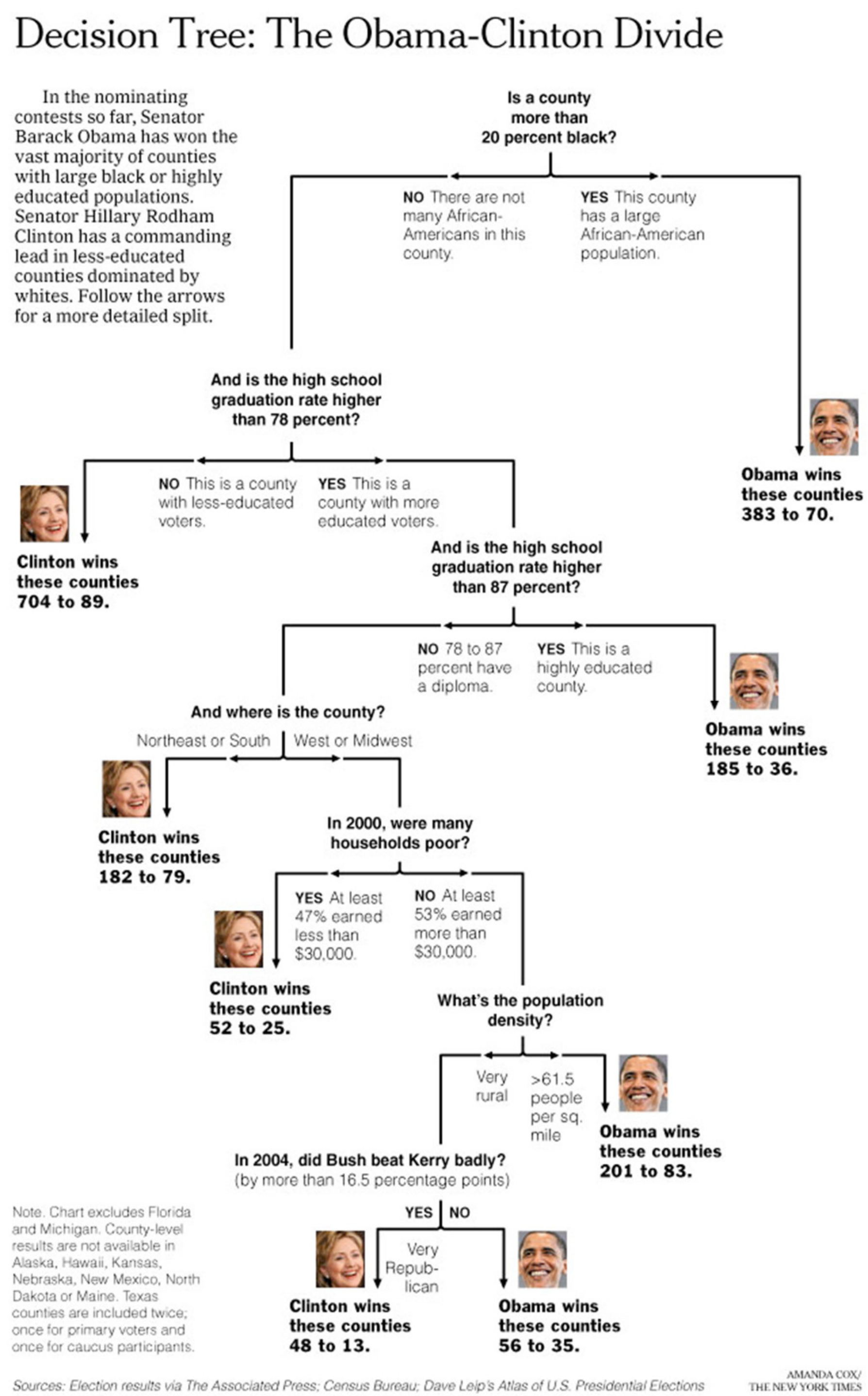


$p(\text{class } 0|\mathbf{x})$
2
1
0
−1
−2
x_2
−2
−1
0
1
2
x_1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


x1 ≤ -0.6916
samples = 400
cl_samples = [200, 200]
class = 0
True
False
x1 ≤ -0.8238
samples = 86
cl_samples = [2, 84]
class = 1
x1 ≤ 0.7389
samples = 314
cl_samples = [198, 116]
class = 0
samples = 73
cl_samples = [0, 73]
class = 1
x1 ≤ -0.8104
samples = 13
cl_samples = [2, 11]
class = 1
x2 ≤ 0.729
samples = 251
cl_samples = [197, 54]
class = 0
x1 ≤ 0.847
samples = 63
cl_samples = [1, 62]
class = 1
samples = 1
cl_samples = [1, 0]
class = 0
x1 ≤ -0.779
samples = 12
cl_samples = [1, 11]
class = 1
x2 ≤ -0.6866
samples = 223
cl_samples = [195, 28]
class = 0
x2 ≤ 1.0928
samples = 28
cl_samples = [2, 26]
class = 1
x1 ≤ 0.8206
samples = 5
cl_samples = [1, 4]
class = 1
samples = 58
cl_samples = [0, 58]
class = 1
samples = 6
cl_samples = [0, 6]
class = 1
samples = 6
cl_samples = [1, 5]
class = 1
samples = 31
cl_samples = [4, 27]
class = 1
samples = 192
cl_samples = [191, 1]
class = 0
samples = 6
cl_samples = [2, 4]
class = 1
samples = 22
cl_samples = [0, 22]
class = 1
samples = 4
cl_samples = [0, 4]
class = 1
samples = 1
cl_samples = [1, 0]
class = 0

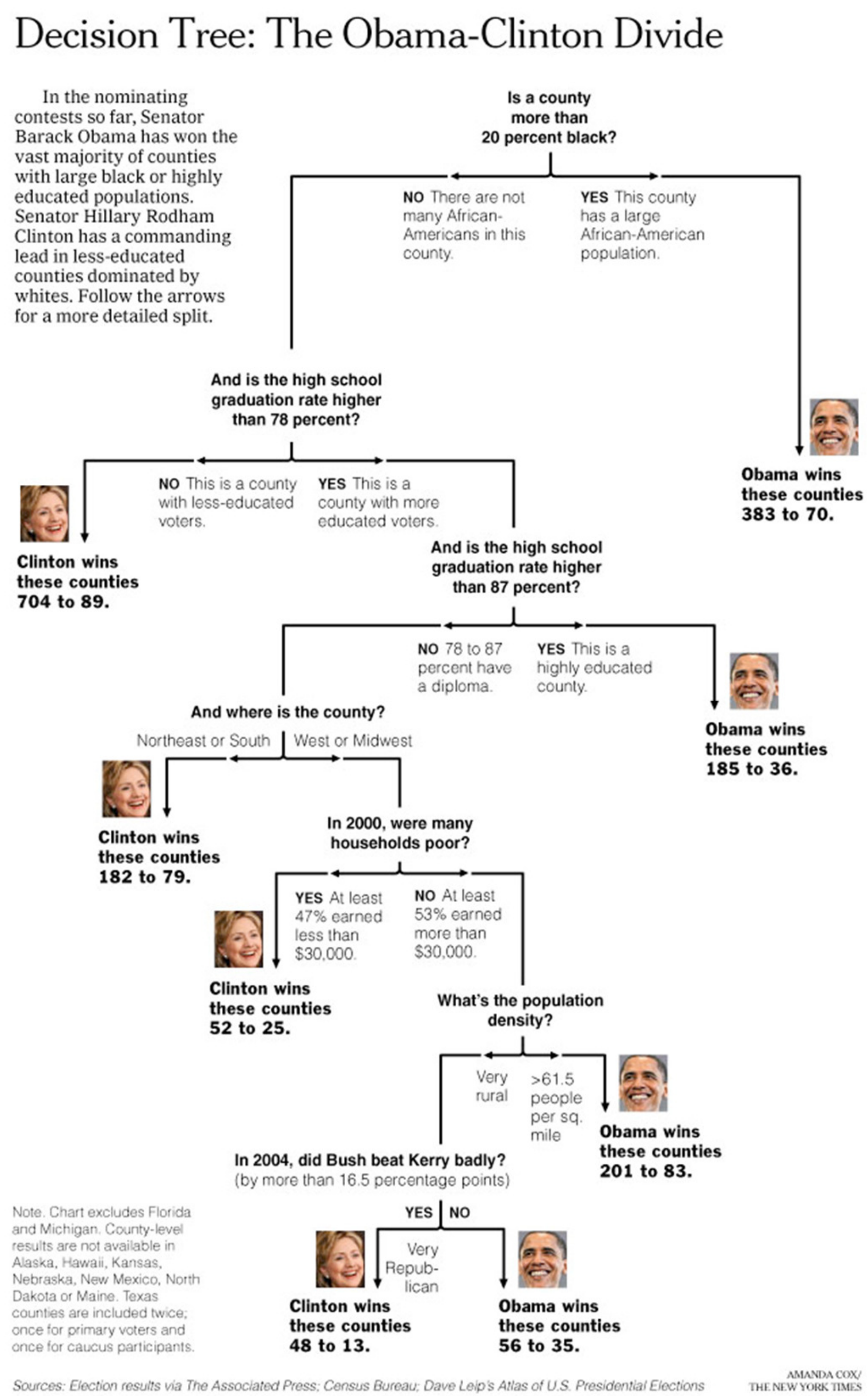
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

