Лекция 7:
Задача классификации и деревья решений

Задачи классификации

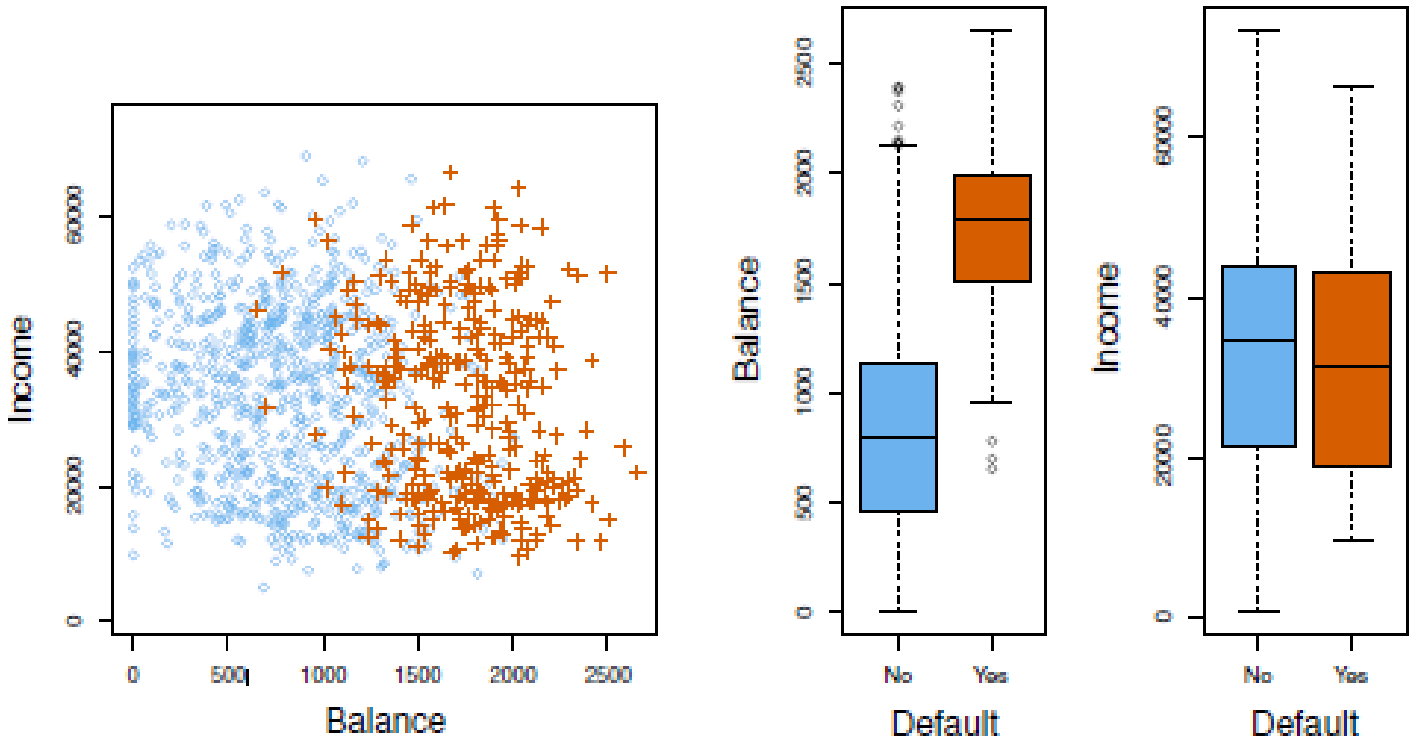
Переменная отклика *Y* является *категориальной* – например, почтовое сообщение может принадлежать одному из следующих классов *C* = (spam; ham) (ham = обычная почта), изображение цифры может принадлежать одному из классов *C = {0, 1, …, 9}.*

Цель задачи:

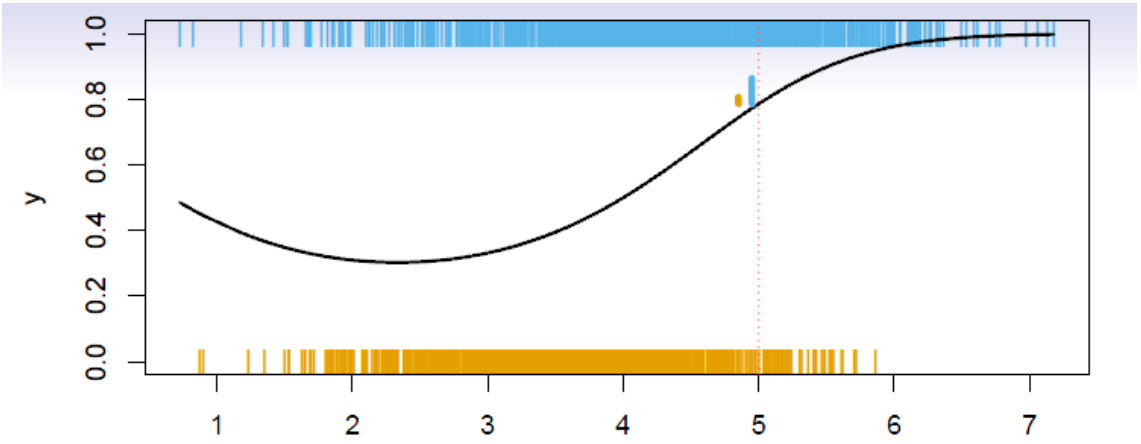
- Построить классификатор *C(X)*, который связывает метку класса из множества *C* с неразмеченным образцом *X*.
- Оценка степени неточности для каждой классификации
- Понимание роли различных предикторов среди

$$X = (X_1, X_2, \ldots, X_p).$$

Пример: Банкротство по кредитным картам



Задача классификации



Существует ли идеальная *C(X)*? Пусть K классов в множестве C пронумерованы как 1, 2,…, K. Пусть

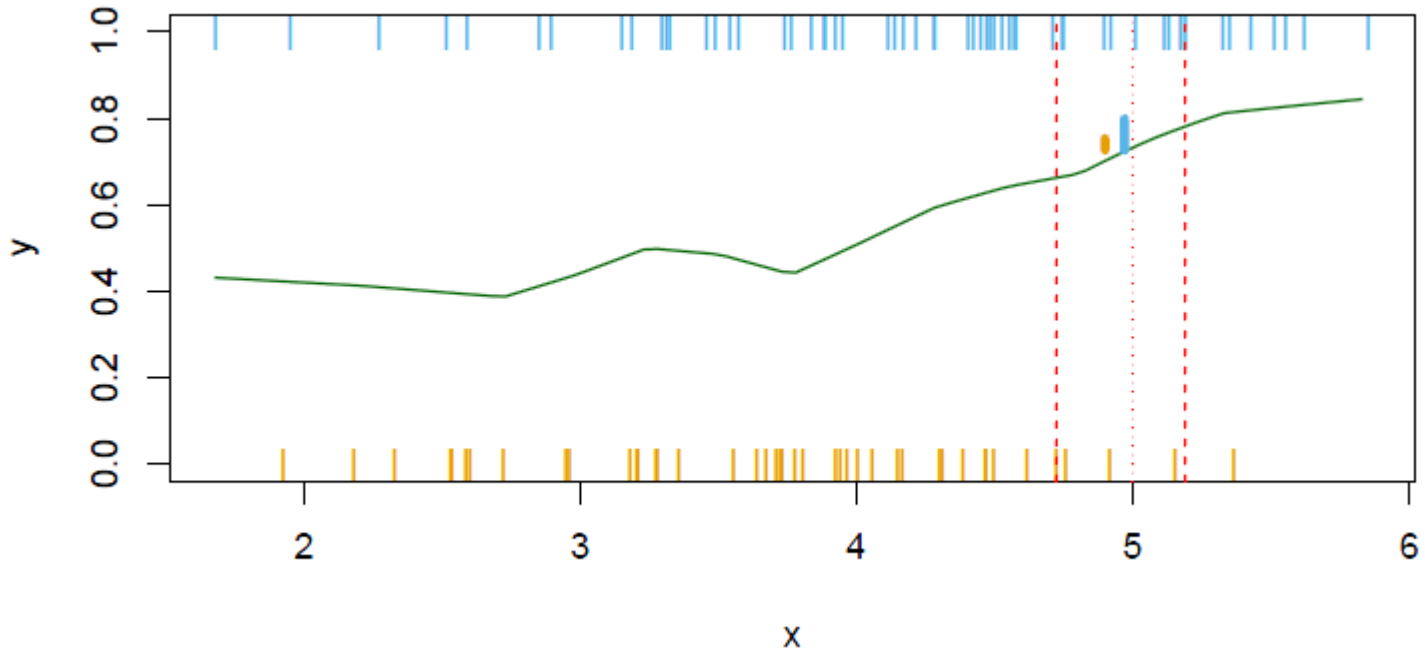
$$p_k(x) = \Pr(Y = k|X = x),\ k = 1, 2, \ldots, K.$$

Это условные вероятности классов в точке *x*. Тогда оптимальный классификатор Байеса в точке *x*:

$$C(x) = j \text{ if } p_j(x) = \max\{p_1(x), p_2(x), \ldots, p_K(x)\}$$

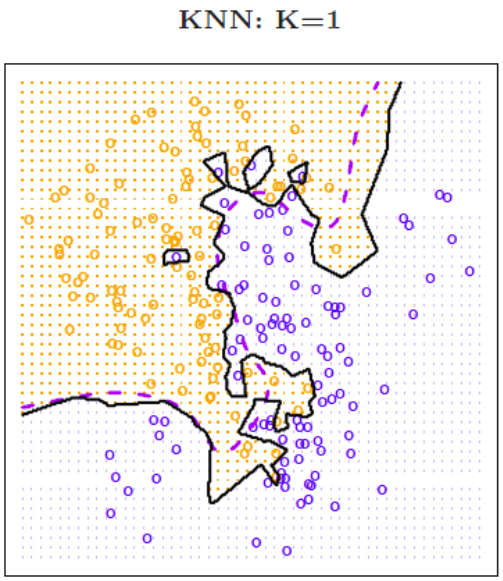
Качество классиффикации: $\mathrm{Err}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}} I[y_i \neq \hat{C}(x_i)]$

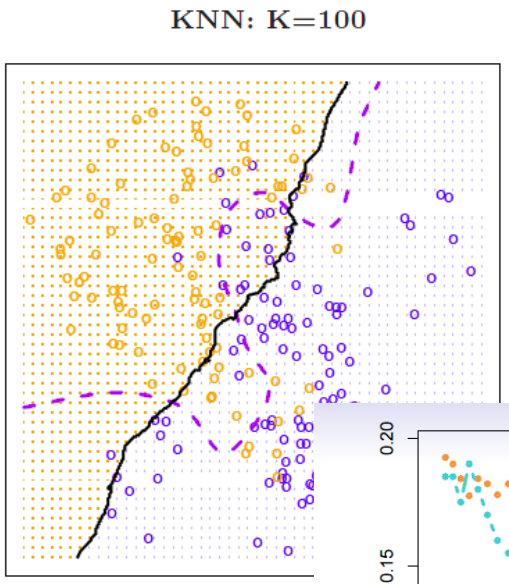
Задача классификации KNN

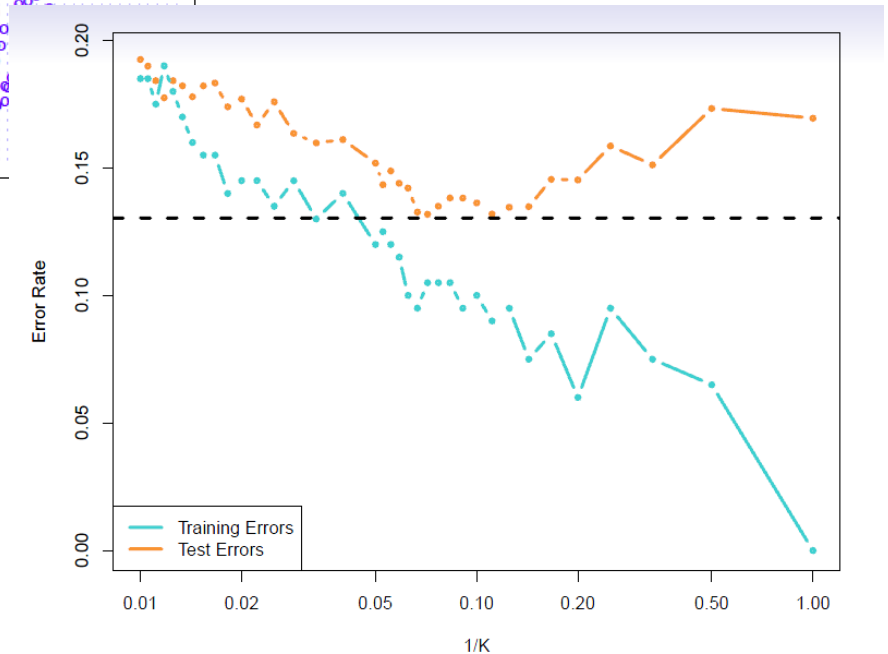


- Также может быть использовано усреднение методом ближайших соседей.
- Хотя такой подход плохо применим при возрастании размерности.

Пример: KNN для двух измерений







Методы Байеса

- Вероятностная постановка задачи прогнозирования:
  - В явном виде вычисляется вероятность для каждой гипотезы (варианта прогноза), очень важно для многих задач
- Возможность дообучения:
  - Каждый размеченный пример итеративно увеличивает/уменьшает вероятность корректности гипотезы
  - Априорные знания легко внедрить в модель
- Вероятностный прогноз:
  - Несколько вариантов гипотез с оценкой достоверности
- Де факто - эталон:
  - Даже когда методы Байеса вычислительно трудоемки, они могут быть полезны в качестве разовой оценки для сравнения с более быстрыми методами в плане оценки точности последних

Теорема Байеса

- Теореме Байеса:
  - Если Z - тренировочный набор, то апостериорная вероятность гипотезы *h, P(h|Z)* удовлетворяет соотношению:

$$P(h|Z) = \frac{P(Z|h)P(h)}{P(Z)}$$

- MAP (maximum posteriori) гипотеза

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|Z) = \arg\max_{h \in H} P(Z|h)P(h).$$

- «Необходимые» оценки вероятностей:
  - P(Z) – константа для всех классов
  - P(h) - относительная частота класса в выборке
  - Надо найти h такое, что P(h|Z) максимально = h такое, что P(Z|h)·P(h) максимально
- Проблема:
  - вычисление P(Z|h) на всех комбинациях переменных вычислительно сложно или невозможно

Метод Naïve Байеса

- Naïve упрощение:
  - Атрибуты независимы: $P(x_1,\ldots,x_n|C) = P(x_1|C)\cdot\ldots\cdot P(x_n|C)$
- Оценка $P(x_i|C)$:
  - Если атрибут категориальный, то относительная частота того, сколько примеров с классом C имеют значение атрибута x_i
  - Если атрибут числовой, то по плотности распределения, например:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

    где μ_k – это среднее, и σ_k^2 - дисперсия (в классе *k*).
  - И так, и так считается быстро
- Предположение о независимости
  - Делает метод вычислительно эффективным
  - Приводит к оптимальному классификатору (если и правда независимые атрибуты), но на практике почти всегда зависимы
- Попытки преодолеть эти ограничения:
  - Байесовские сети

Пример: построить модель прогноза возможности поиграть в теннис

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(p) = 9/14
P(n) = 5/14

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

Пример

- Новый пример:
  - x = <rain, hot, high, false>
- Расчет вероятностей:
  - $P(x|p)\cdot P(p) = P(rain|p)\cdot P(hot|p)\cdot P(high|p)\cdot P(false|p)\cdot P(p) = 3/9\cdot 2/9\cdot 3/9\cdot 6/9\cdot 9/14 = 0.010582$
  - $P(x|n)\cdot P(n) = P(rain|n)\cdot P(hot|n)\cdot P(high|n)\cdot P(false|n)\cdot P(n) = 2/5\cdot 2/5\cdot 4/5\cdot 2/5\cdot 5/14 = 0.018286$
- Вывод:
  - $P(x|n)\cdot P(n) > P(x|p)\cdot P(p)$
  - Значит скорее всего не поиграть

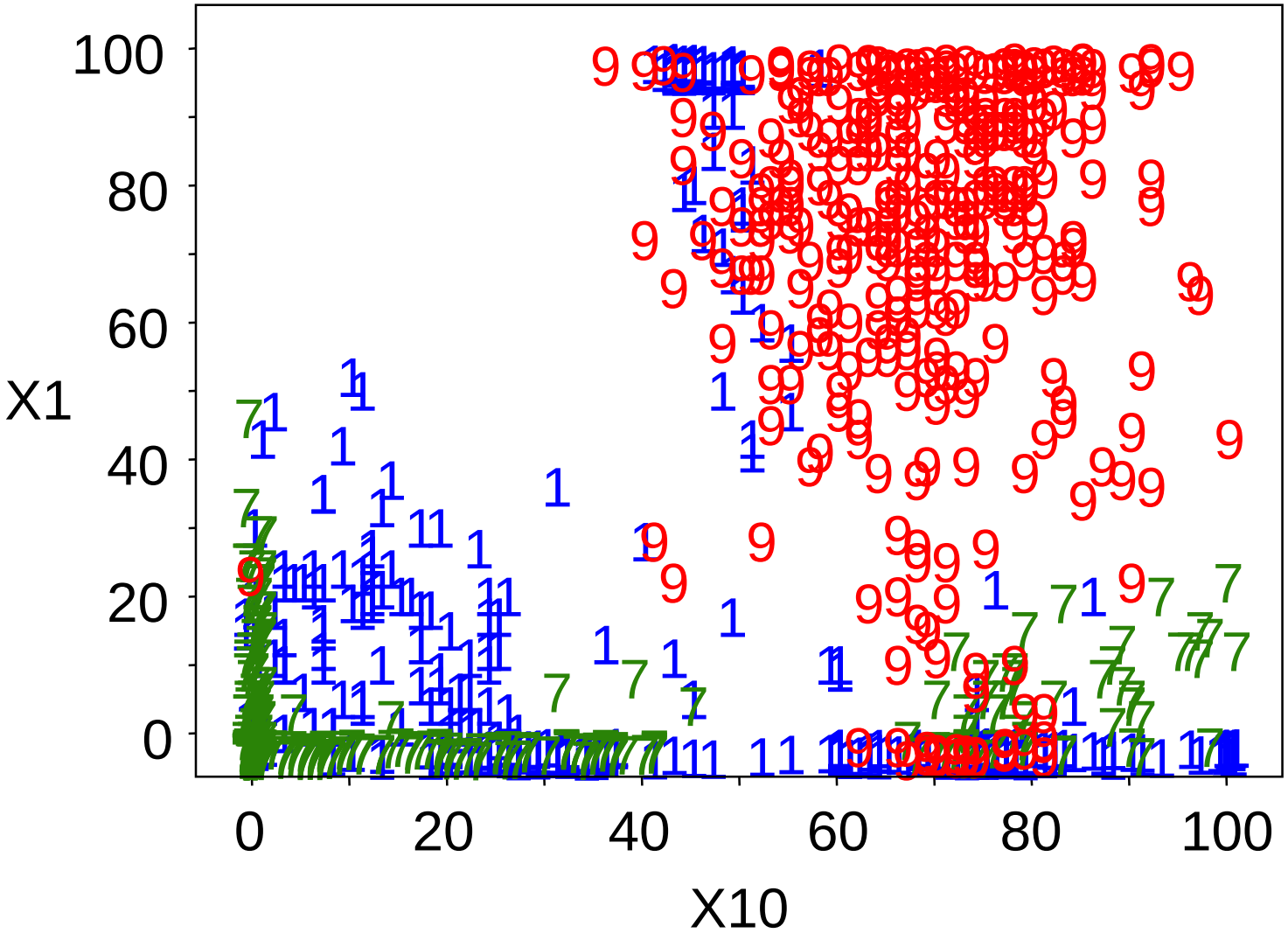
Методы, основанные на деревьях решений

- Эти методы используют *стратификацию или сегментирование* пространства признаков на области.
- Для сегментации пространства признаков может использоваться набор правил, который можно представить в виде дерева
- Деревья решений могут применяться как к задачам регрессии, так и к классификации.
- Методы, основанные на деревьях, просты в интерпретации, при этом показывают достаточно хорошие результаты по точности прогнозирования.
- Нестабильные модели – это плюс для ансамблей *бэггинг, методы случайного леса и бустинг*. Эти методы строят множество деревьев, результаты прогнозирования которых потом объединяются для получения итогового прогноза.