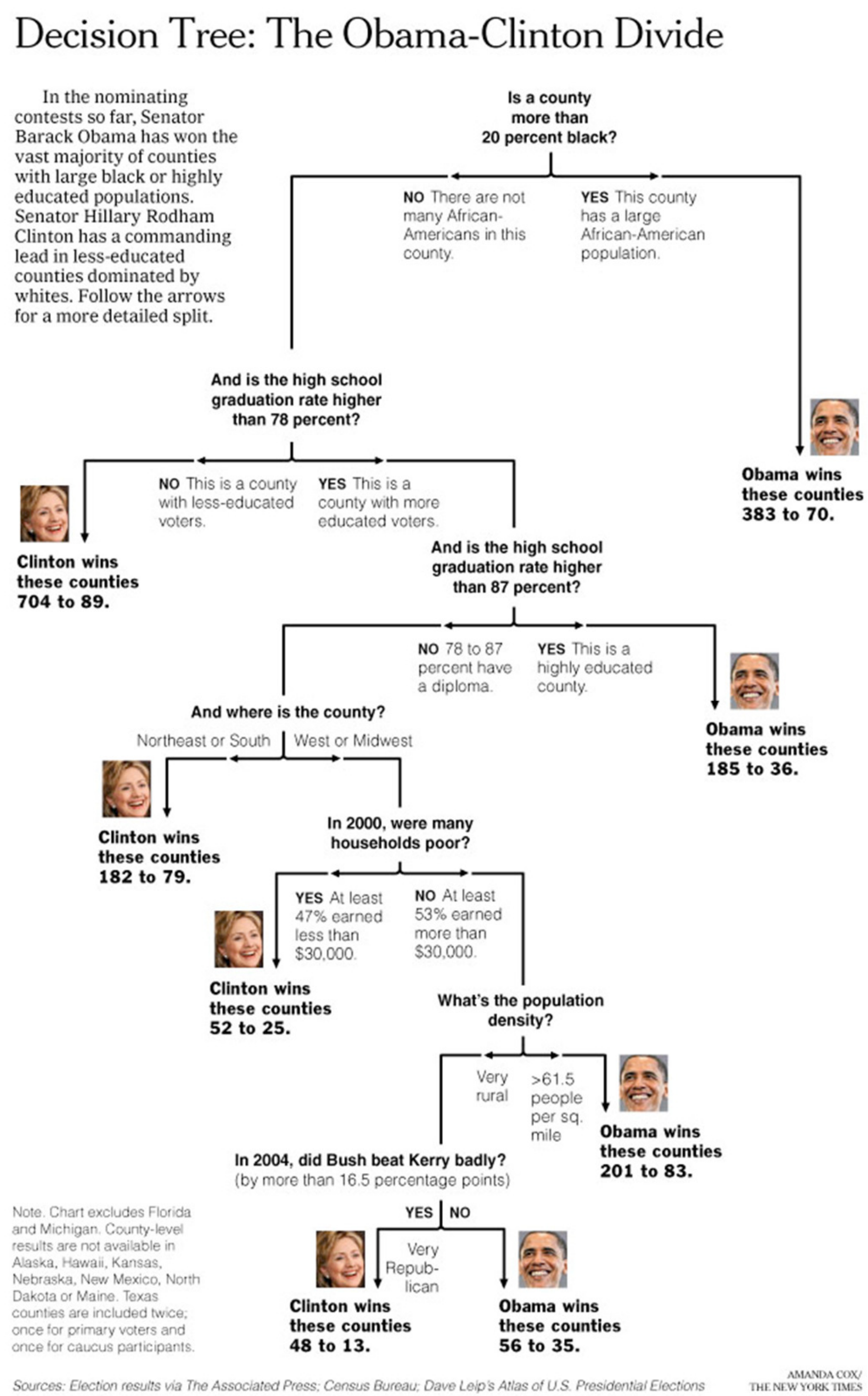


New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



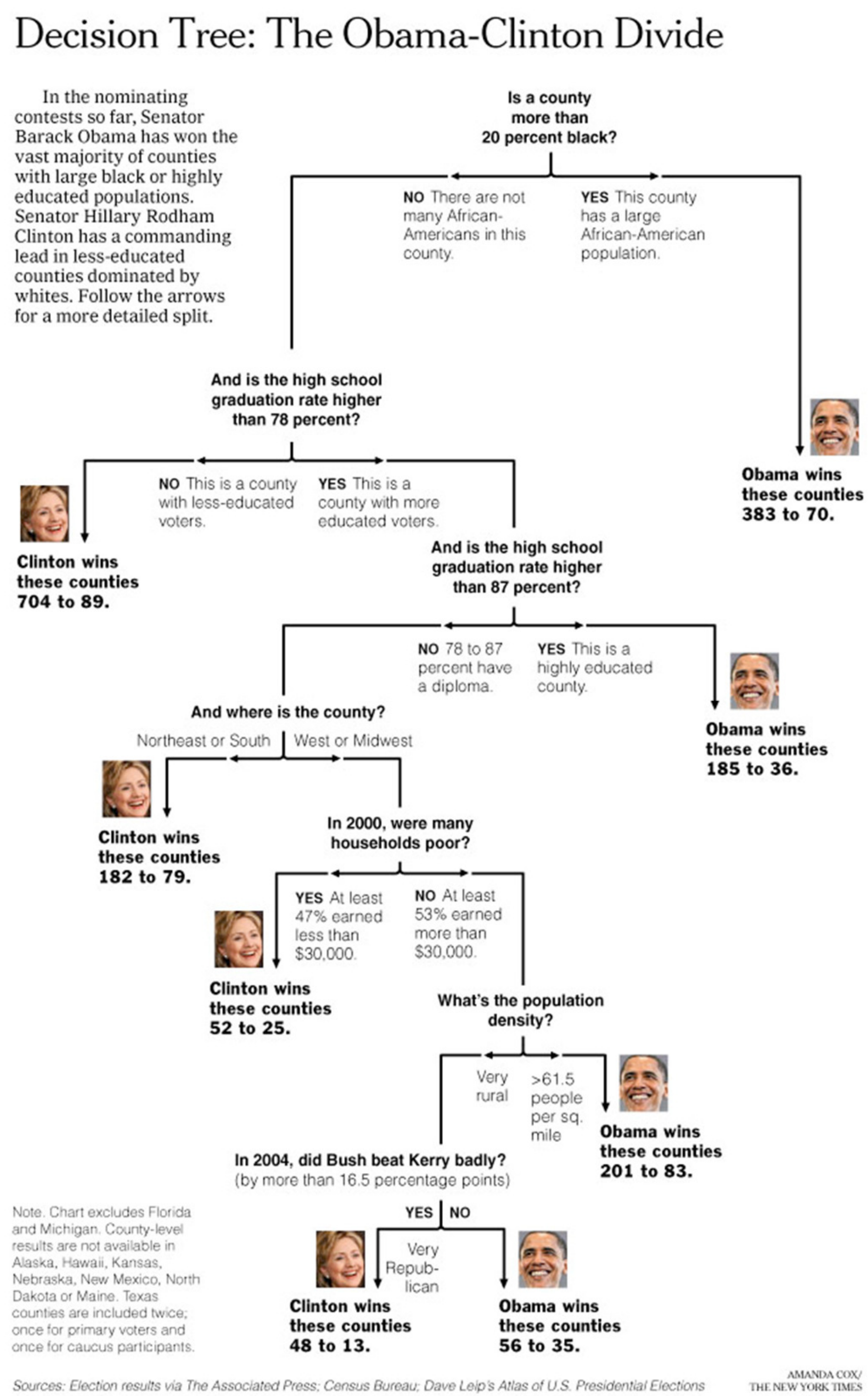
New York Times
April 16, 2008



Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008

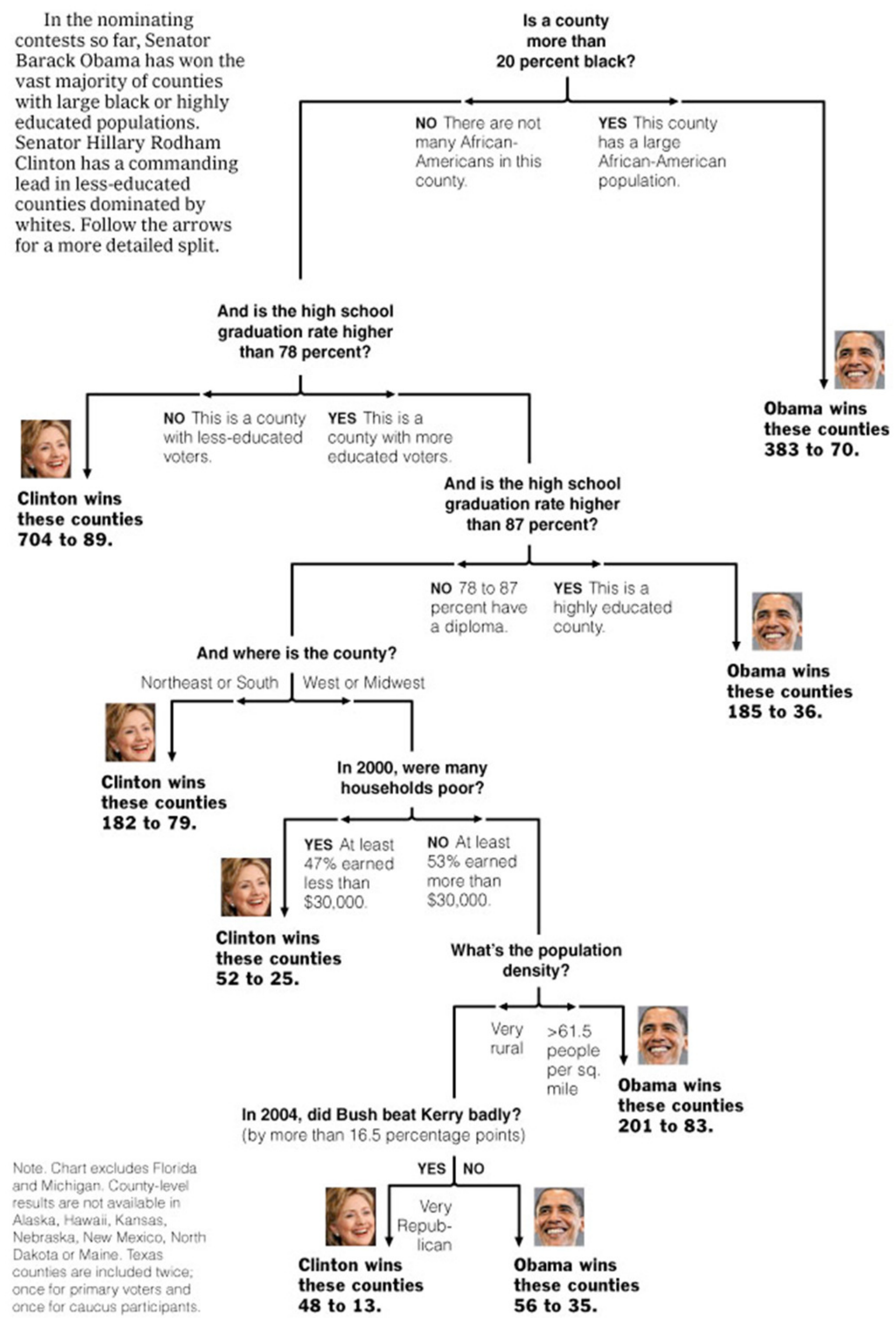
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural
for humans.
Style of many “How to”
manuals and trouble-shooting
procedures.

Decision Tree: The Obama-Clinton Divide

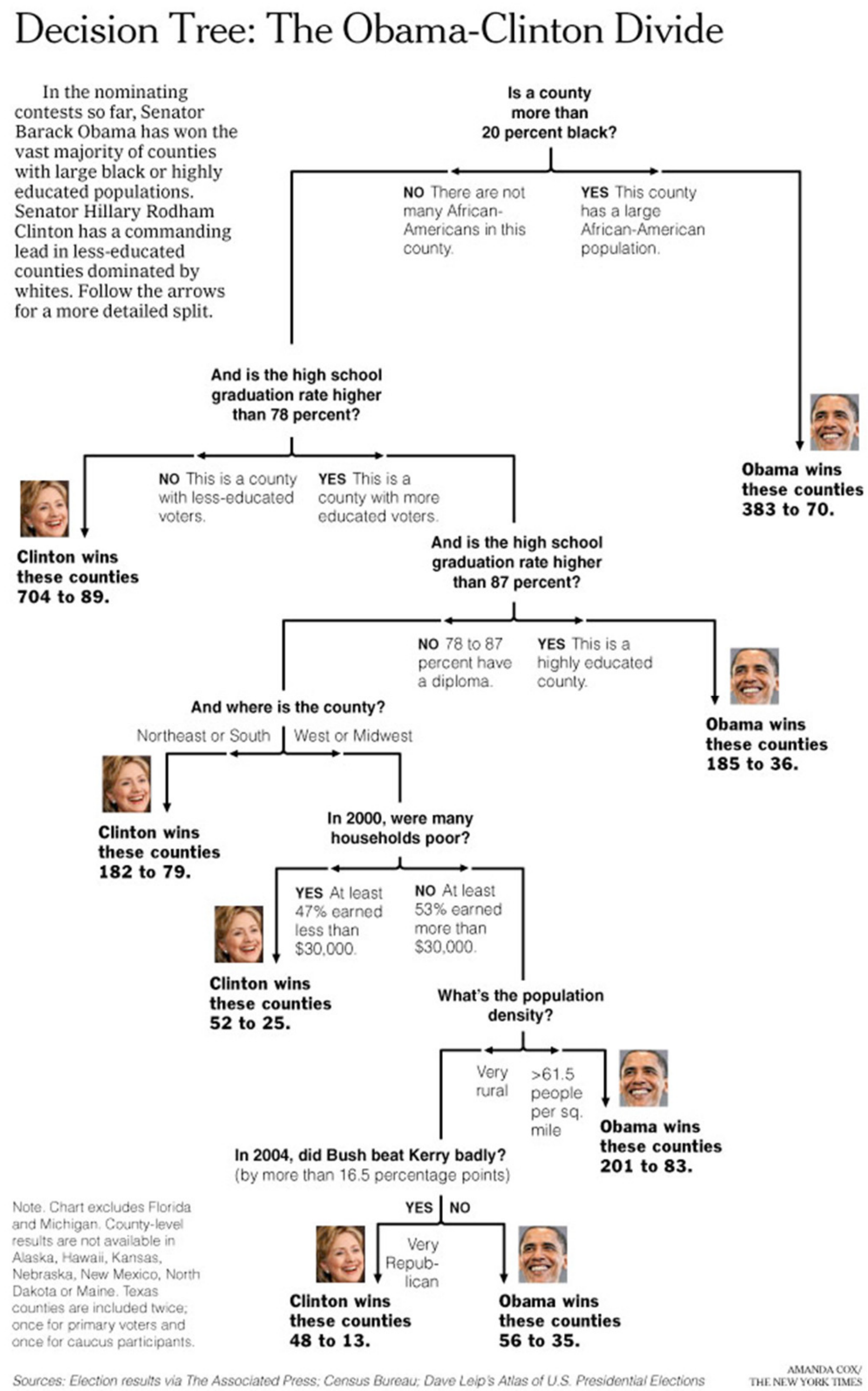


Sources: Election results via The Associated Press; Census Bureau; Dave Leip's Atlas of U.S. Presidential Elections