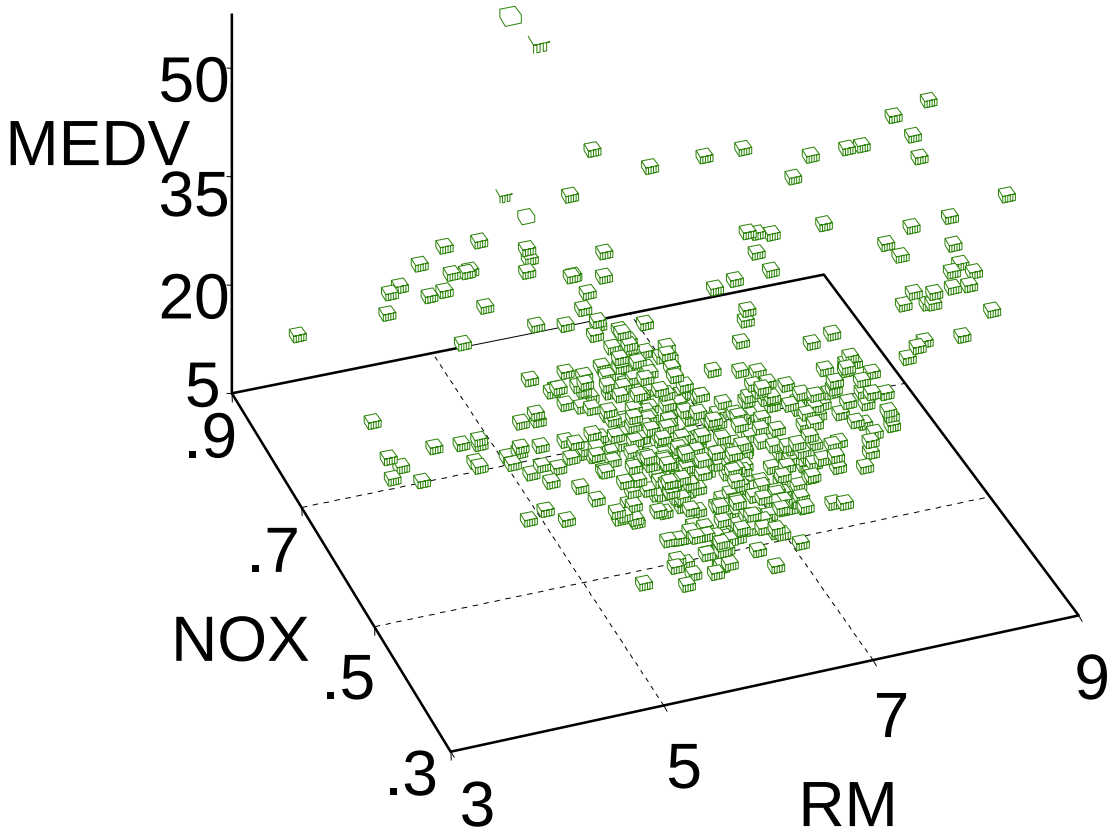
Деревья решений в задачах классификации и регрессии

- Дерево решений - граф (древовидная структура), в котором:
  - Внутренние узлы – условия на атрибуты
  - Каждая исходящая ветка соответствует выходному значению условия, ветка целиком – альтернативное решение
  - В листьях метки классов (или распределение меток классов) или значения целевой переменной для регрессии
  - Каждому узлу соответсвует область в пространсве признаков R
  - Области для листьев – финальные, не содержат внутри других областей
- Построение дерева обычно – 2 фазы
  - Построение: в начале в корне все примеры, далее рекурсивное разбиение множества примеров по выбранному атрибуту
  - «отсечение» ветвей pruning - выявление и удаление ветвей (решений), приводящих к шуму или к выбросам
- Применение дерева решений для нового объекта
  - Проверка атрибутов – путь по ветви до листа. В листе отклик.

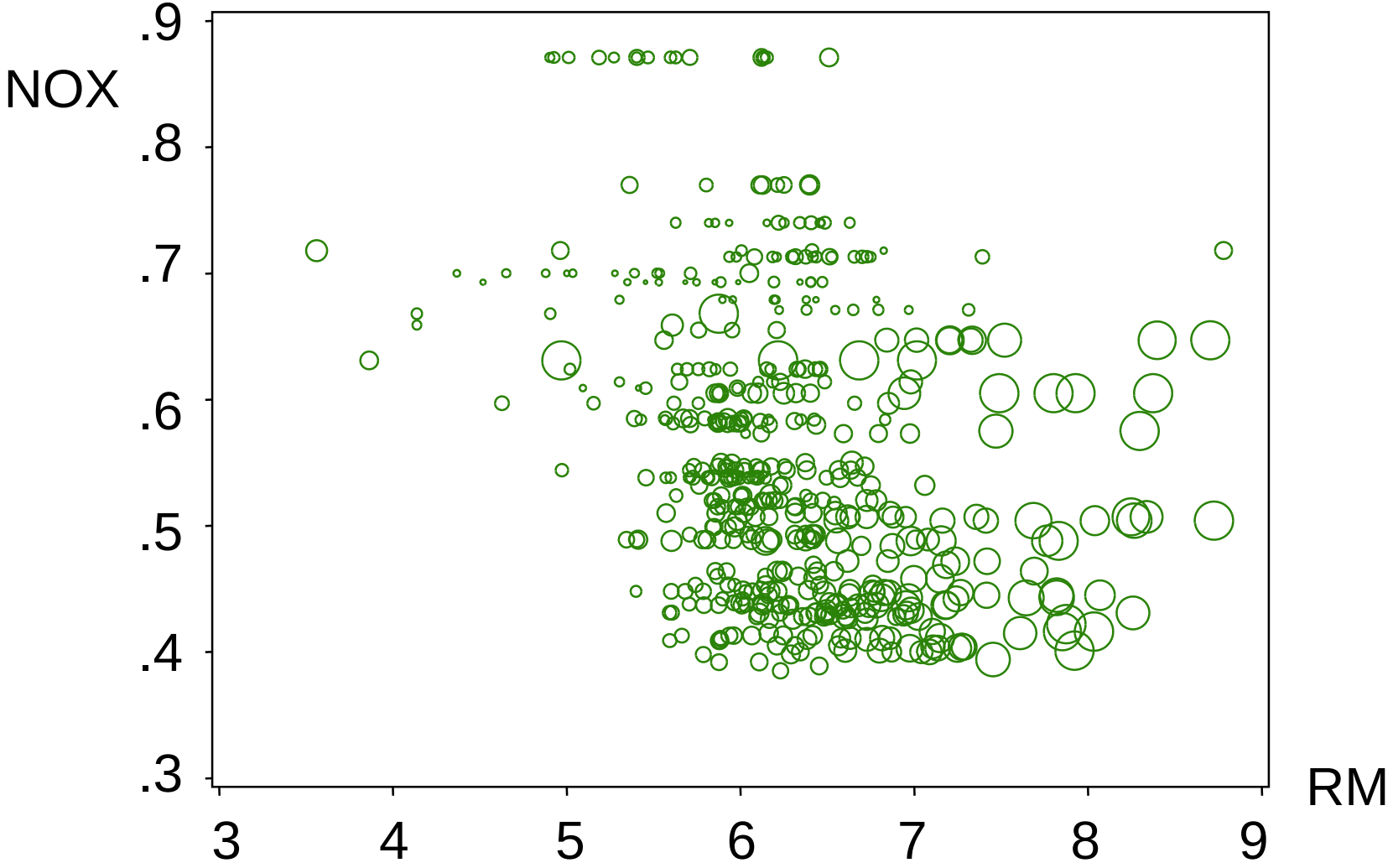
Пример: категориальный отклик



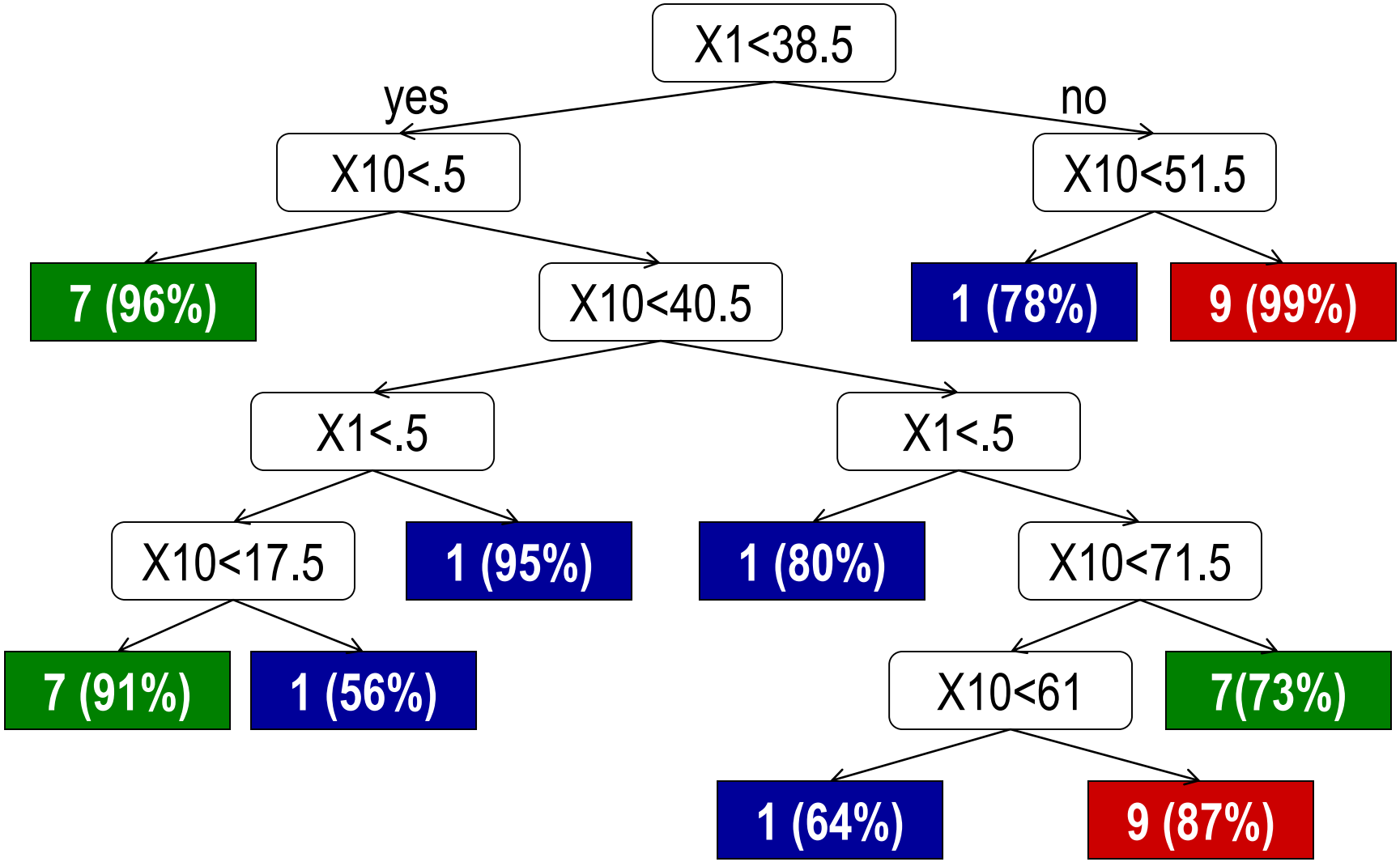
Пример: непрерывный отклик



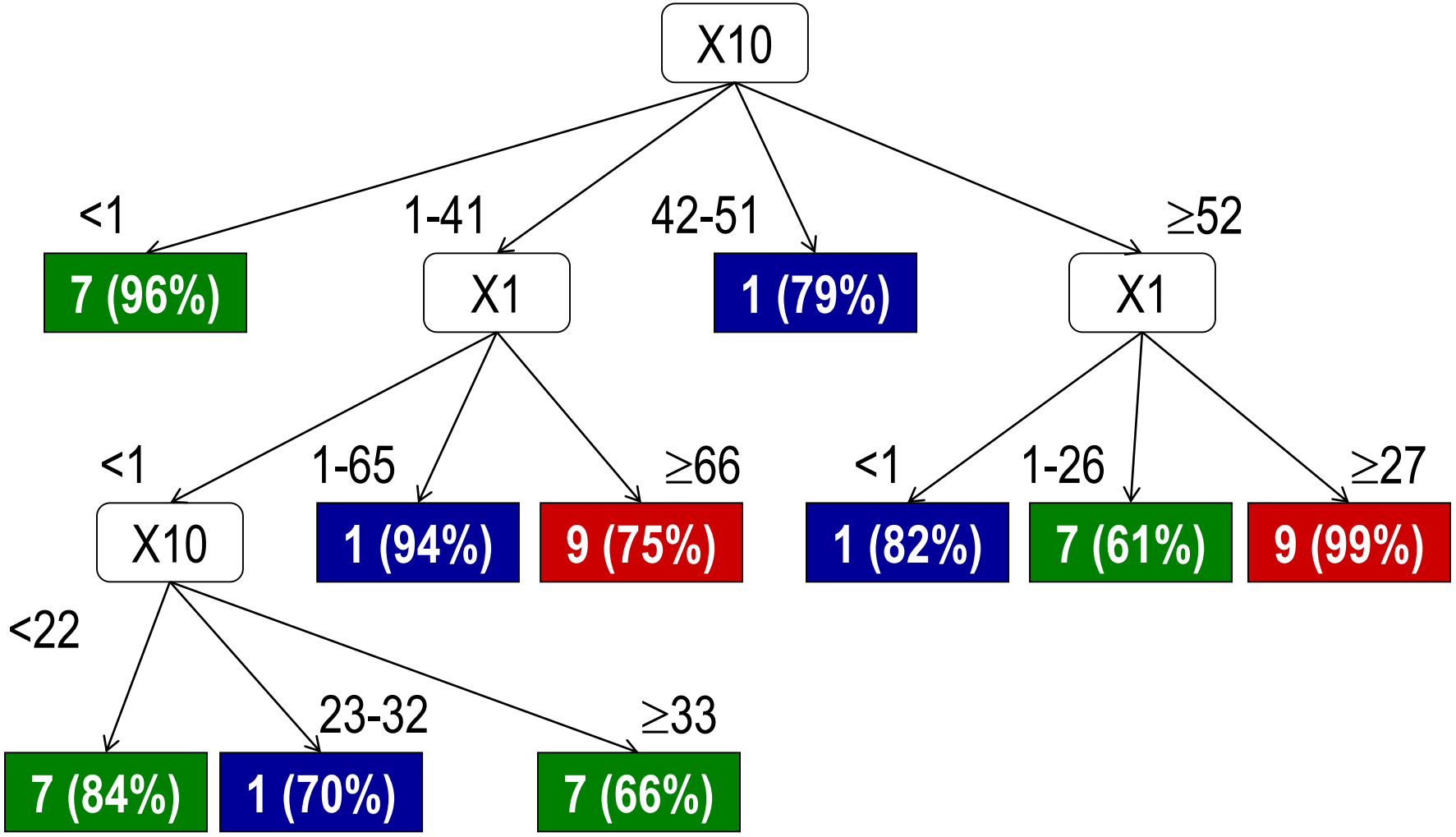
Проекция непрерывного отклика на пространство признаков



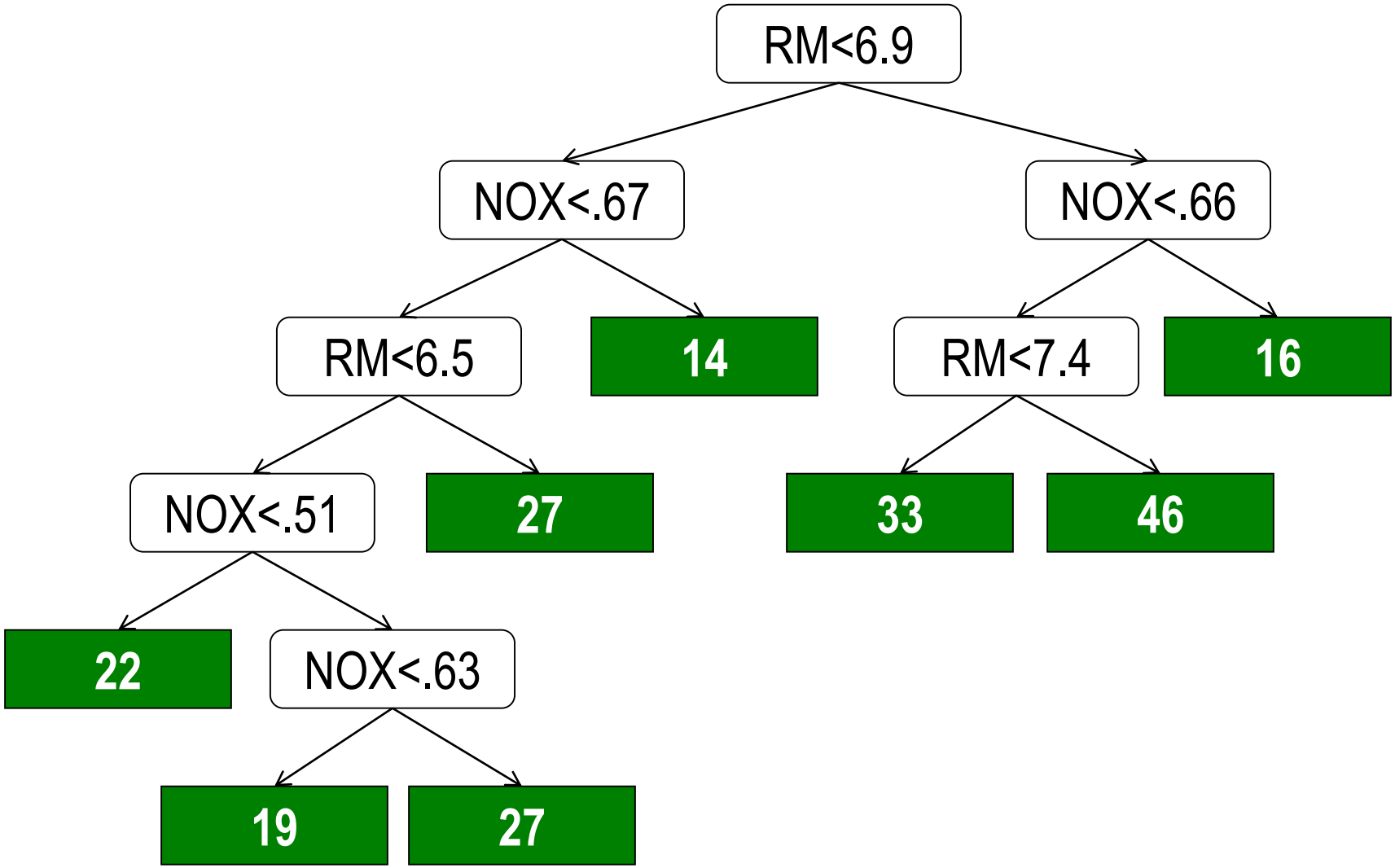
Дерево решений для классификации



Множественные разбияния (не бинарное дерево)