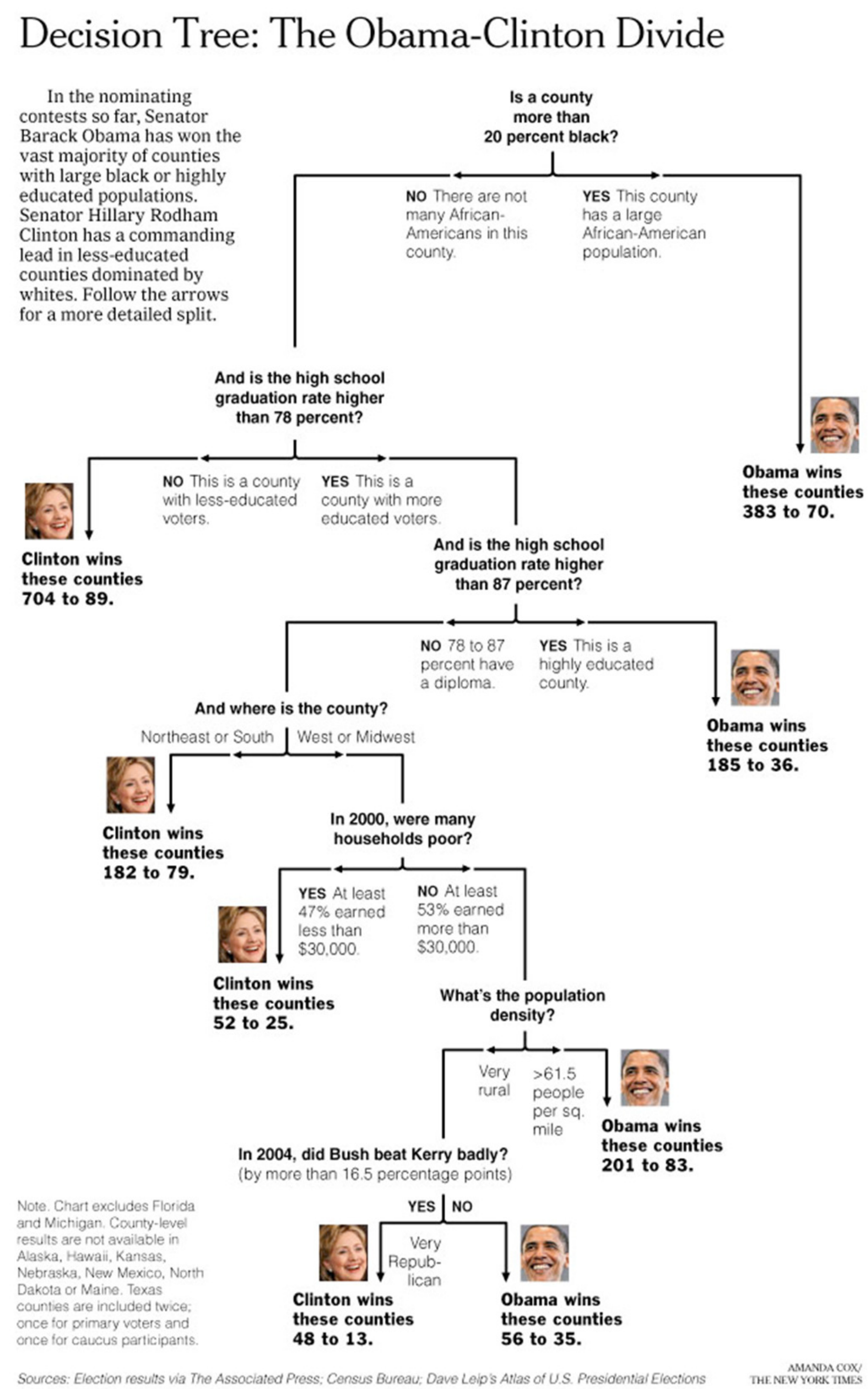
AMANDA COX/
THE NEW YORK TIMES

New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



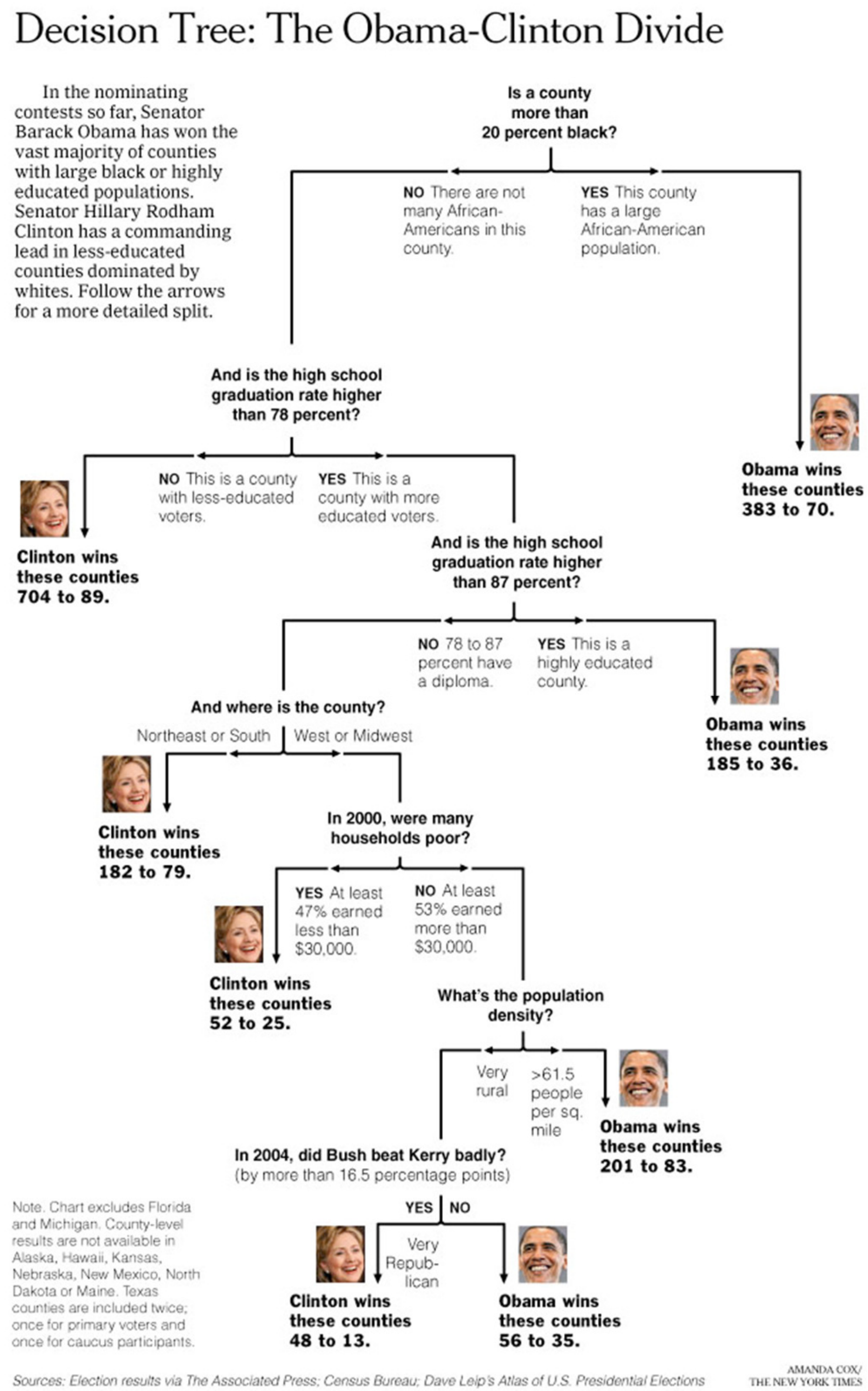
New York Times
April 16, 2008



Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

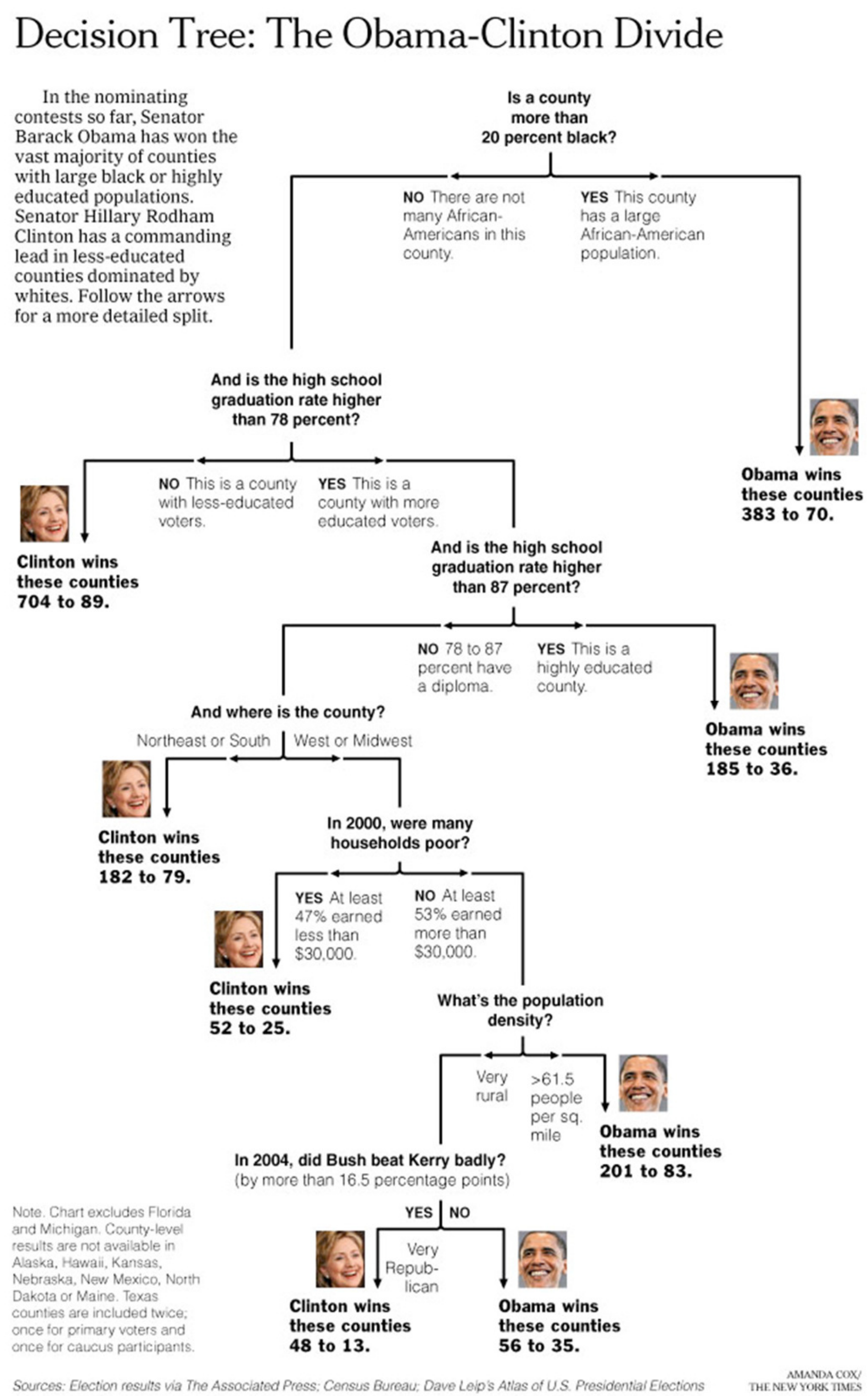
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural
for humans.
Style of many "How to"
manuals and trouble-shooting
procedures.



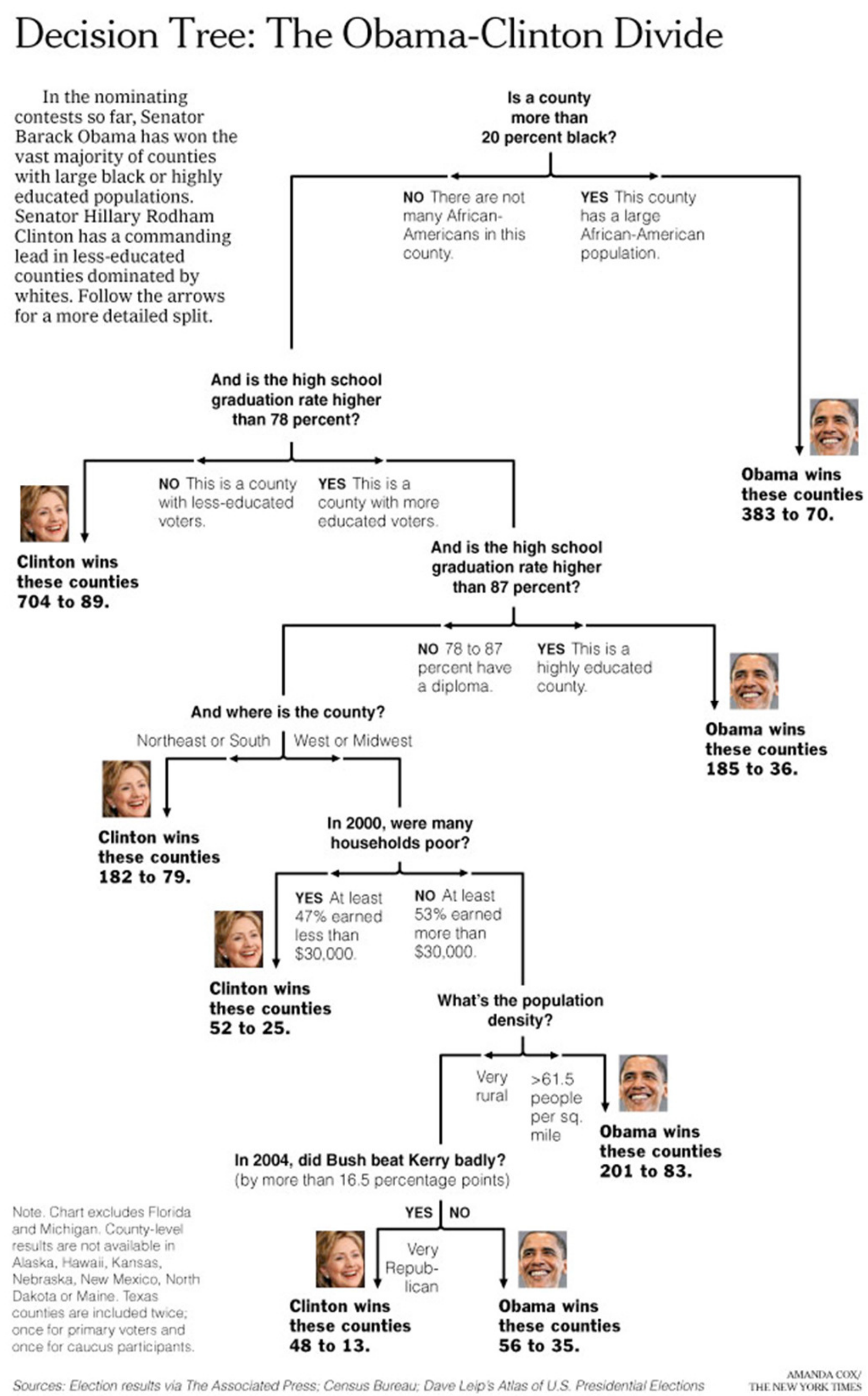
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



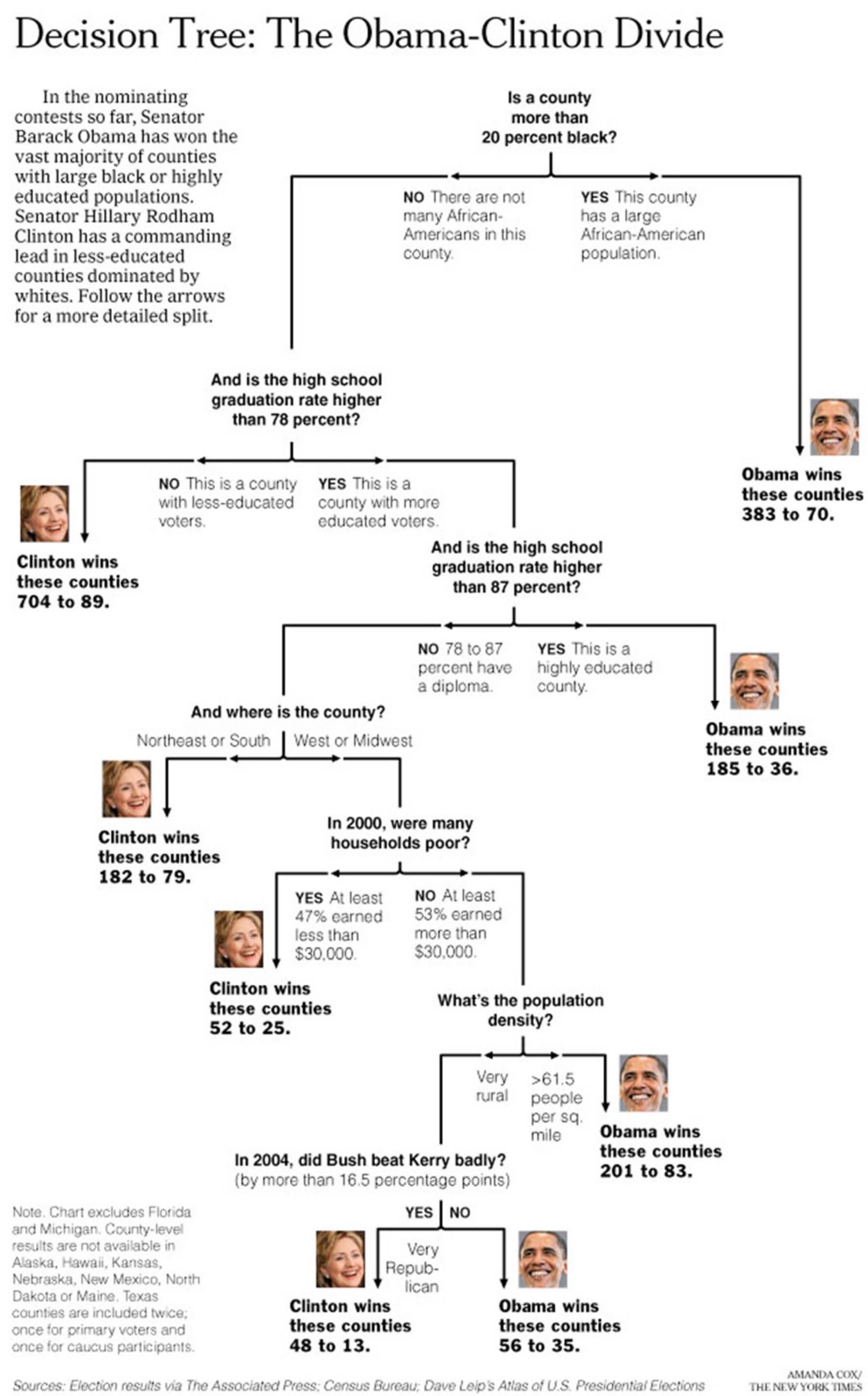
New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural
for humans.
Style of many "How to"
manuals and trouble-shooting
procedures.



New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

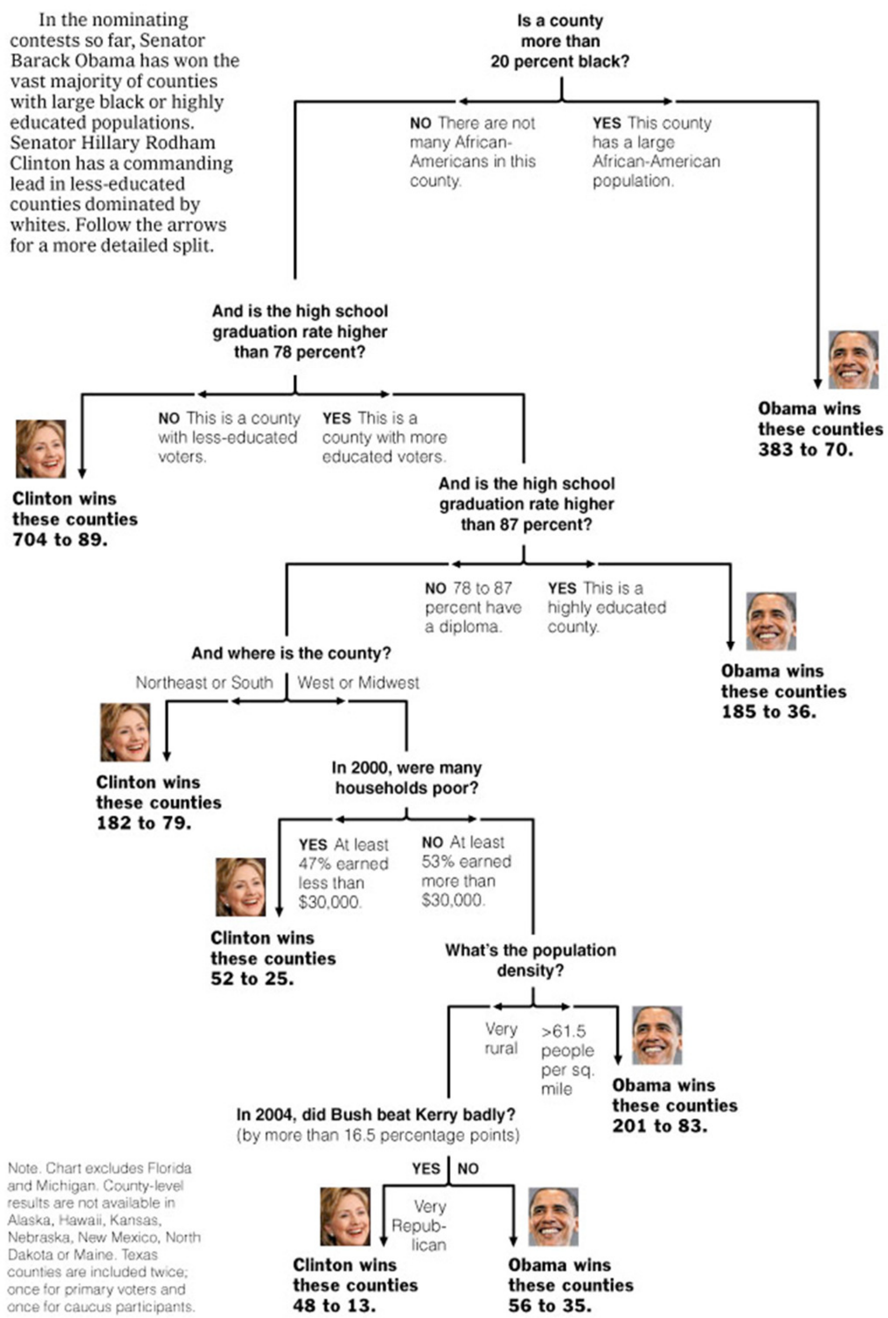


New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

Decision Tree: The Obama-Clinton Divide

In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites. Follow the arrows for a more detailed split.

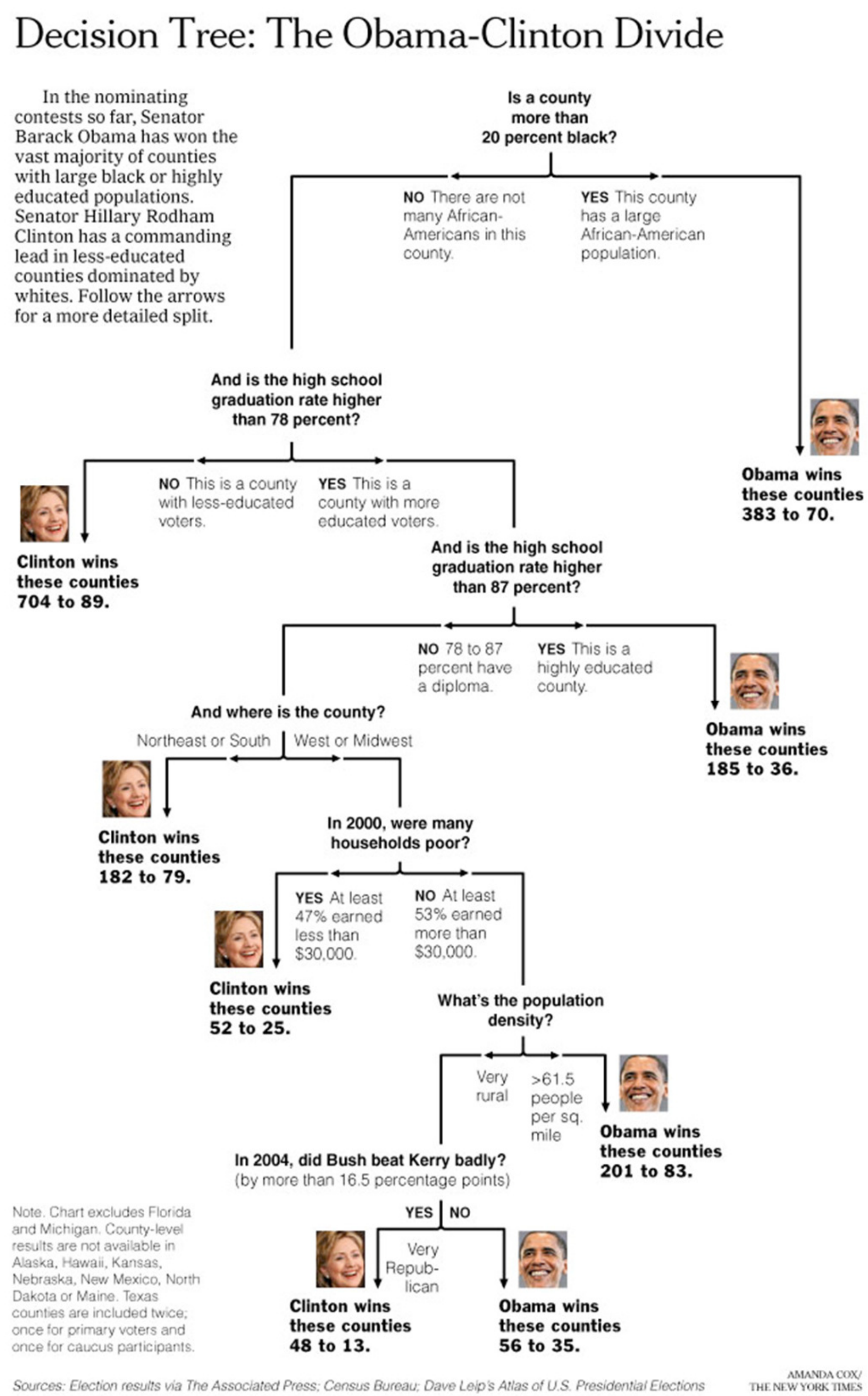


Note. Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Texas counties are included twice; once for primary voters and once for caucus participants.