Дерево решений для регрессии

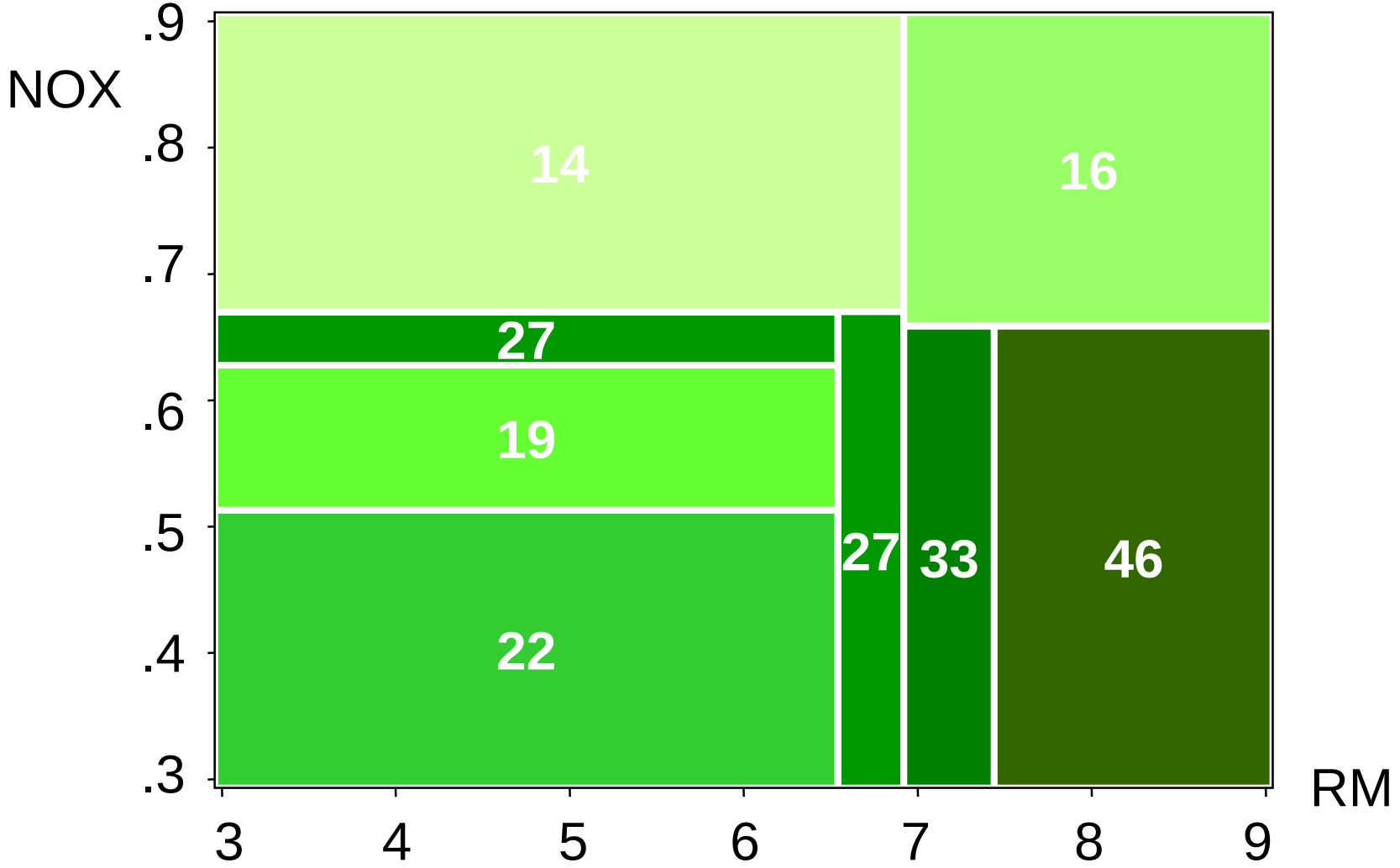


Листья = Логические правила

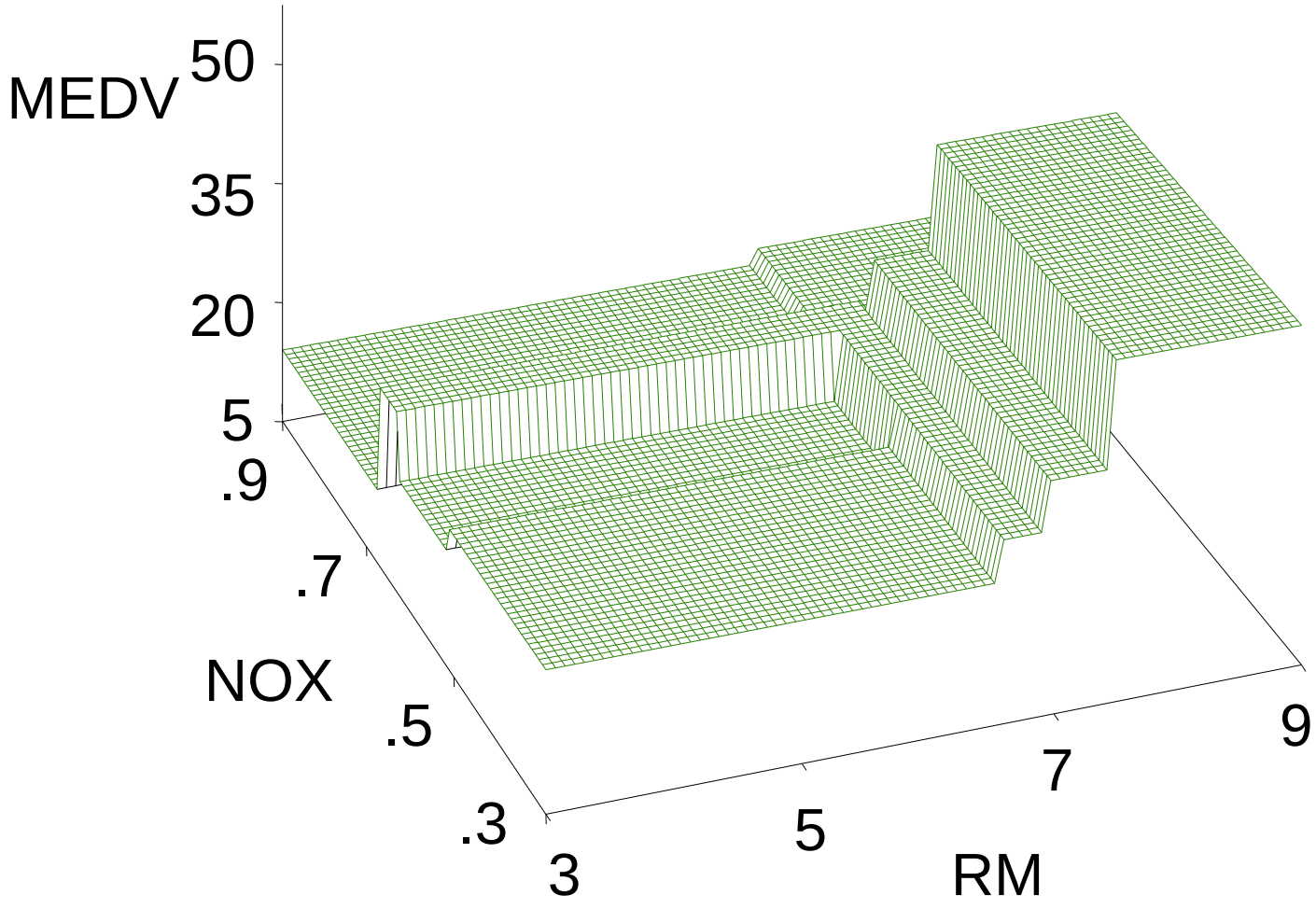
If RM $\in$ {*values*} and NOX $\in$ {*values*}, then MEDV=*value*.

Leaf	RM	NOX	Predicted MEDV
1	<6.5	<.51	22
2	<6.5	[.51, .63)	19
3	<6.5	[.63, .67)	27
4	[6.5, 6.9)	<.67	27
5	<6.9	≥.67	14
6	[6.9, 7.4)	<.66	33
7	≥7.4	<.66	46
8	≥6.9	≥.66	16

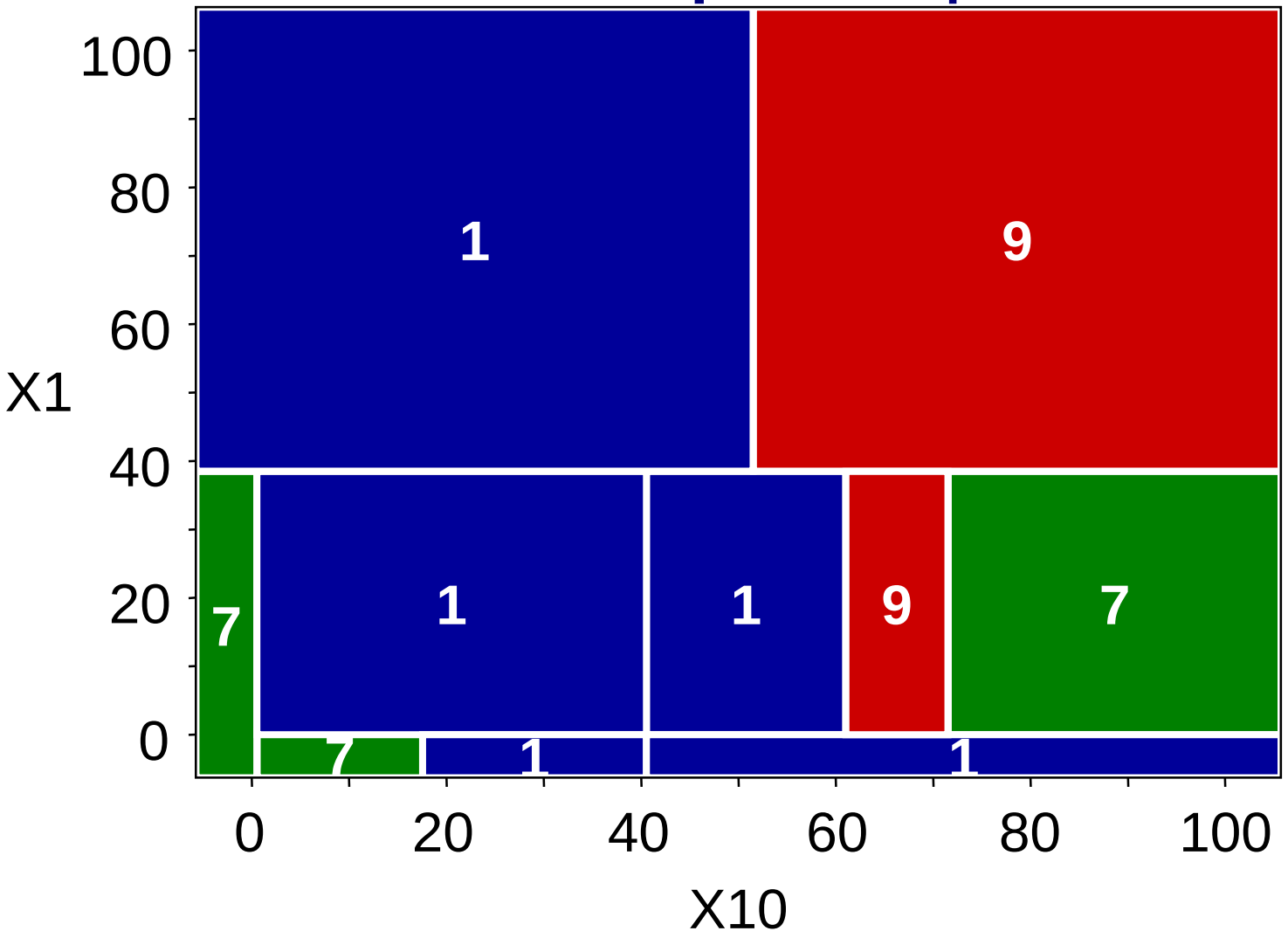
Разбиение пространства признаков на области



Многомерная ступенчатая функция



Регионы решений для классификации



Листья дерева решений для классификации

Leaf	Pr(1|**x**)	Pr(7|**x**)	Pr(9|**x**)	Decision
1	.03	.96	.01	**7**
2	.09	.91	.00	**7**
3	.56	.44	.00	**1**
4	.95	.05	.00	**1**
5	.80	.10	.10	**1**
6	.64	.09	.27	**1**
7	.00	.13	.87	**9**
8	.10	.73	.17	**7**
9	.78	.01	.21	**1**
10	.01	.00	.99	**9**

Процесс построения деревьев решений

- Разделяем пространство признаков (то есть, набор возможных значений для $X_1, X_2,\ldots,X_p$) в J отдельных и непересекающихся областей $R_1,R_2,\ldots,R_J$. Таким образом, чтобы поведение отклика в каждой области было «однородным».
- Теоретически, области могут иметь любую форму. Тем не менее, чаще всего используются многомерные *прямоугольники* для простоты и удобства интерпретации полученной модели.
- Цель состоит в том, чтобы найти прямоугольные области $R_1, \ldots,R_J$, так чтобы минимизировать некий целевой критерий – *критерий разбиения*.

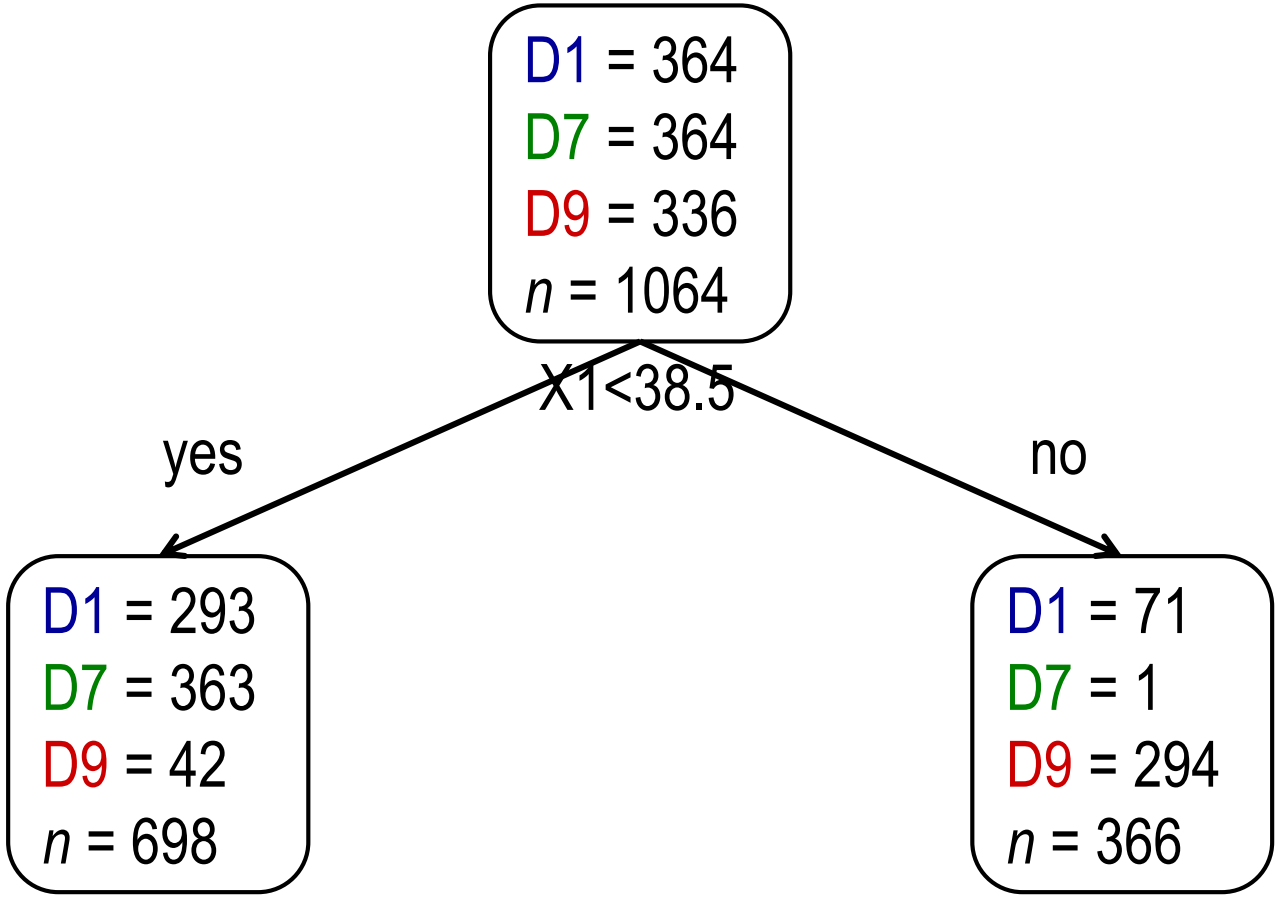
Процесс построения деревьев решений

- Вычислительно нецелесообразно рассматривать все возможные разбиения пространства признаков в *J* областей.
- По этой причине используется *нисходящий, жадный* подход, известный как рекурсивное разделение.
- Подход называется *нисходящий*, потому что он начинается в верхней части дерева, а затем последовательно разбивает пространство признаков; каждое разбиение приводит к образованию новых ветвей, расположенных ниже по дереву.
- Подход называется *жадным*, потому что на каждом этапе процесса построения деревьев *лучшее* разбиение осуществляется на конкретном шаге, вместо того, чтобы просматривать дальше и выбирать разбиение, которое приведет к лучшему дереву в дальнейшем.

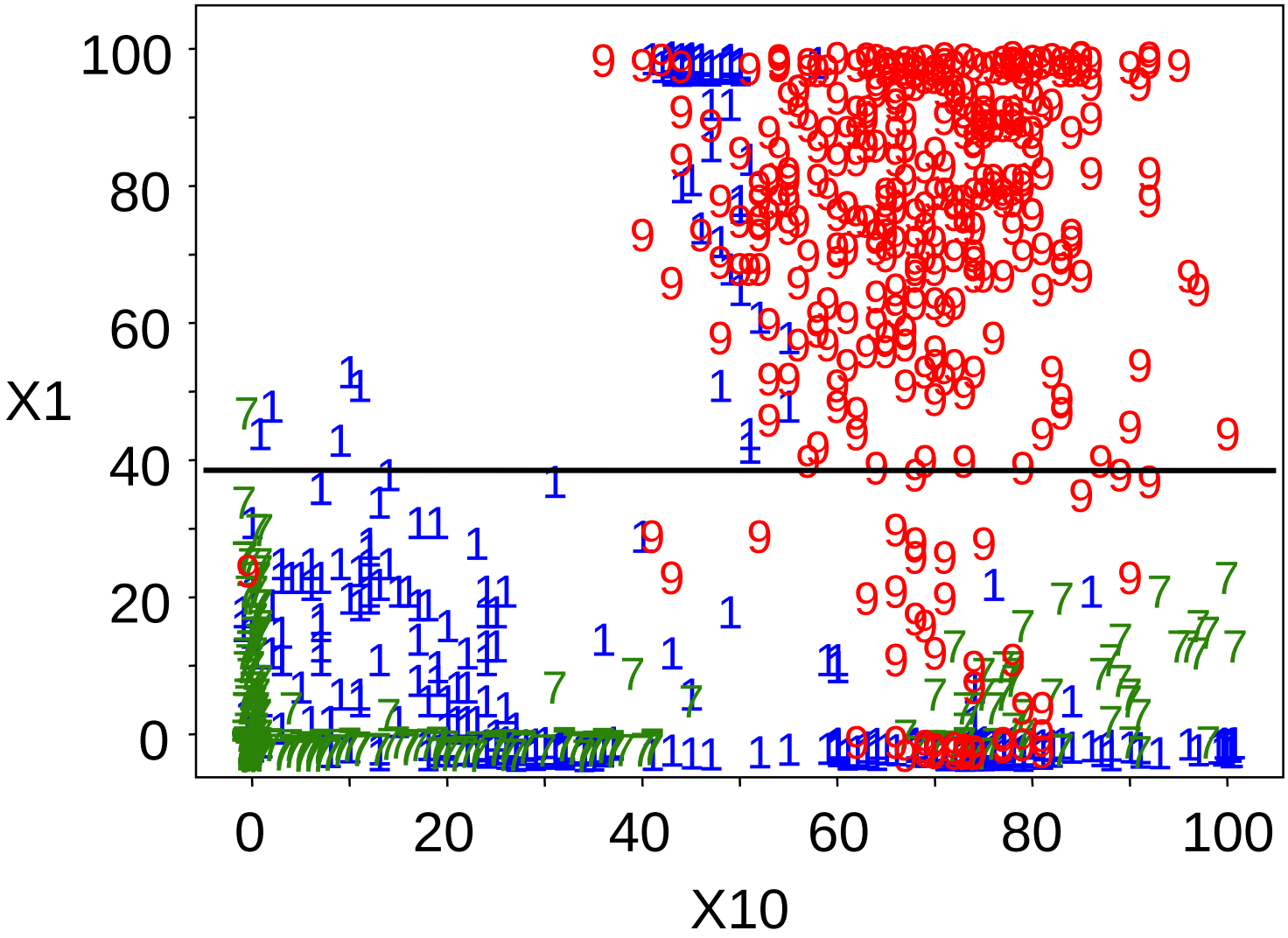
Процесс построения деревьев решений

- Сначала выбирается переменная X_j и точка разбиения s, так что разбиение пространства признаков на области $\{X|X_j < s\}$ и $\{X|X_j >= s\}$ приводит к максимально возможному улучшению критерия (для бинарного дерева).
- Затем повторяется этот процесс и ищется лучшая переменная и точка разбиения в каждой из результирующих областей.
- Однако на этот раз, вместо разделения всего пространства признаков, мы разделяем одну из ранее выделенных областей.
- Процесс продолжается до тех пор, пока не будет достигнут критерий останова; например, мы можем продолжать процесс до тех пор, пока не останется областей, содержащих более пяти наблюдений.