Sources: Election results via The Associated Press; Census Bureau; Dave Leip's Atlas of U.S. Presidential Elections

AMANDA COX/ THE NEW YORK TIMES

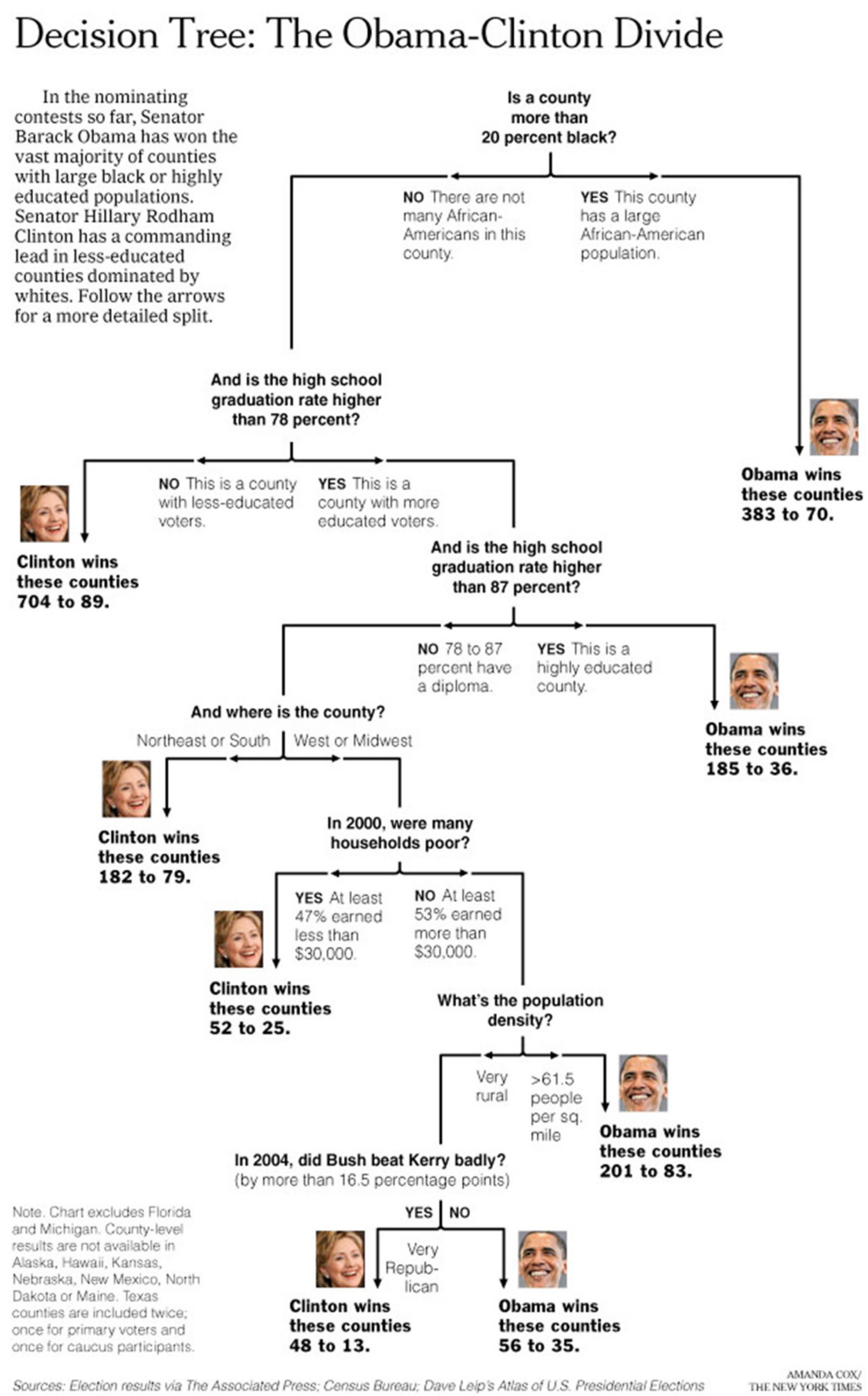
New York Times
April 16, 2008



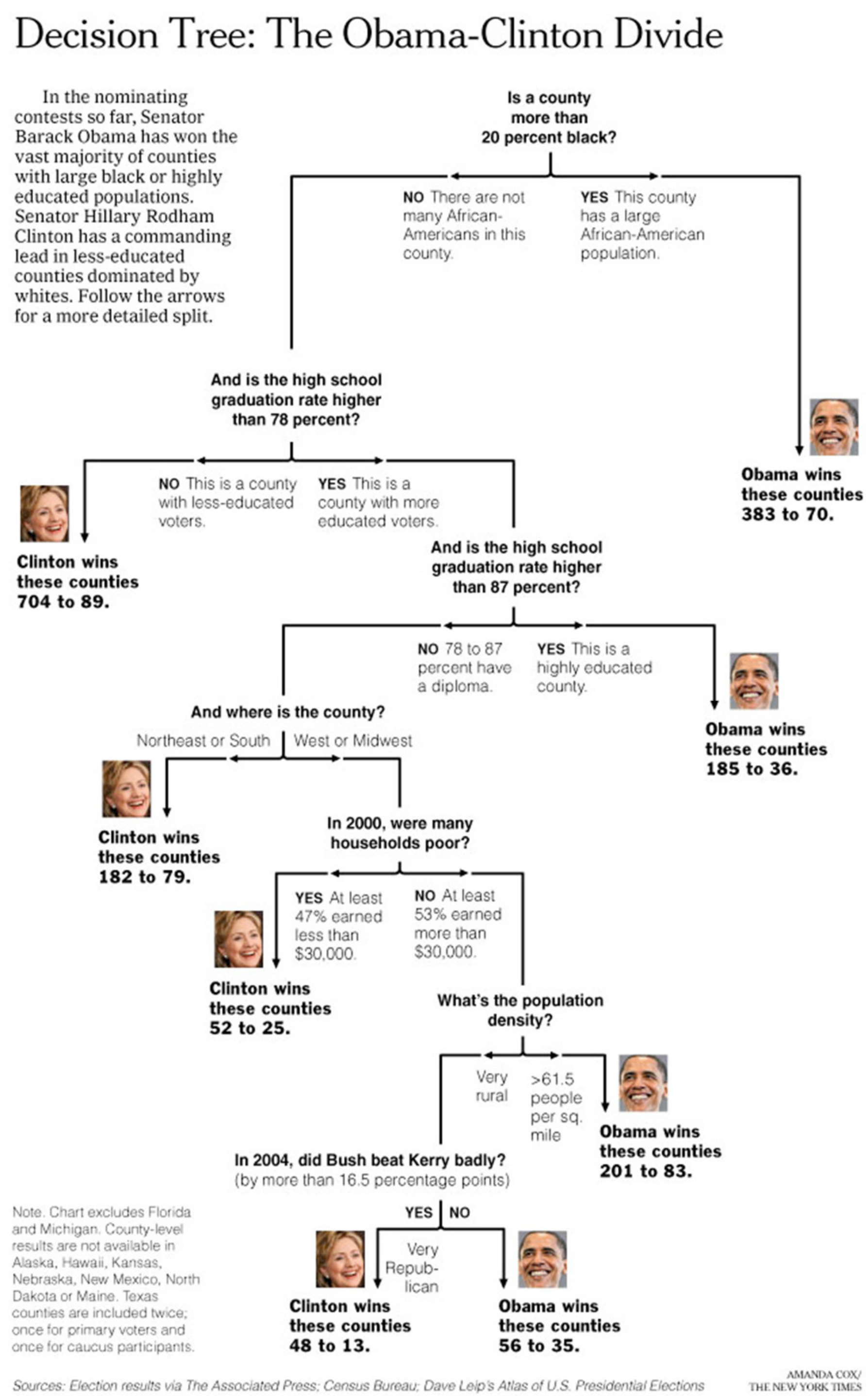
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



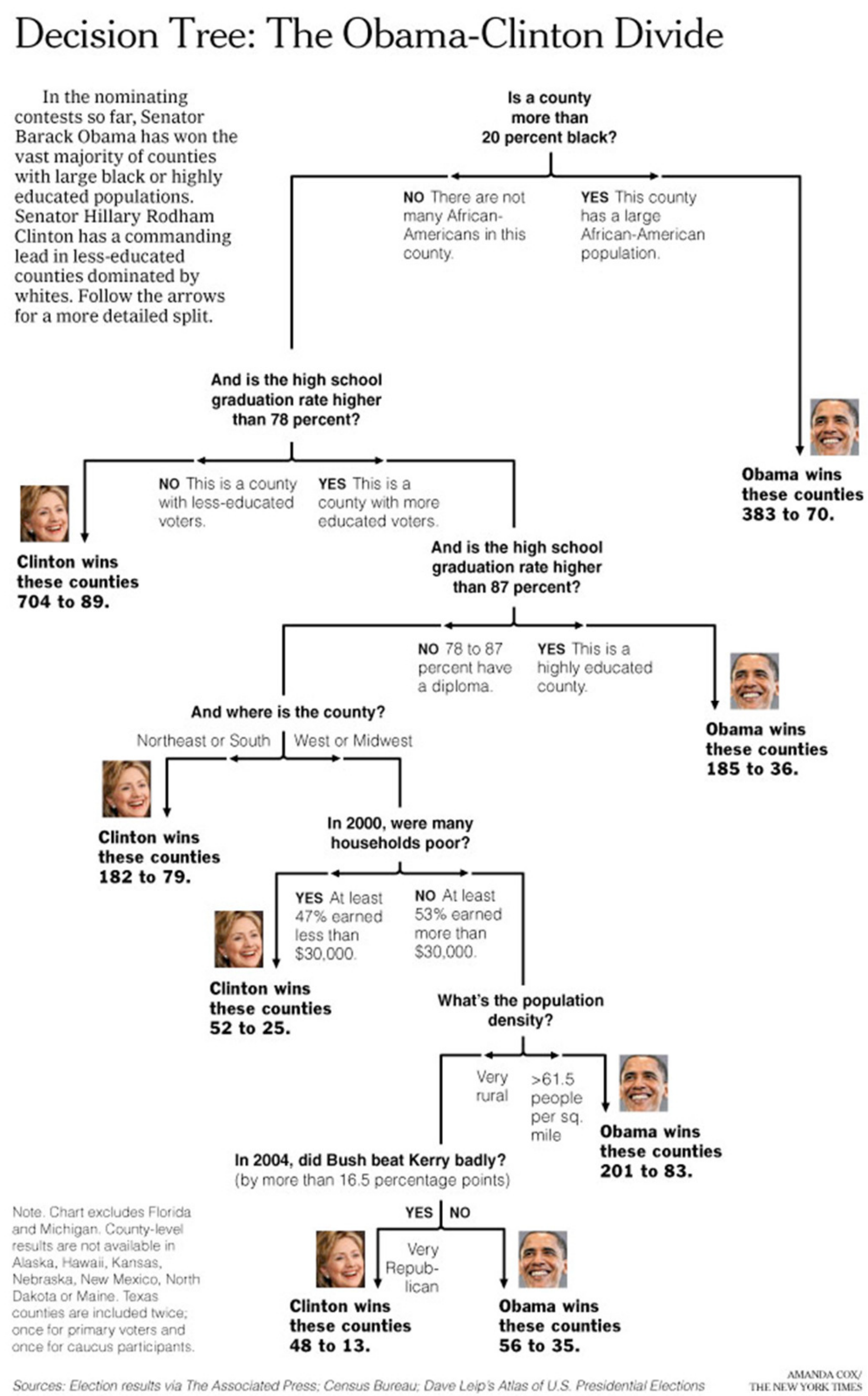
New York Times
April 16, 2008



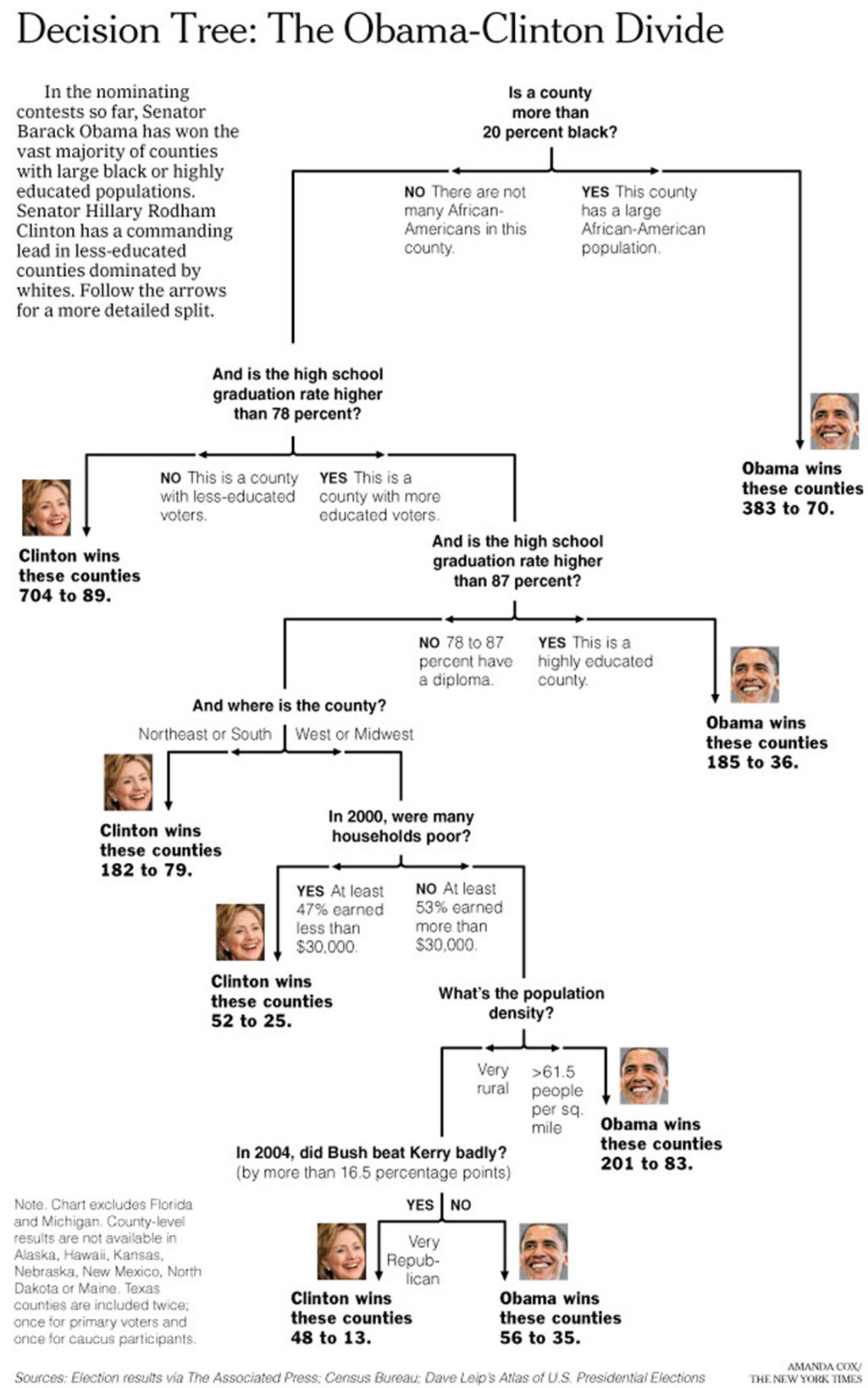
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