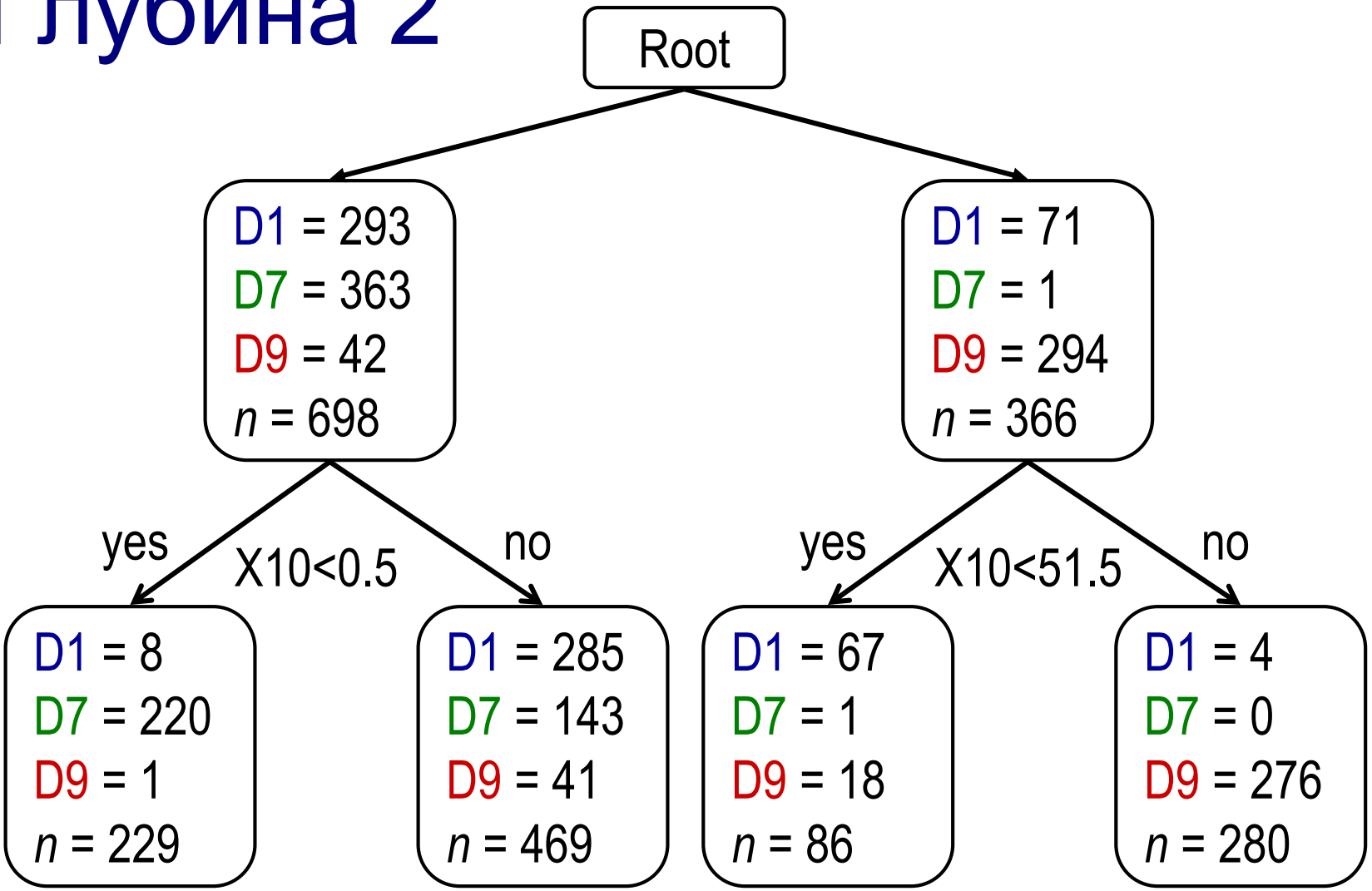
Шаг разбиения узла



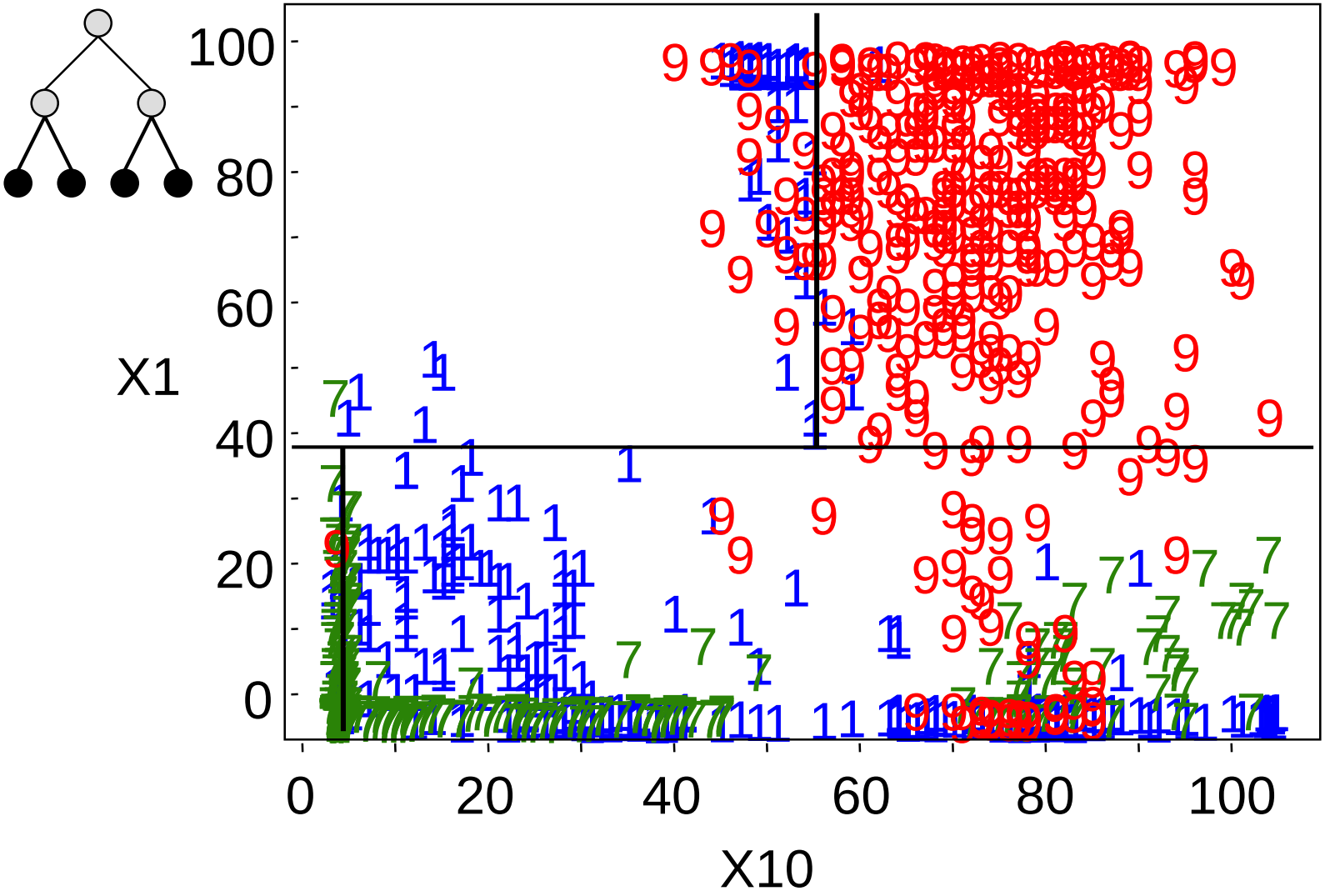
Разбиение деревом глубины 1



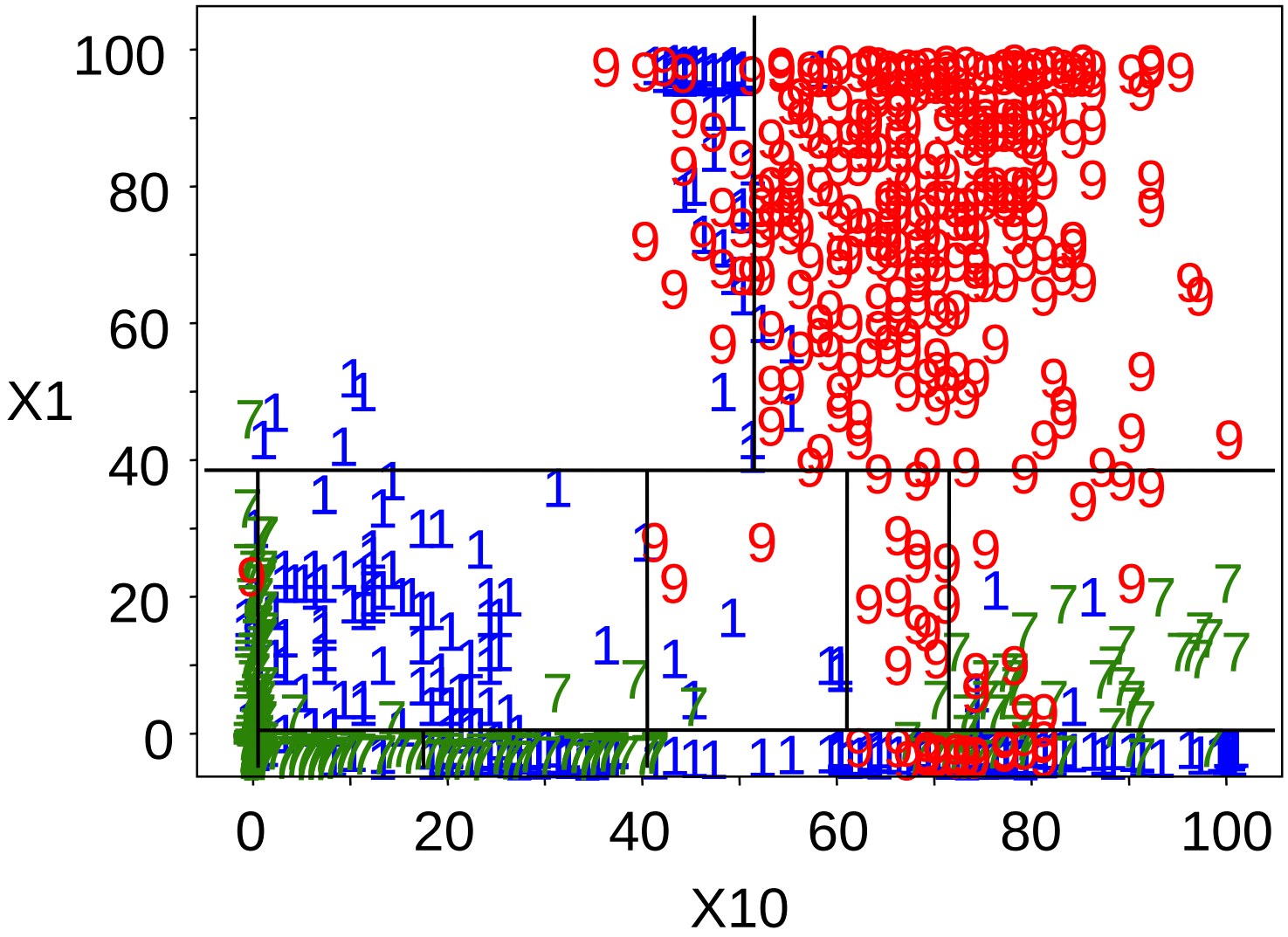
Глубина 2


Root
D1 = 293
D7 = 363
D9 = 42
n = 698
D1 = 71
D7 = 1
D9 = 294
n = 366
yes
X10<0.5
no
yes
X10<51.5
no
D1 = 8
D7 = 220
D9 = 1
n = 229
D1 = 285
D7 = 143
D9 = 41
n = 469
D1 = 67
D7 = 1
D9 = 18
n = 86
D1 = 4
D7 = 0
D9 = 276
n = 280

Разбиение деревом глубины 2

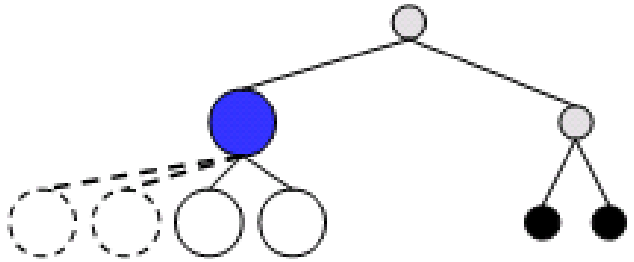


Финальное разбиение



Рассмотрим

- Поиск разбиения по переменным
  - Ординальным
  - Категориальным
- Множественные разбиения
- Критерии разбиения
  - Уменьшение разнородности
  - Хи2 тест
- Регрессионные деревья
- Пропущенные значения



Variable	Values
X10	0.5
X10	1.8
X10	11, 46
X1	2.4
X1	1, 4, 61
⋮	⋮

Разбиение по ординальной переменной

Splits

1—234
12—34
123—4

$$\binom{3}{1} = 3$$

$$\binom{L-1}{B-1} = \frac{(L-1)!}{(B-1)!\,(L-B)!}$$

1—2—34
1—23—4
12—3—4

$$\binom{3}{2} = 3$$

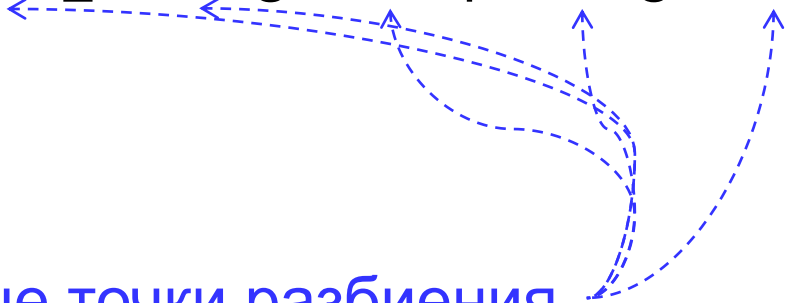
$$\sum_{l=2}^{L} \binom{L-1}{l-1} = 2^{L-1} - 1$$

1—2—3—4

$$\binom{3}{3} = 1$$

Ординальные переменные

X	.20	1.7	3.3	3.5	14	2515
ln(X)	–1.6	.53	1.2	1.3	2.6	7.8
rank(X)	1	2	3	4	5	6



Потенциальные точки разбиения

Разбиение категориальной переменной

1—234
2—134
3—124
4—123
12—34
13—24
14—23
1—2—34
1—3—24
1—4—23
2—3—14
2—4—13
3—4—12
1—2—3—4

$$S(L,B) = B \cdot S(L-1,B) + S(L-1,B-1)$$

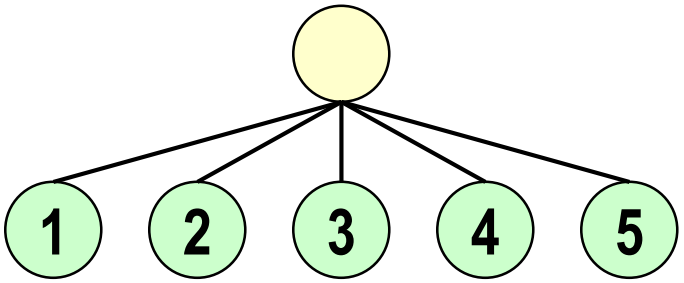
L \ B:	2	3	4	total
2	1			1
3	3	1		4
4	7	6	1	14
5	15	25	10	51
6	31	90	65	202
7	63	301	350	876
8	127	966	1701	4139
9	255	3025	7770	21146

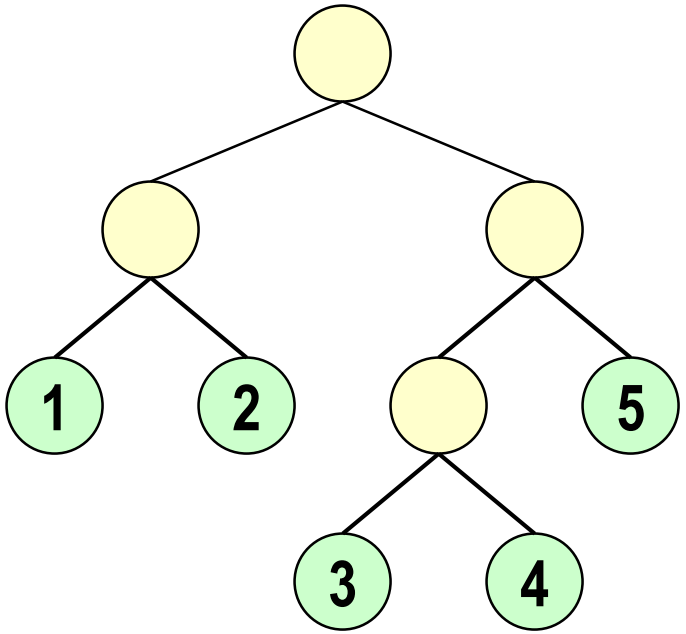
Основные моменты поиска разбиения

- Только бинарное разбиение
  - ординальные = L – 1
  - категориальные = $2^{L-1} - 1$
- Агломеративная кластеризация

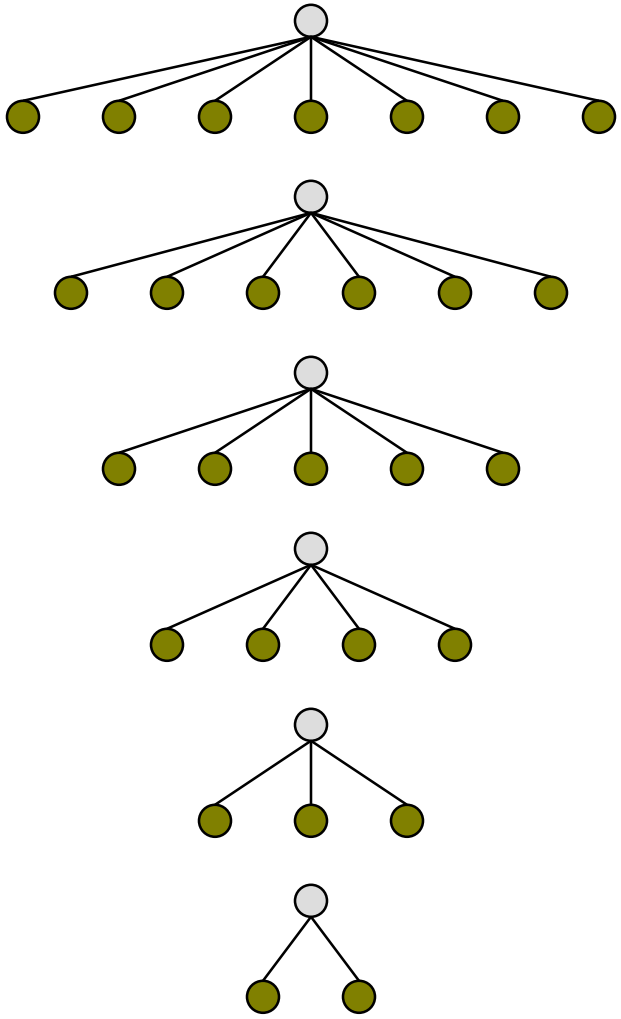
Вариантов разбиения

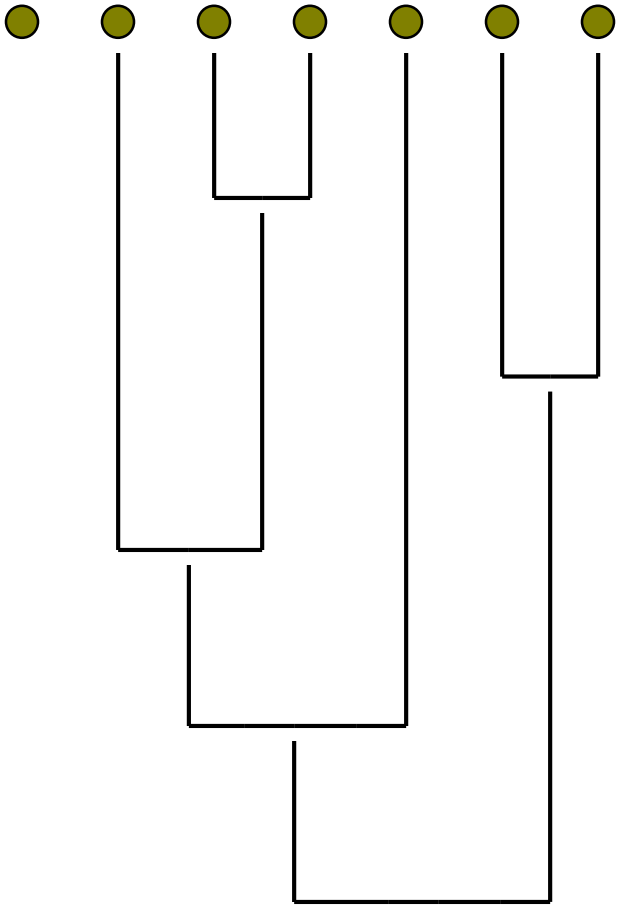
  - Kass (1980)
- Минимальный размер листа
- Множественное или бинарное





Кластеризация гипотез





Критерии разбиения для категороиального отклика

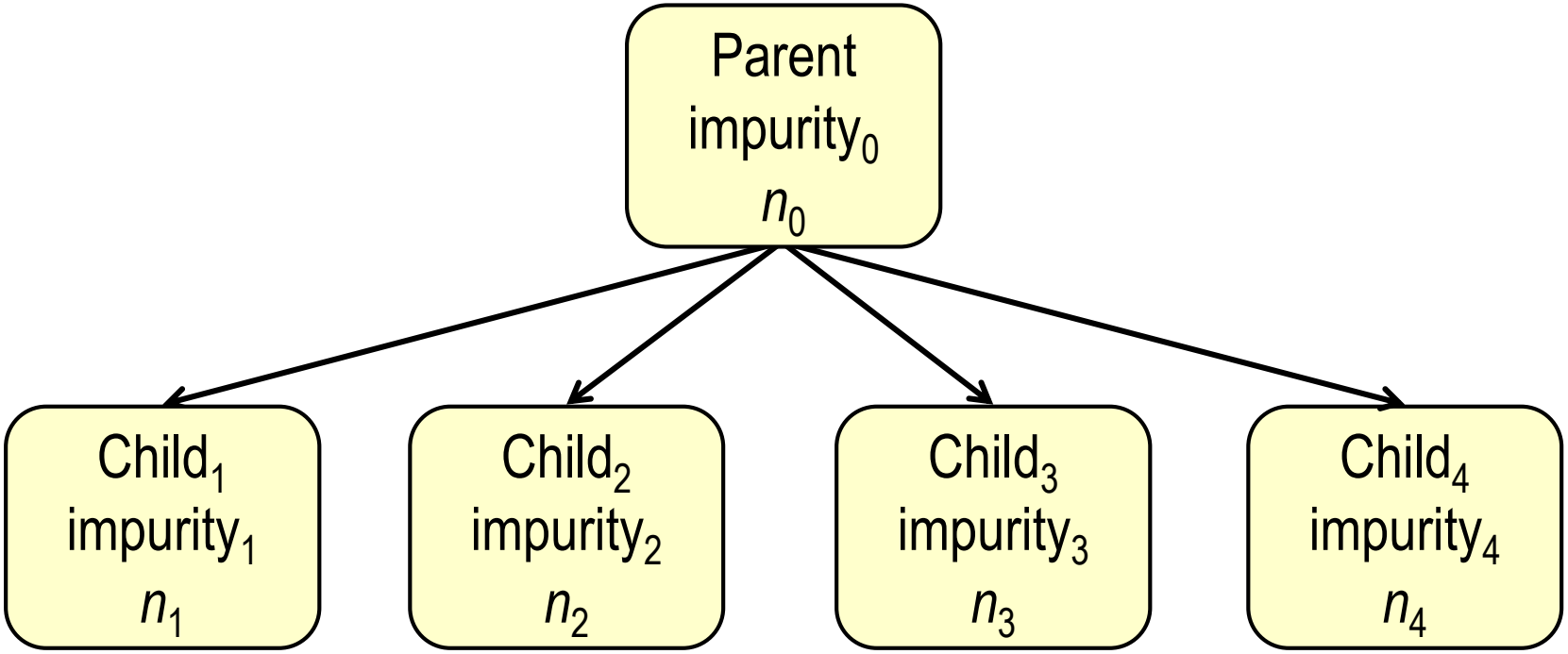
X1:	<38.5	≥38.5
1	293	71
7	363	1
9	42	294

ΔGini	Δentropy	logworth
.197	.504	140

X10:	<0.5	1-41	42-51	≥51.5
1	9	143	65	147
7	221	88	1	54
9	1	4	16	315

ΔGini	Δentropy	logworth
.255	.600	172

Уменьшение разнородности

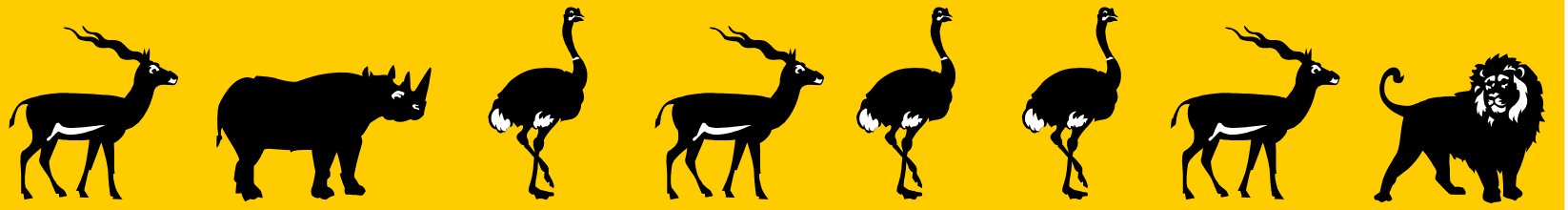


$$\Delta i = i(0) - \left(\frac{n_1}{n_0} i(1) + \frac{n_2}{n_0} i(2) + \frac{n_3}{n_0} i(3) + \frac{n_4}{n_0} i(4) \right)$$

Индекс Джини

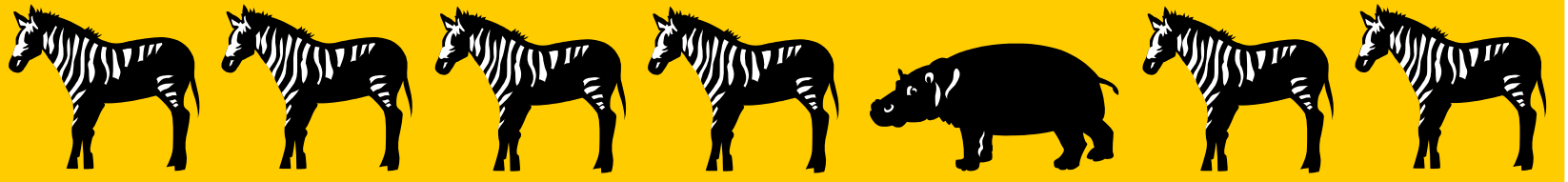
$$1 - \sum_{j=1}^{r} p_j^2 = 2 \sum_{j<k} p_j p_k$$

Высокая вариация, низкая чистота



Pr(interspecific encounter) = 1-2(3/8)^2-2(1/8)^2 = .69

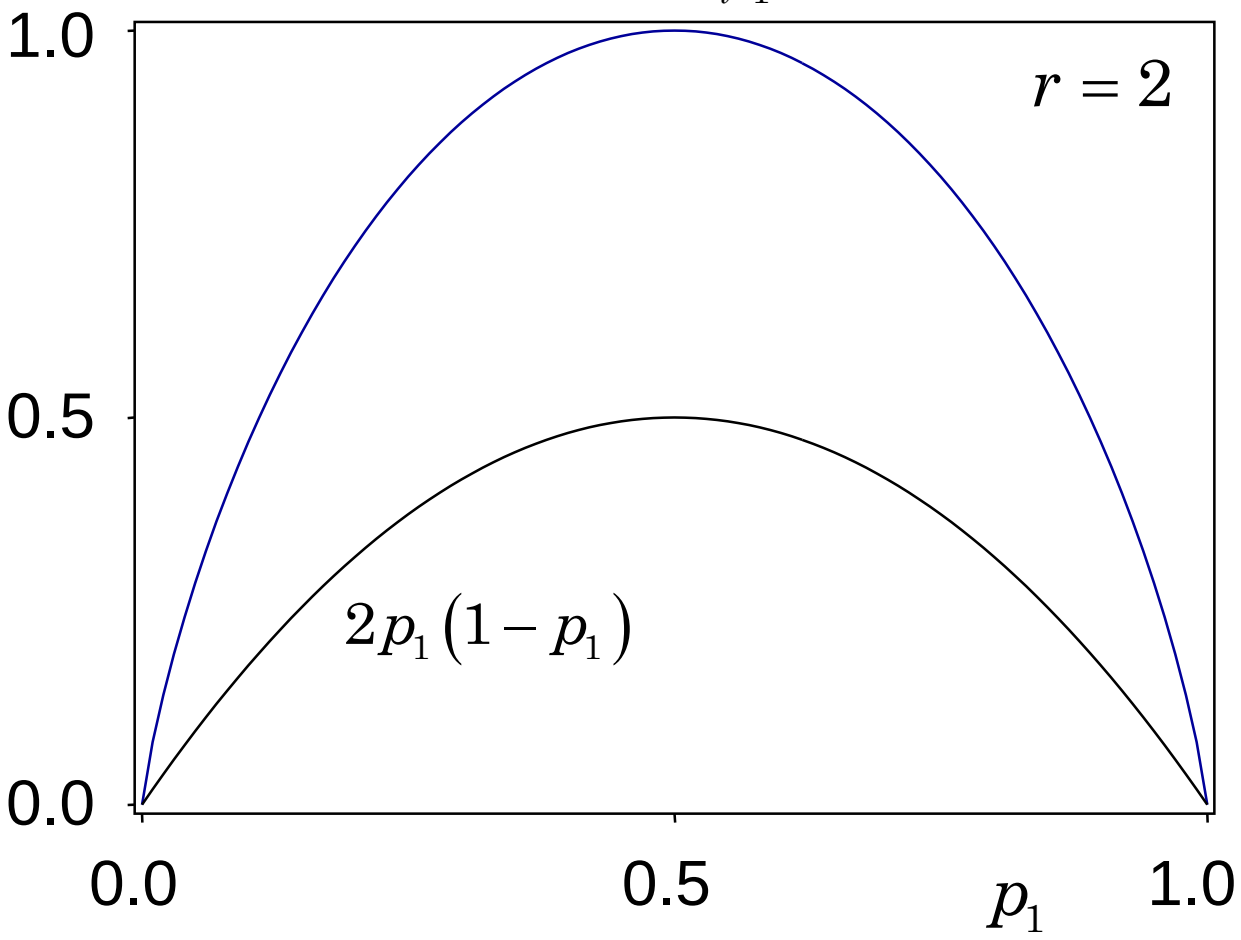
Низкая вариация, высокая чистота



Pr(interspecific encounter) = 1-(6/7)^2-(1/7)^2 = .24

Энтропия

$$H(p_1, p_2, \ldots, p_r) = -\sum_{i=1}^{r} p_i \log_2(p_i)$$



Хи2 тест

Наблюдаемые частоты

X1: <38.5≥38.5

1	293	71	.342
7	363	1	.342
9	42	294	.316
	.656	.344	*n*=1064

Ожидаемые (по гипотезе)

239	125
239	125
225	116

$\frac{(O-E)^2}{E}$

12	23
64	123
149	273

$$\chi^2_\nu = \sum \frac{(O-E)^2}{E} = 644$$

$$\nu = (3-1)(2-1) = 2$$

Корректировка *p*-Value

X1: 38.5

1	293	71
7	363	1
9	42	294

X1: 17.5 36.5

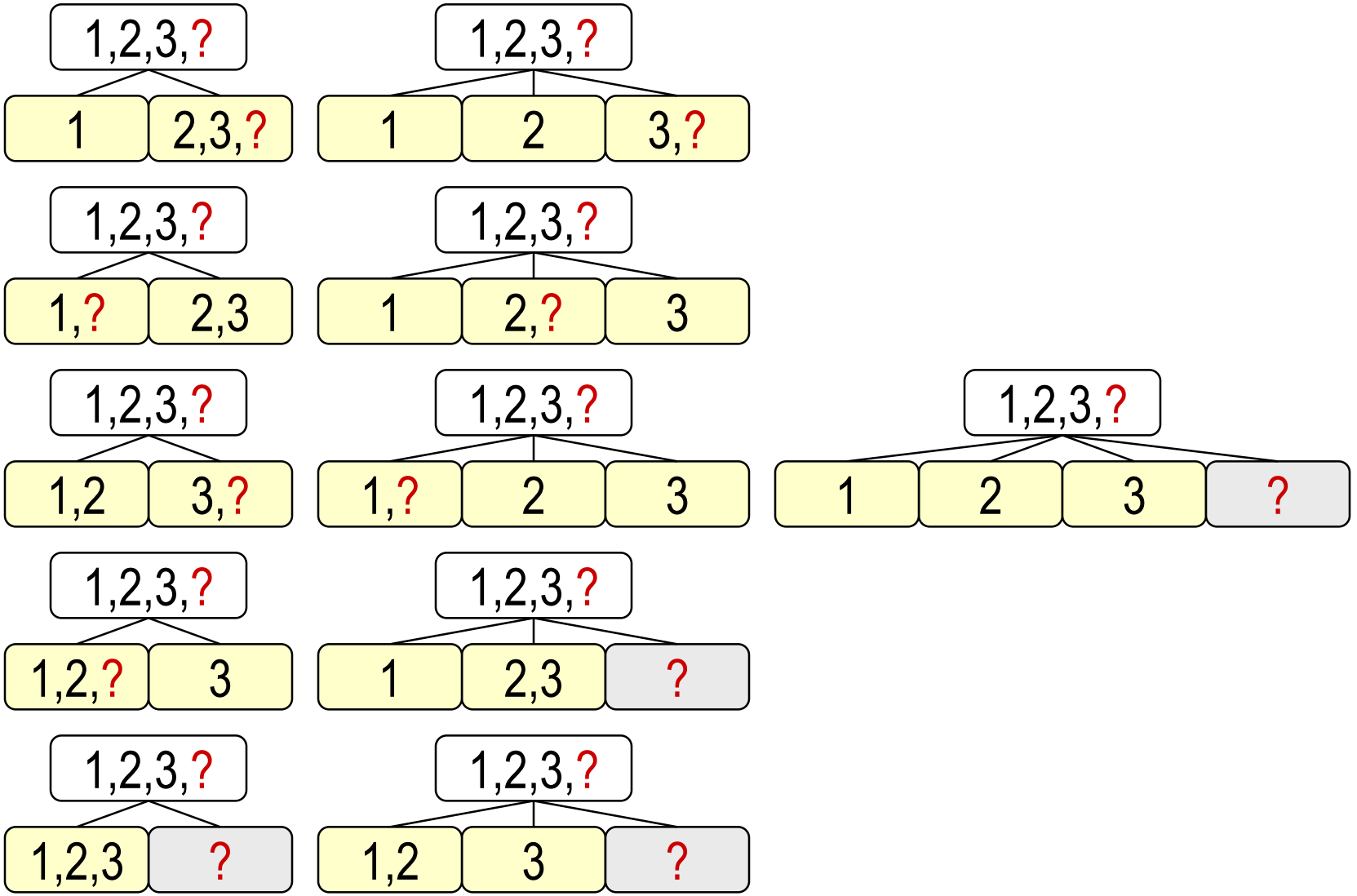
1	249	42	73
7	338	25	1
9	26	16	294

X10: 0.5 41.5 51.5

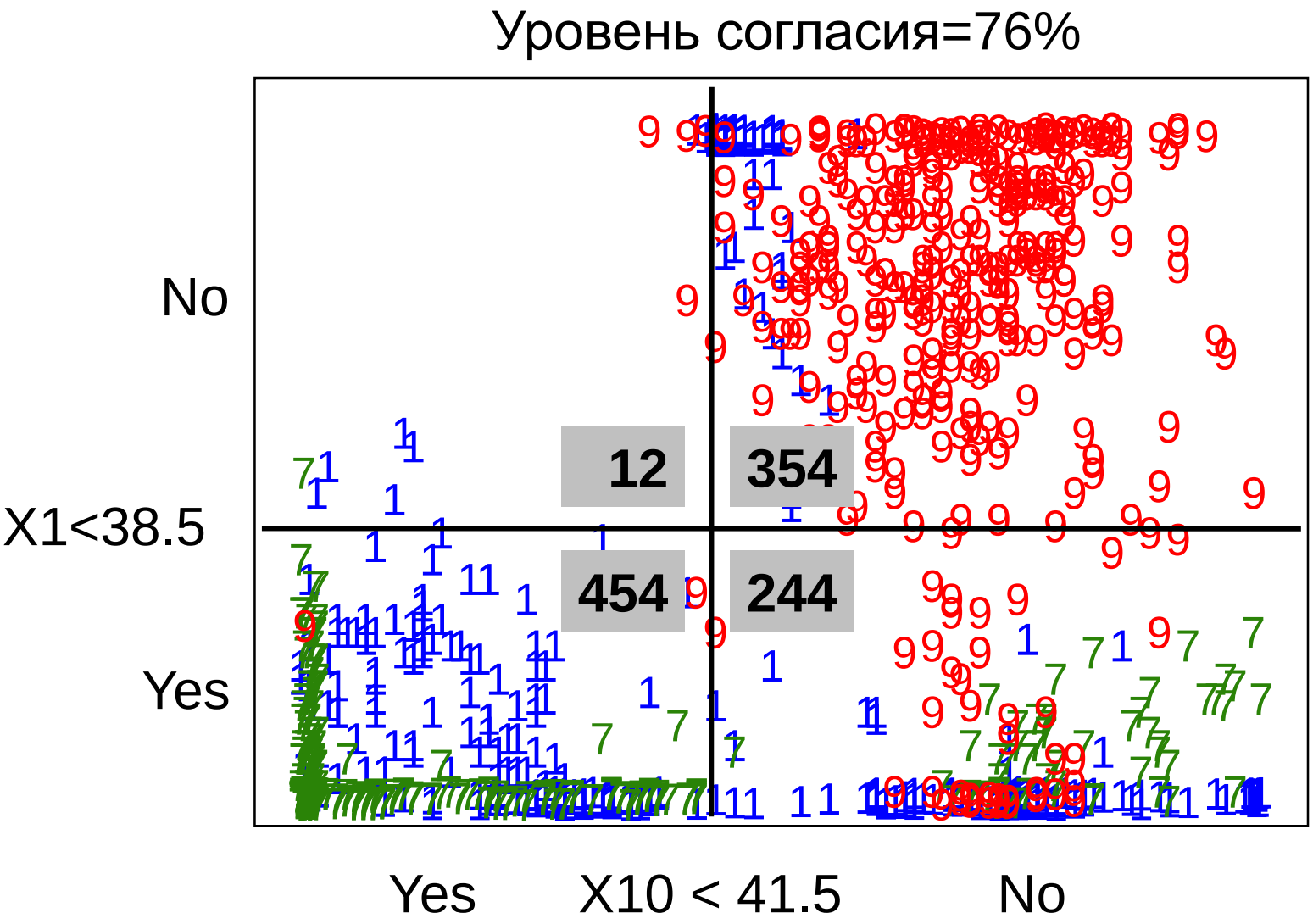
1	9	143	65	147
7	221	88	1	54
9	1	4	16	315