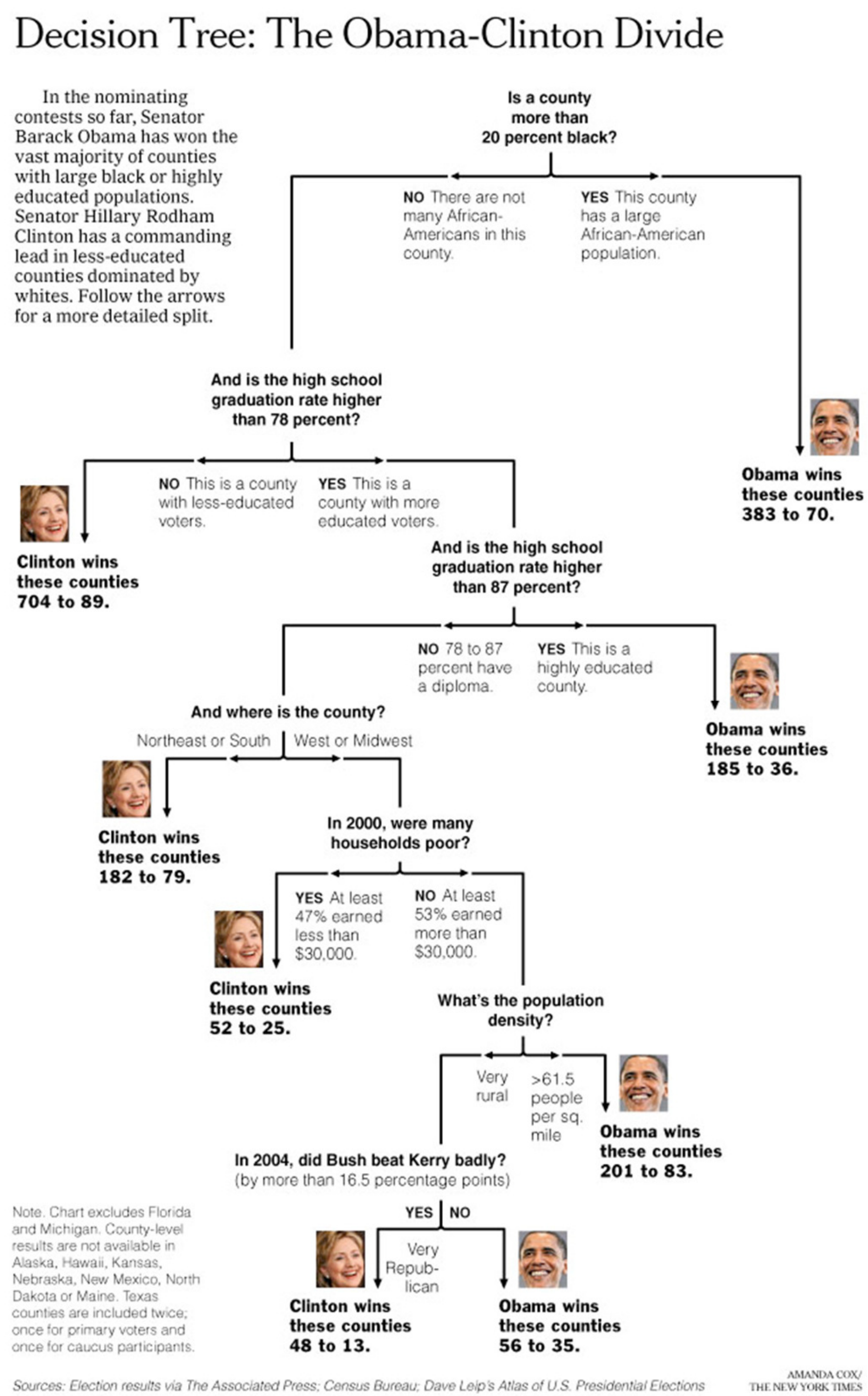


New York Times
April 16, 2008



Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

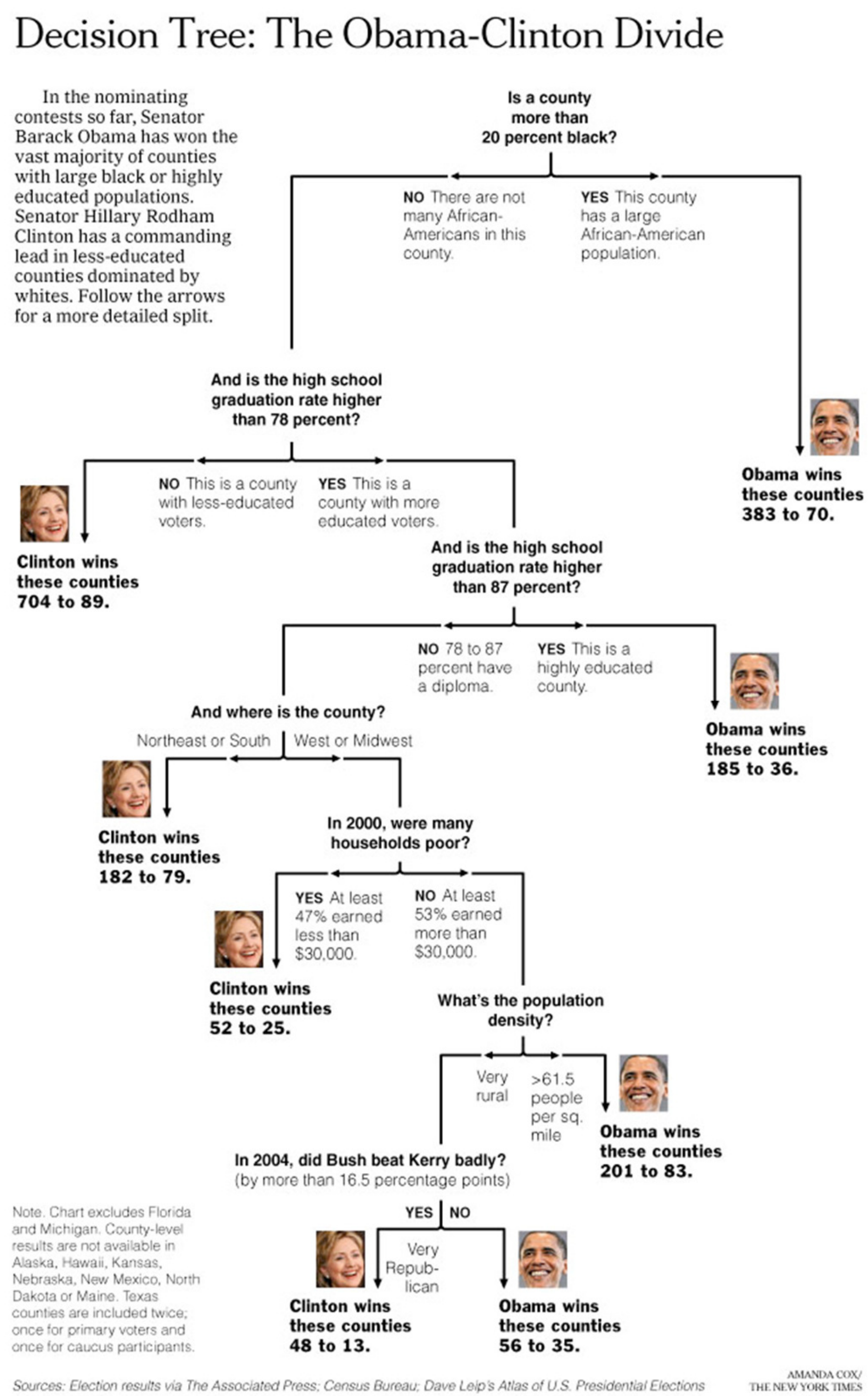
New York Times
April 16, 2008



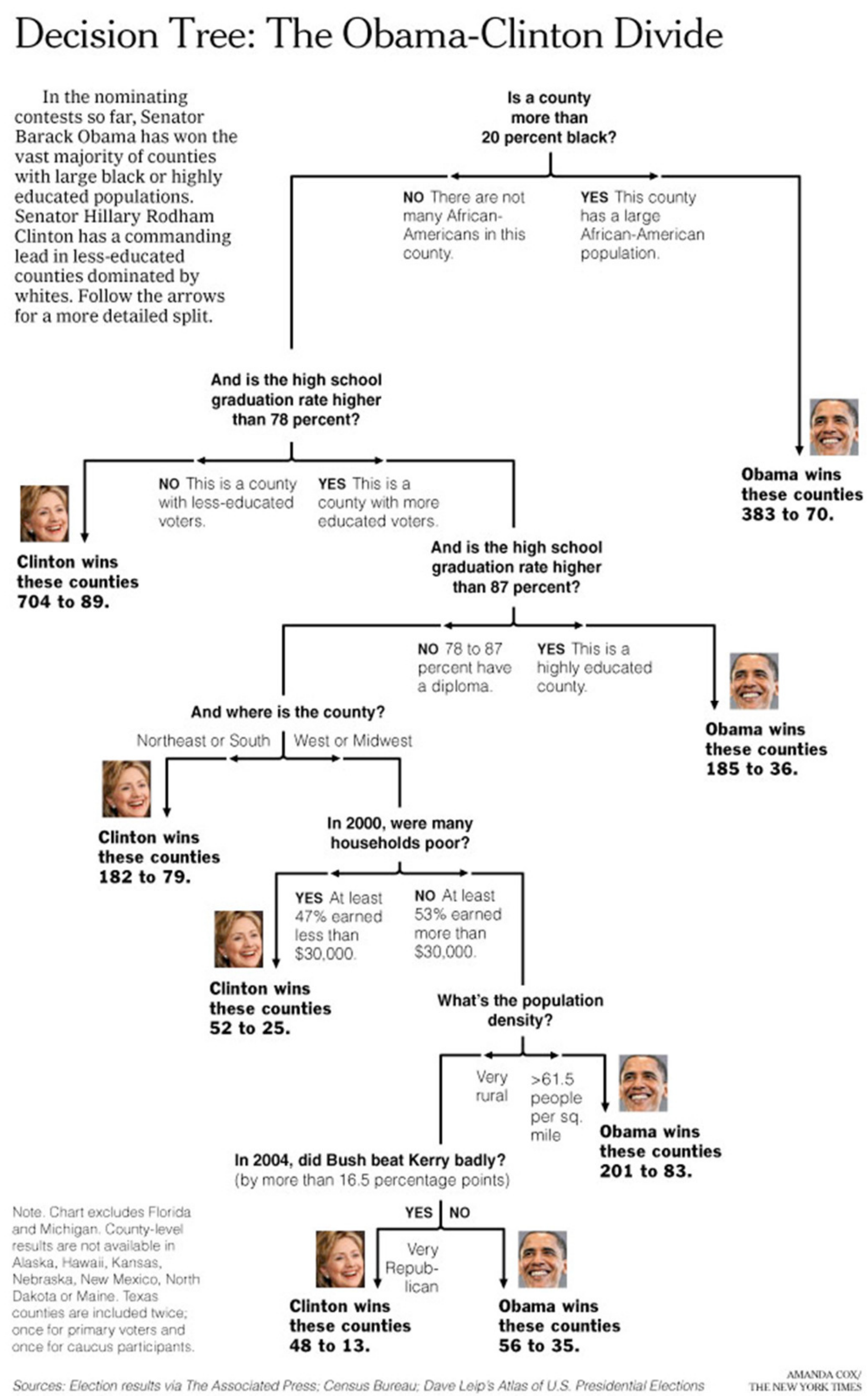
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008

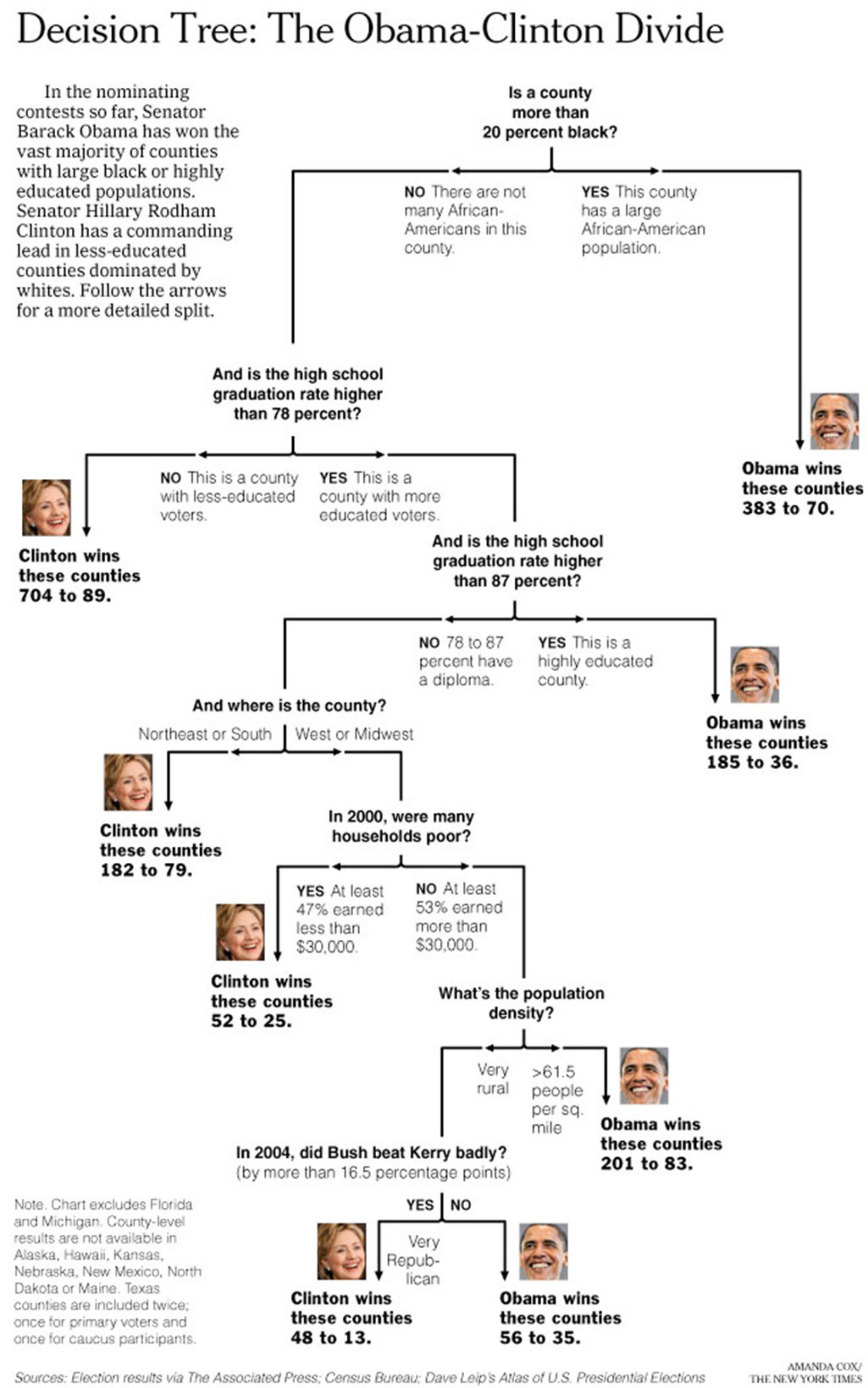
Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



New York Times
April 16, 2008



Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.



New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.

New York Times
April 16, 2008

Decision Trees:
a sequence of tests.
Representation very natural for humans.
Style of many "How to" manuals and trouble-shooting procedures.