χ^2_ν	ν	$-\log_{10}(\mathrm{P})$	m	$-\log_{10}(m\mathrm{P})$
644	2	140	96	138
660	4	141	4560	137
814	6	172	156849	167

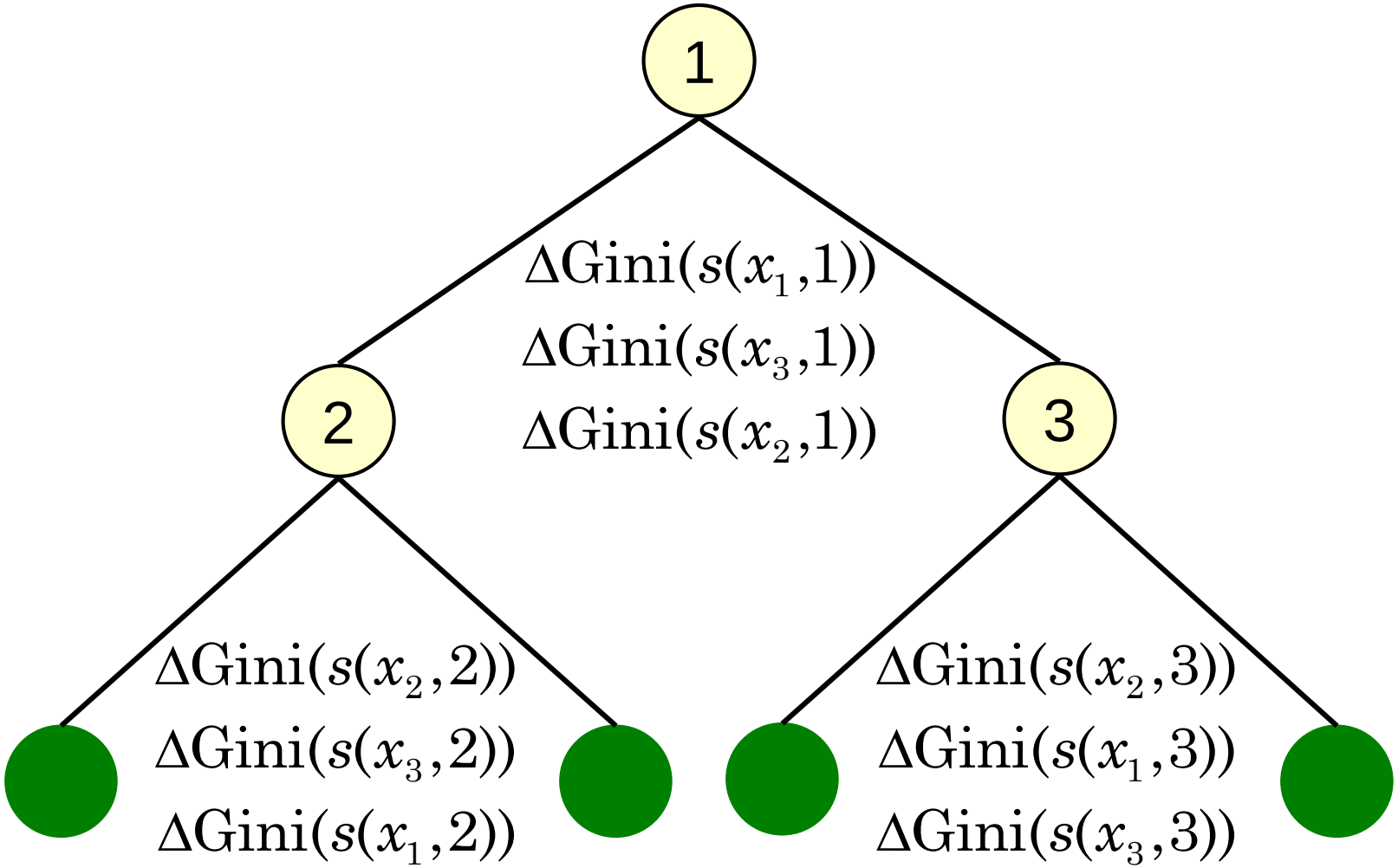
Пропущенные значения



Суррогатные разбиения

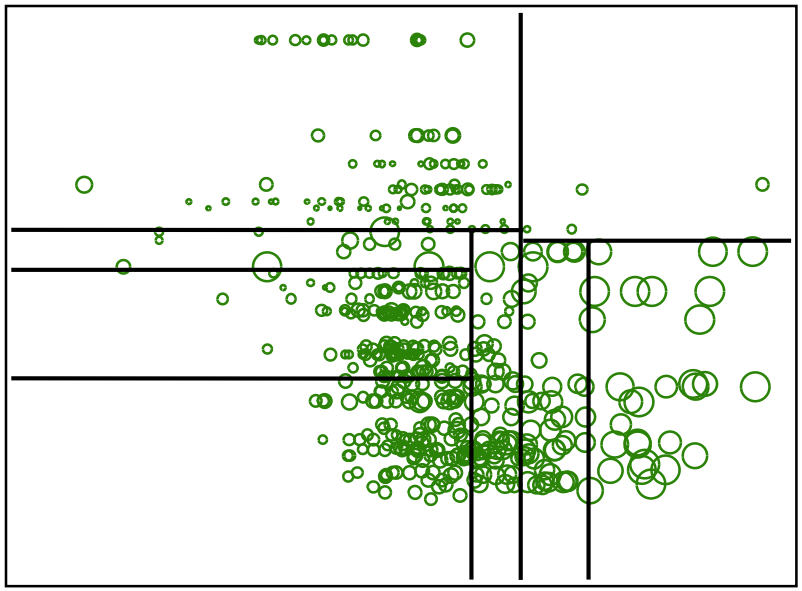


Важность переменных



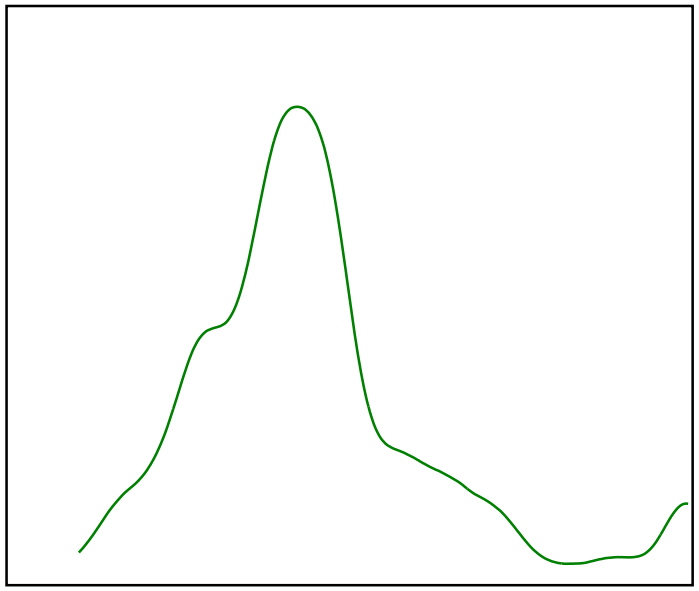
Непрерывный отклик

NOX



RM

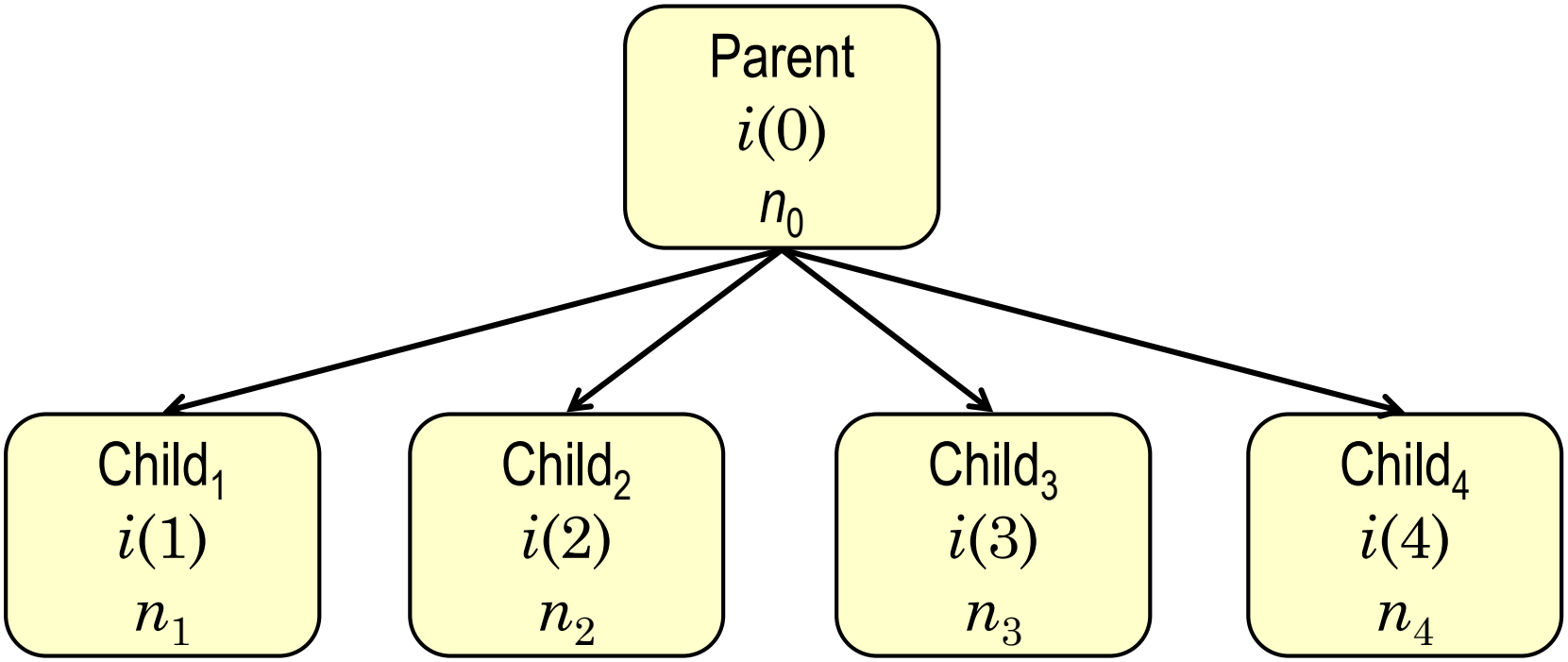
Плотность



0 25 50

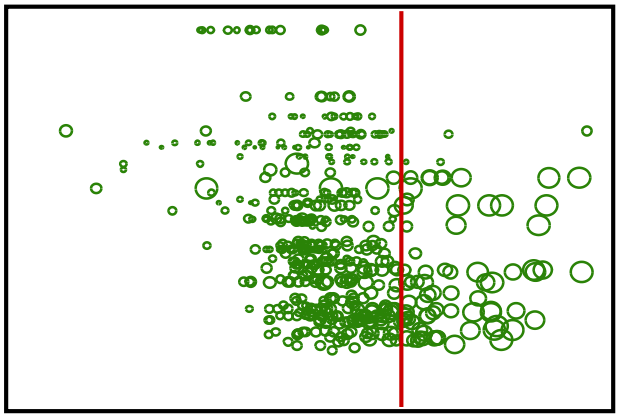
MEDV

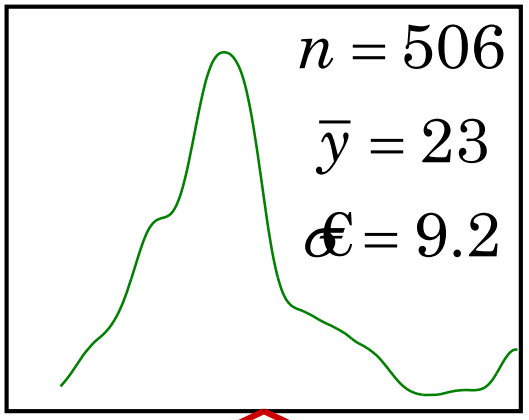
Уменьшение вариации



$$\Delta i = i(0) - \left(\frac{n_1}{n_0} i(1) + \frac{n_2}{n_0} i(2) + \frac{n_3}{n_0} i(3) + \frac{n_4}{n_0} i(4) \right)$$

Уменьшение вариации

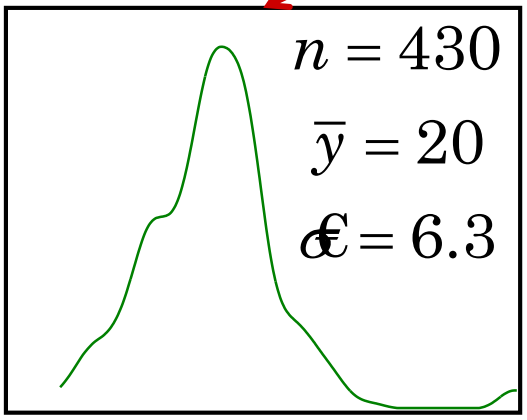


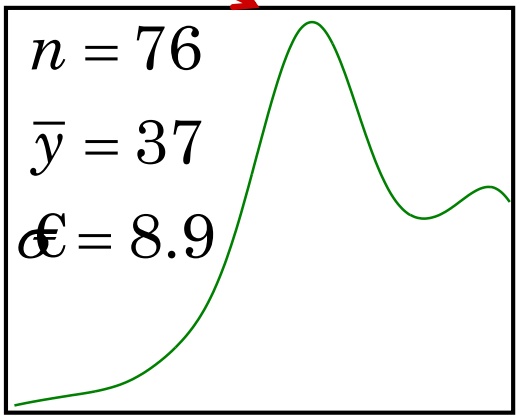


yes

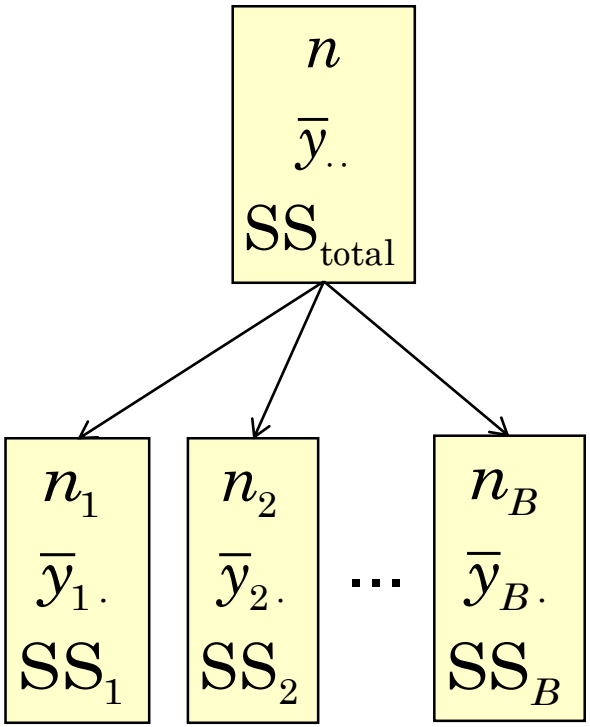
RM<6.94

no





One-Way ANOVA



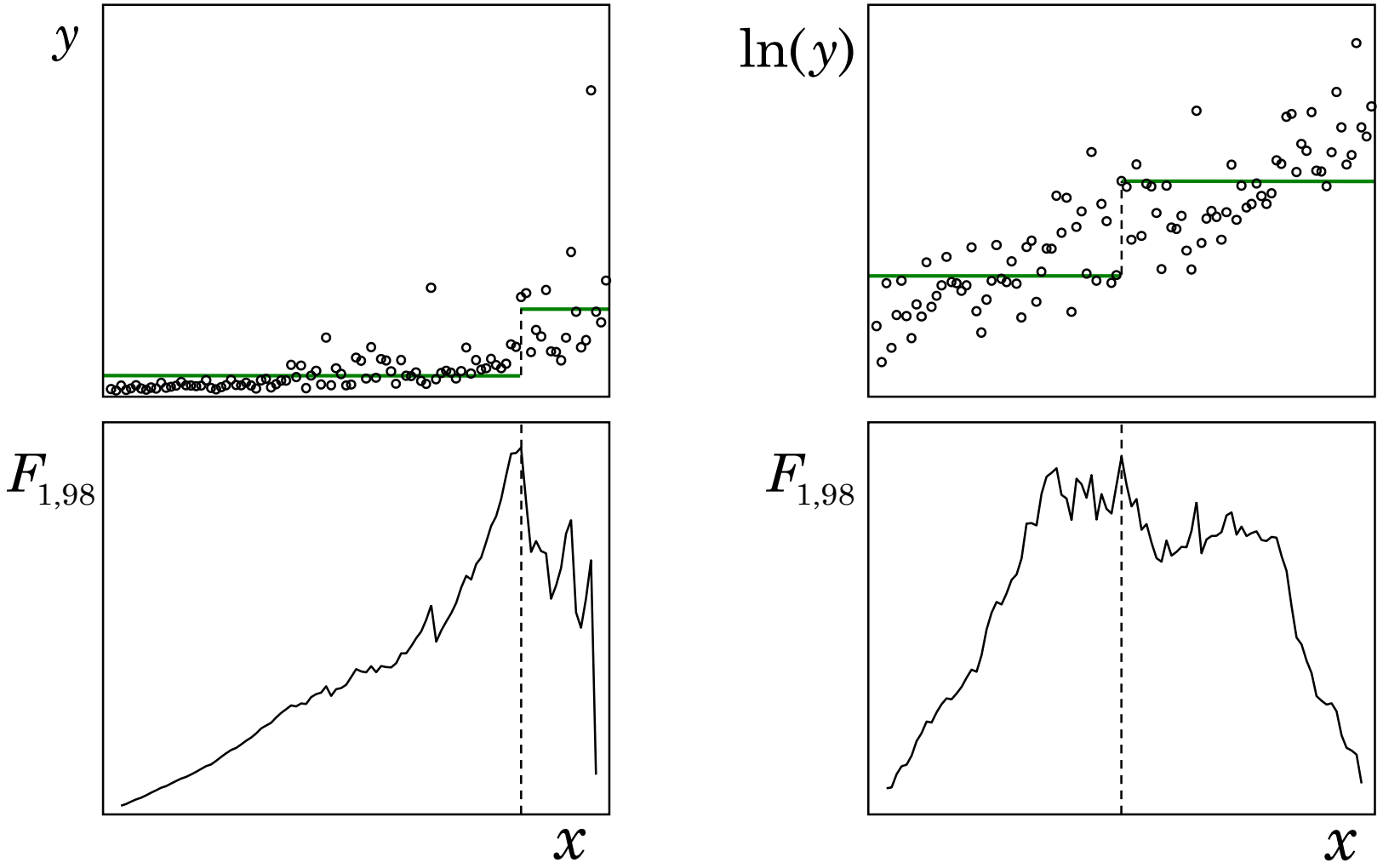
$$F = \left(\frac{SS_{between}}{SS_{within}} \right) \left(\frac{n-B}{B-1} \right) \sim F_{B-1,\, n-B}$$

$$SS_{between} = \sum_{i=1}^{B} n_i \left(\bar{y}_{i.} - \bar{y}_{..} \right)^2$$

$$SS_{within} = \sum_{i=1}^{B} SS_i = \sum_{i=1}^{B} \sum_{j=1}^{n_i} \left(y_{ij} - \bar{y}_{i.} \right)^2$$

$$SS_{total} = \sum_{i=1}^{B} \sum_{j=1}^{n_i} \left(y_{ij} - \bar{y}_{..} \right)^2$$

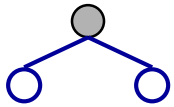
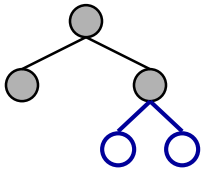
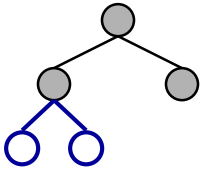
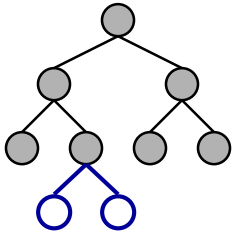
Гетероскедастичность



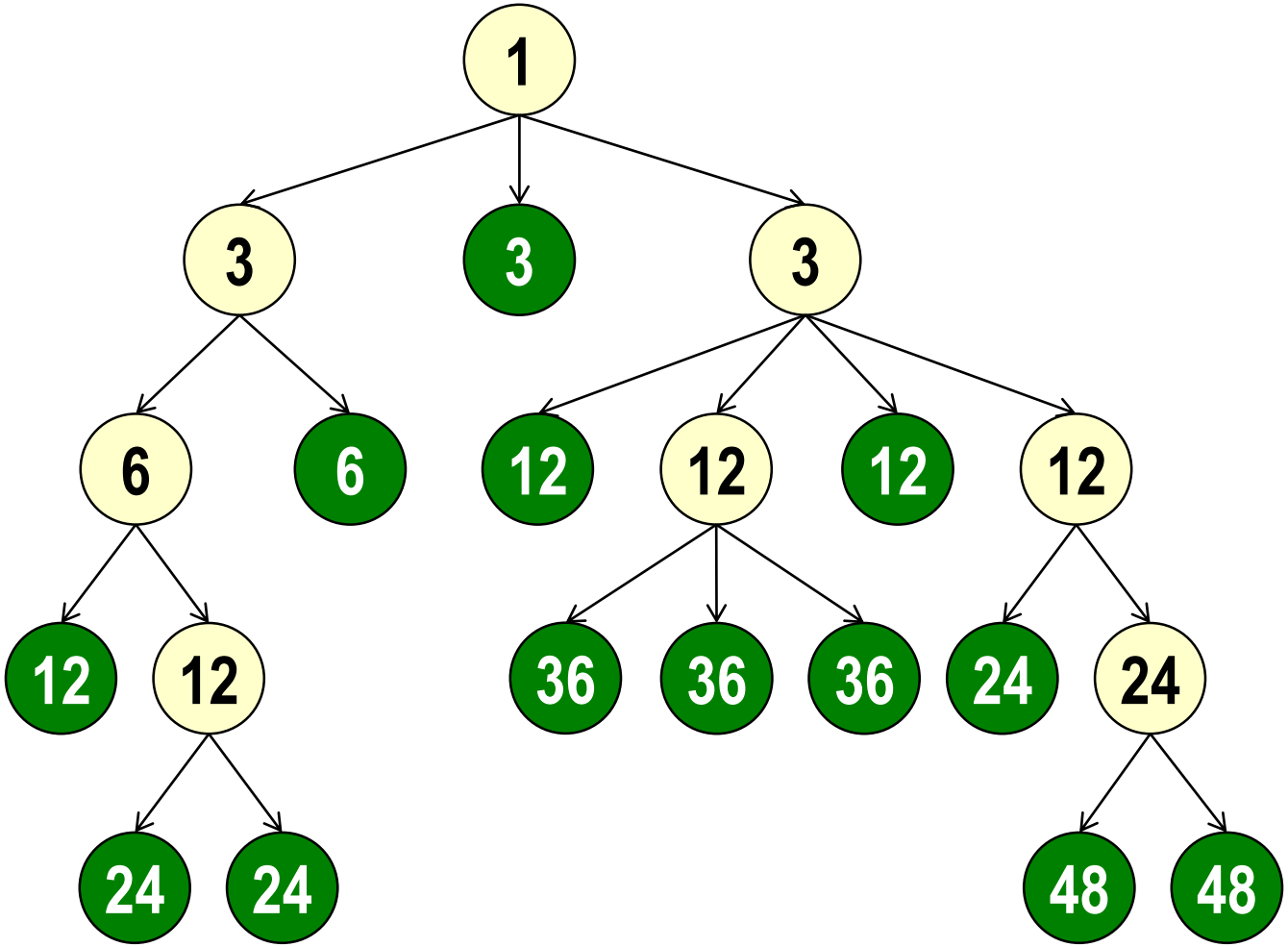
Переобучение деревьев решений

- Описанный выше процесс может давать хорошие прогнозы для обучающего набора, но может привести к *переобучению*, что приведет к ухудшению качества на тестовом наборе.
- Меньшее по размеру дерево с меньшим количеством разбиений (то есть меньшим количеством областей $R_1,\ldots,R_J$) может привести к снижению дисперсии и лучшей интерпретируемости.
- Одной из возможных альтернатив является ранняя остановка построения дерева, определяемая параметрами:
  - Максимальная глубина дерева
  - Минимальное число наблюдений в листе
  - Порог на p-value

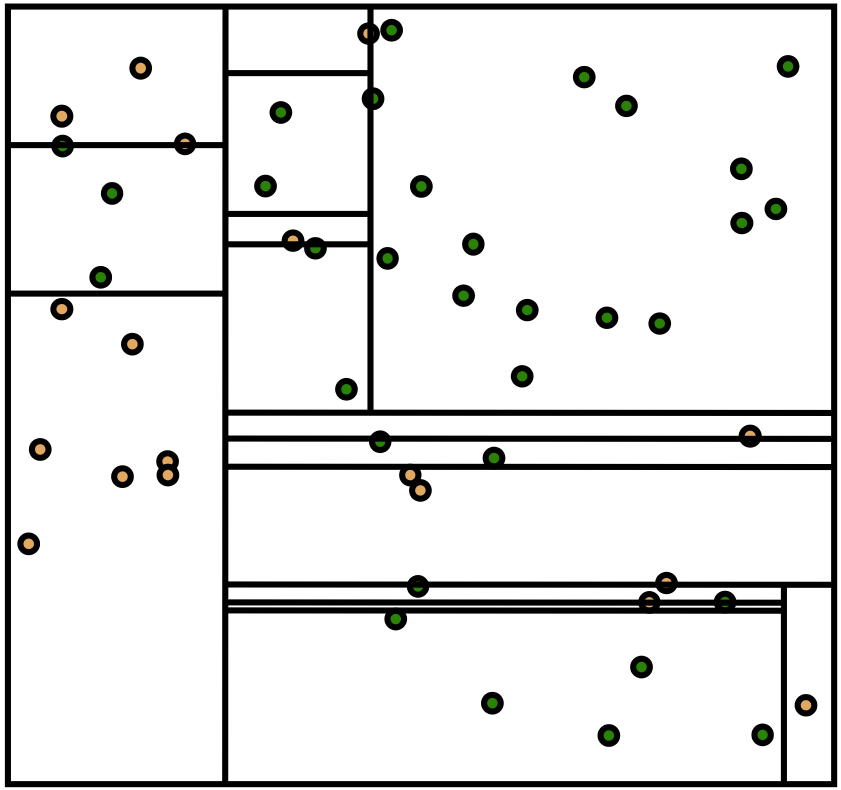
Рост дерева

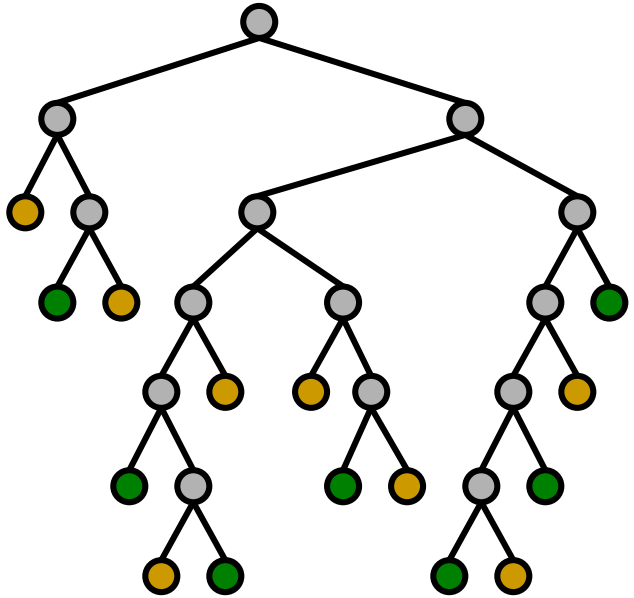
	$-\log_{10}(\mathrm{P})$	m	$-\log_{10}(m\mathrm{P})$	d	$-\log_{10}(2^d m\mathrm{P})$
	26.7	53	24.9	0	24.9
	3.12	14	1.97	1	1.67
	1.63	39	.039	1	–.26
	2.40	11	1.36	2	.76

Множитель глубины



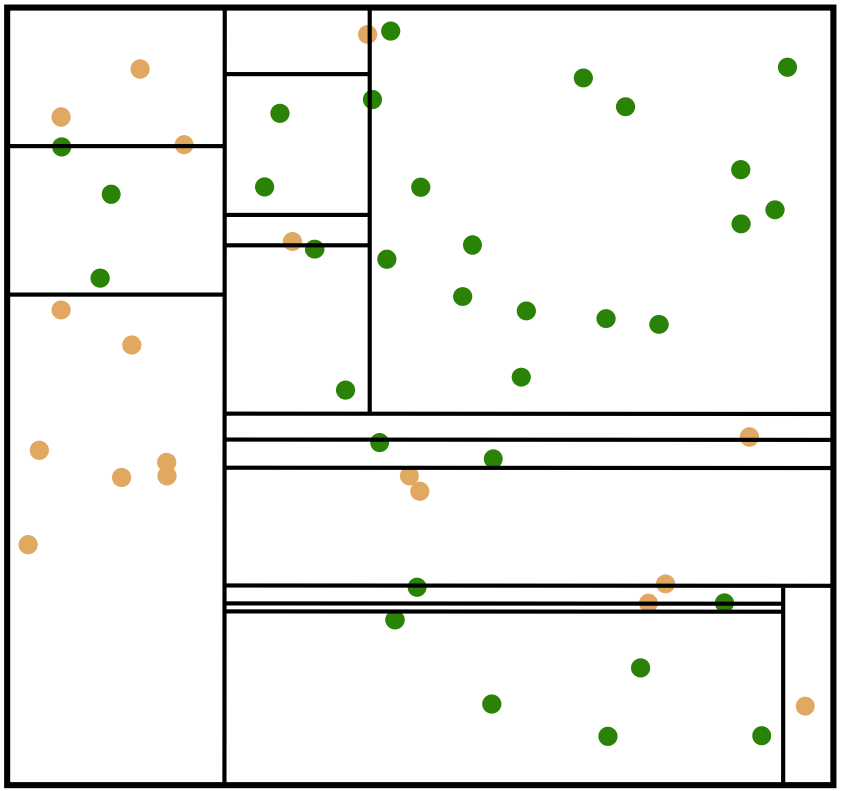
Максимальное дерево



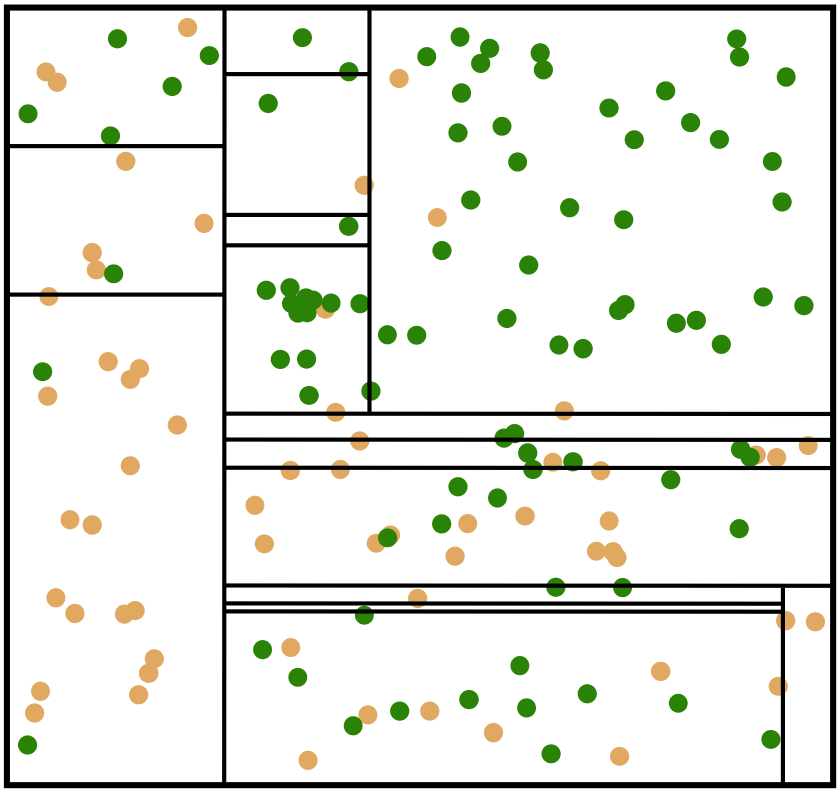


Переобучение

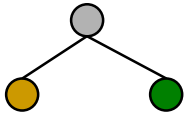
Тренировочный набор



Новые данные

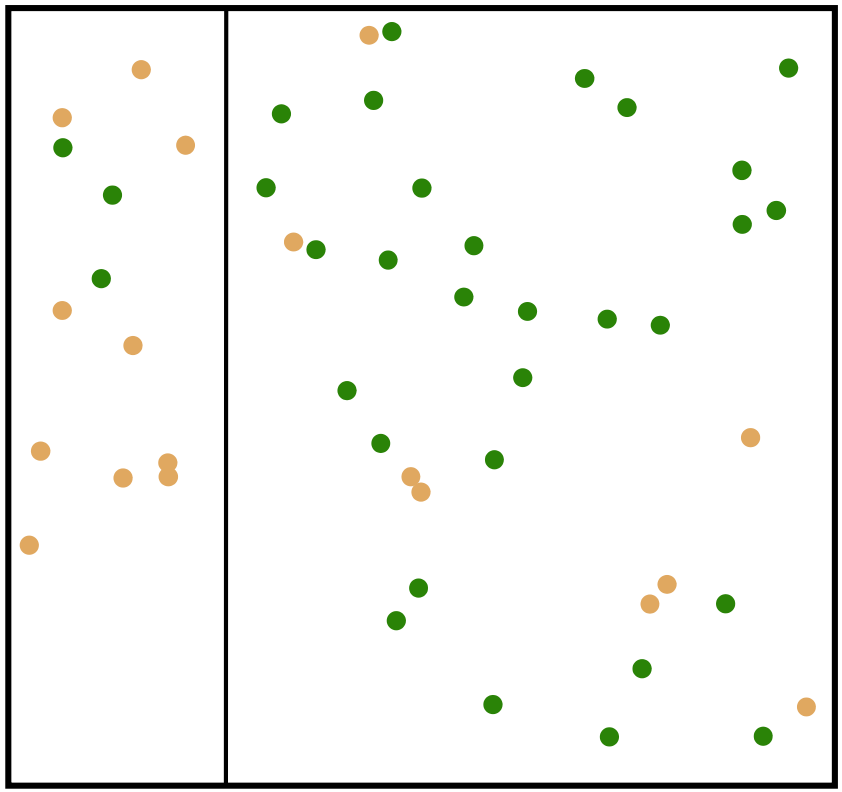


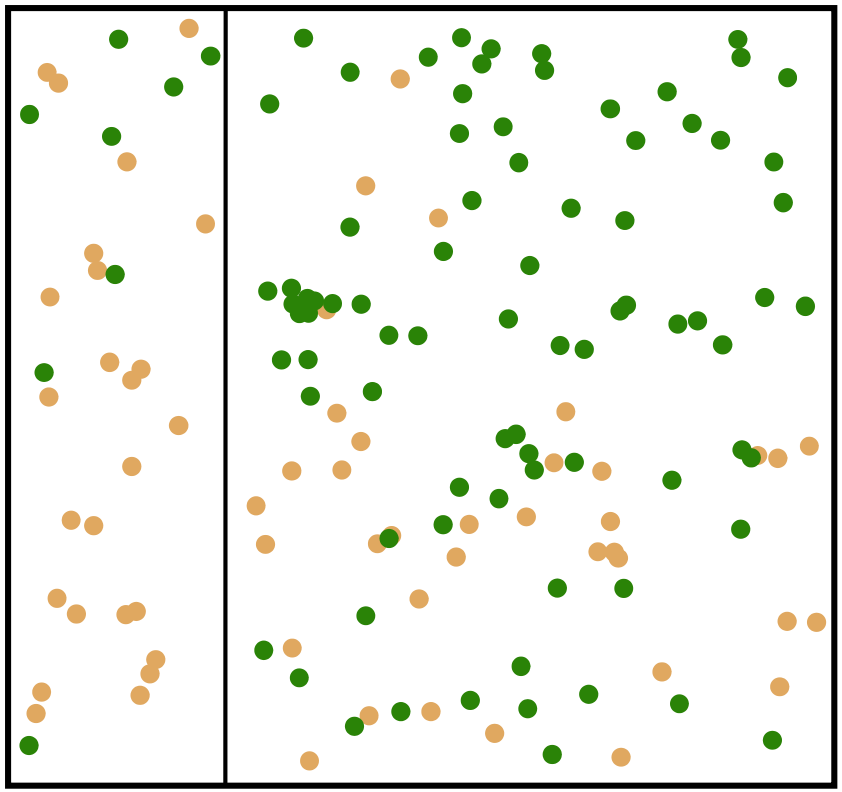
Недообучение



Тренировочный набор

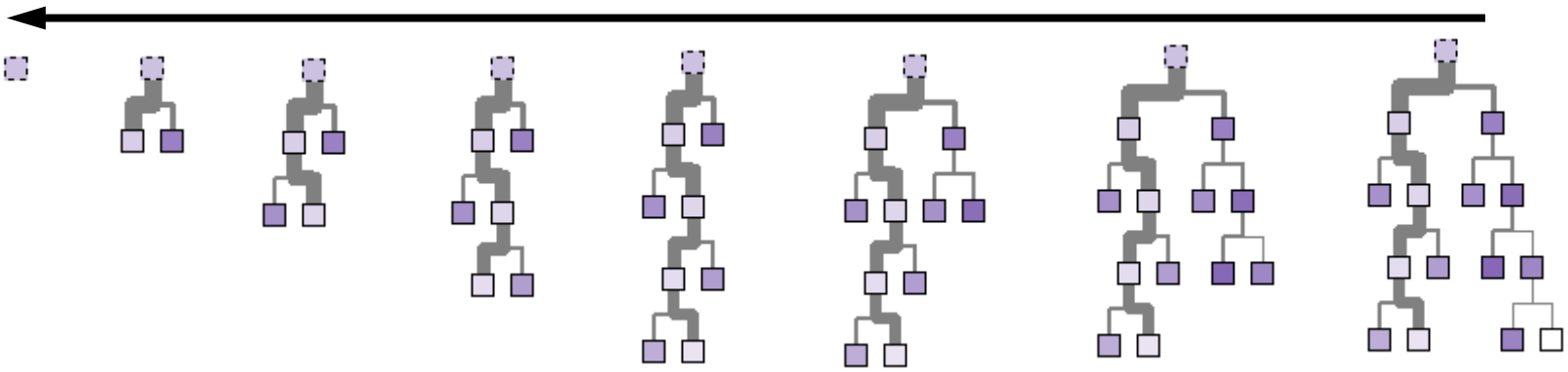
Новые данные





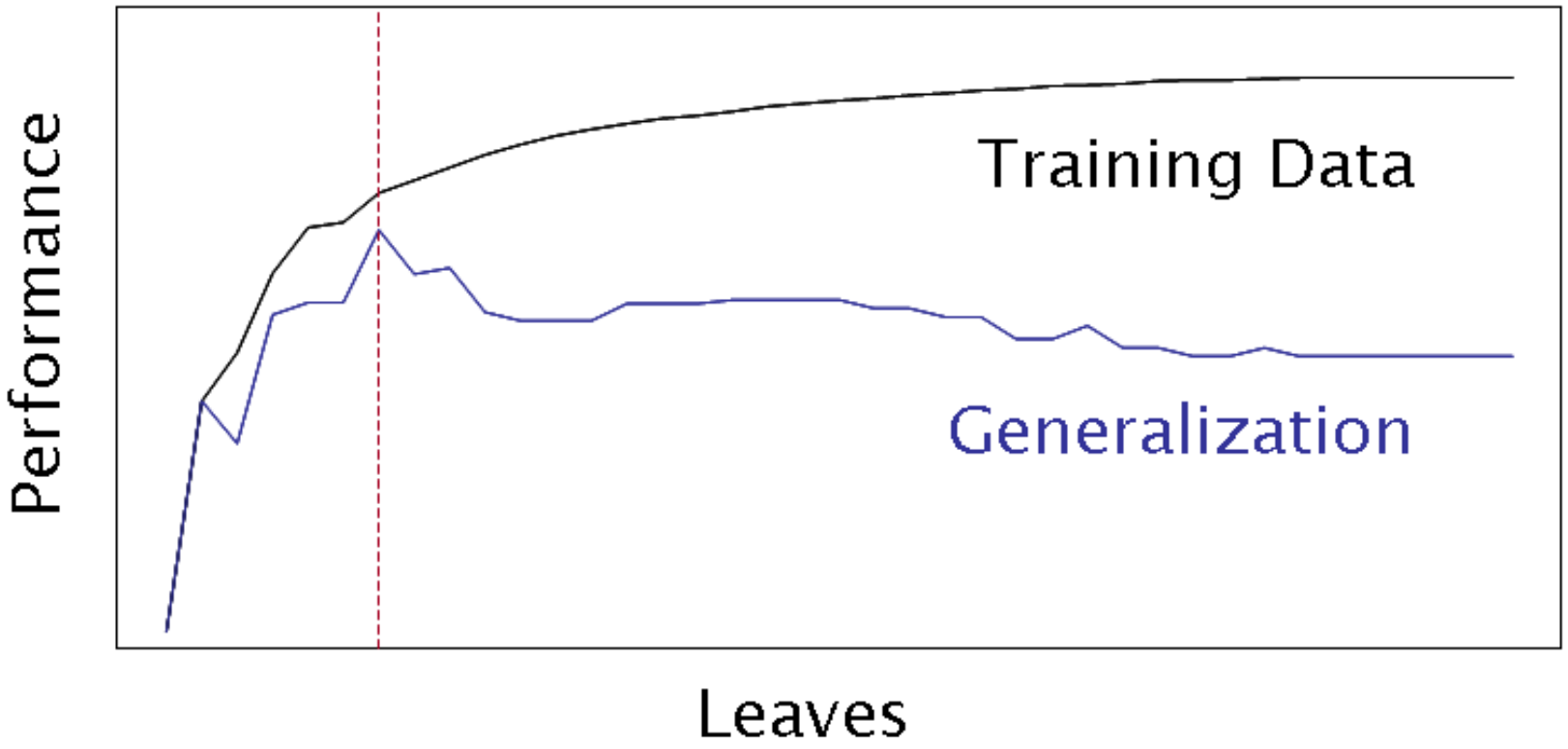
Обрубание дерева

- Лучшей стратегией является построение большого дерева T_0, а затем выполнение *отсечения* для получения *поддерева*
- Для этого используется подход *сокращения сложности,* также известный как *удаление самых слабых связей*
- Используем валидационный набор или кросс валидацию для выбора оптимального размера дерева.
- Алгоритм: строим максимальное дерево и последовательно обрубаем ветки с отбором

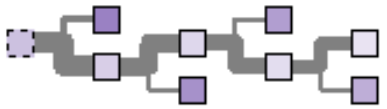
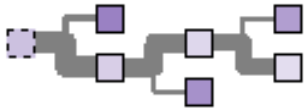
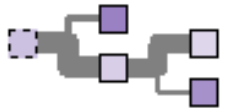
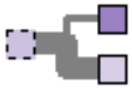


Обрубание ветвей

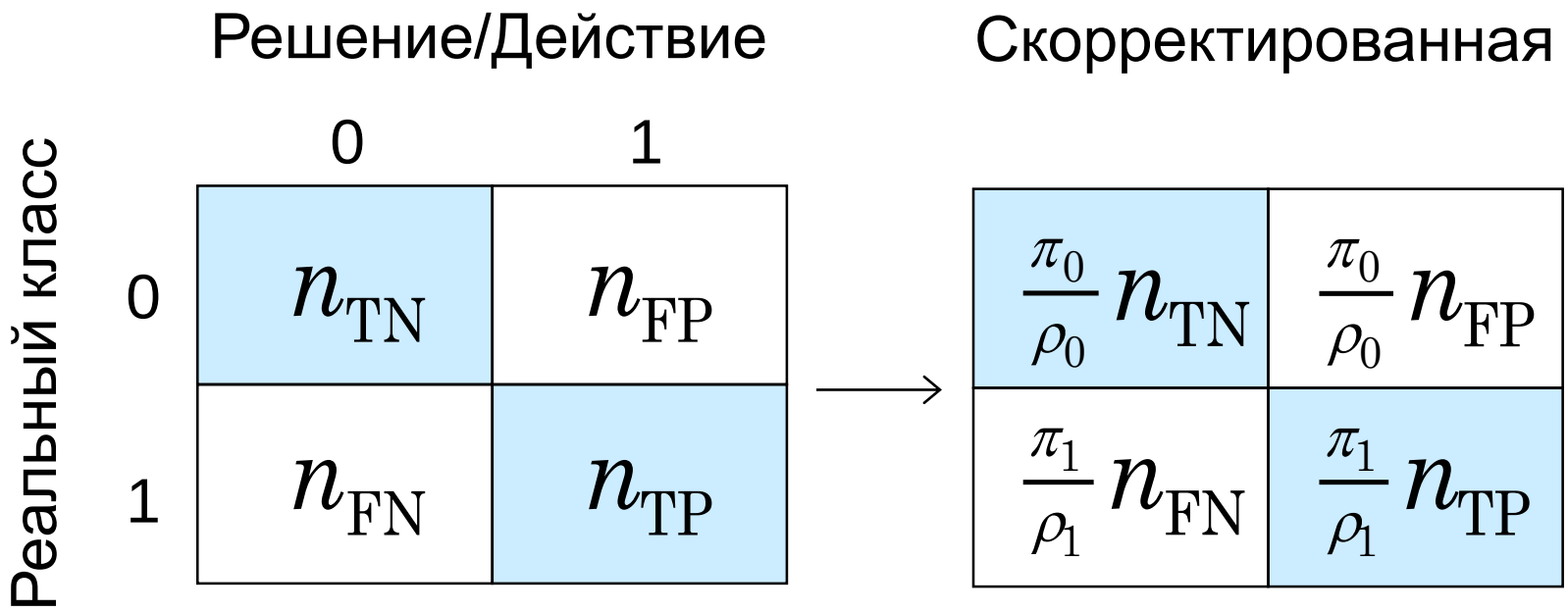
Выбираем лучшее на валидационном наборе.



Критерий выбора поддерева

Leaves	Accuracy	Profit	ASE	
5	.90/.88	.51/.43	.17/.15	
4	.90/.88	.51/.40	.18/.14	
3	.89/.91	.49/.44	.19/.16	
2	.88/.91	.49/.44	.20/.16	
1	.59/.64	.04/ −.1	.48/.46	

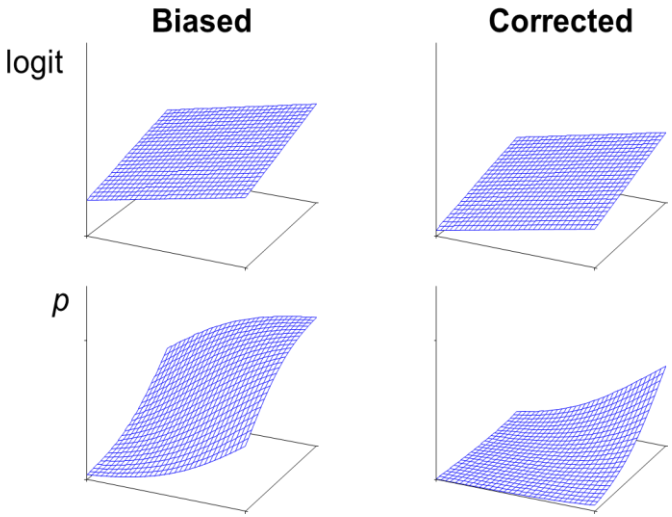
Оценка точности

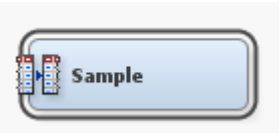


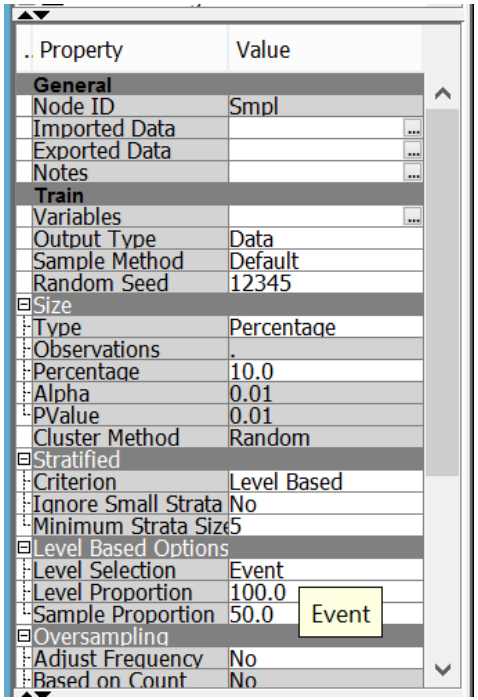
$$\mathrm{Accuracy} = \frac{1}{n}\left(\frac{\pi_0}{\rho_0} n_{\mathrm{TN}} + \frac{\pi_1}{\rho_1} n_{\mathrm{TP}}\right)$$

«Балансировка» выборки (oversampling)

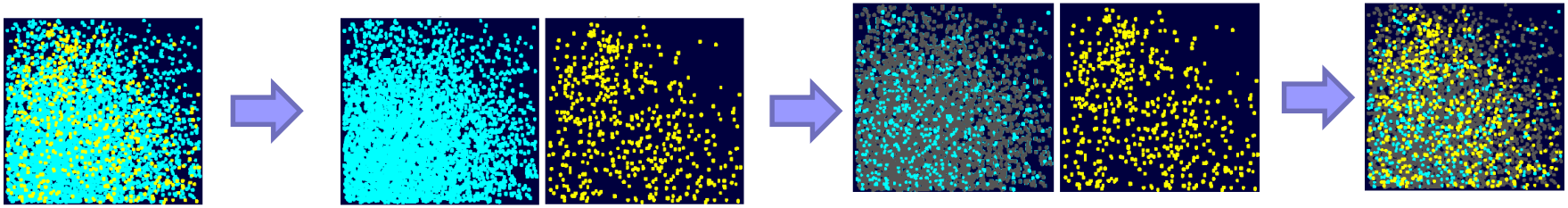
- Порог отсечения для логистической функции:



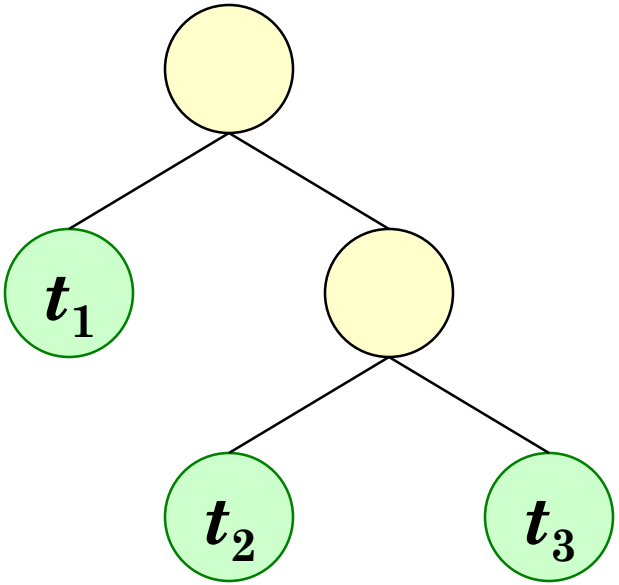




- «Балансировка»:

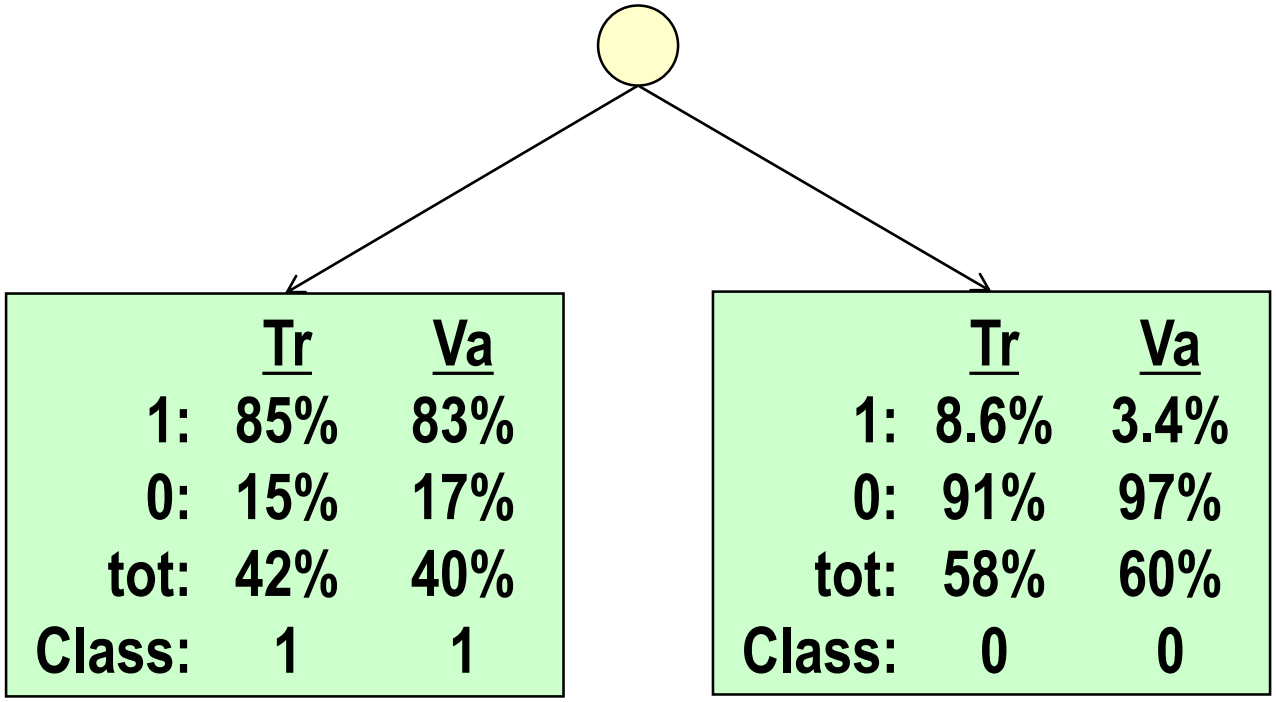


Точность дерева



$$\text{Accuracy} = \frac{1}{n}\left(n(t_1)\,acc(t_1) + n(t_2)\,acc(t_2) + n(t_3)\,acc(t_3)\right)$$

Максимизация точности



Training Accuracy = (.42)(.85) + (.58)(.91) = .88

Validation Accuracy = (.40)(.83) + (.60)(.97) = .91

Матрица выигрыша

Решение

Реальный класс	0	1
0	δ_{TN}	δ_{FP}
1	δ_{FN}	δ_{TP}

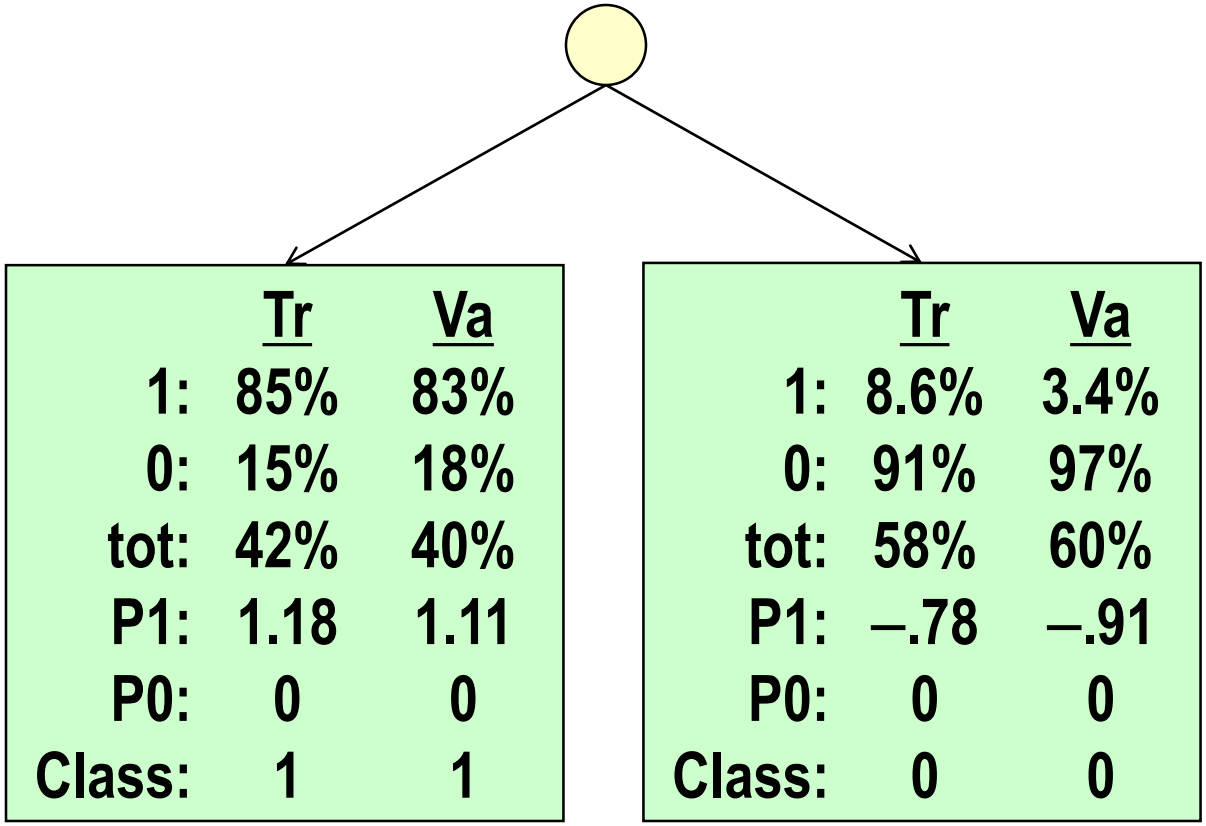
Правило Байеса:

Решение 1 если

$$P > \frac{1}{1 + \left(\frac{\delta_{TP} - \delta_{FN}}{\delta_{TN} - \delta_{FP}} \right)}$$

$$\text{Profit} = \frac{1}{n} \left(\delta_{TN} \frac{\pi_0}{\rho_0} n_{TN} + \delta_{FP} \frac{\pi_0}{\rho_0} n_{FP} + \delta_{FN} \frac{\pi_1}{\rho_1} n_{FN} + \delta_{TP} \frac{\pi_1}{\rho_1} n_{TP} \right)$$

Максимизация выигрыша



	Tr	Va
1:	85%	83%
0:	15%	18%
tot:	42%	40%
P1:	1.18	1.11
P0:	0	0
Class:	1	1

	Tr	Va
1:	8.6%	3.4%
0:	91%	97%
tot:	58%	60%
P1:	–.78	–.91
P0:	0	0
Class:	0	0

реальность \ прогноз	1	0
1	1.56	0
0	–1	0

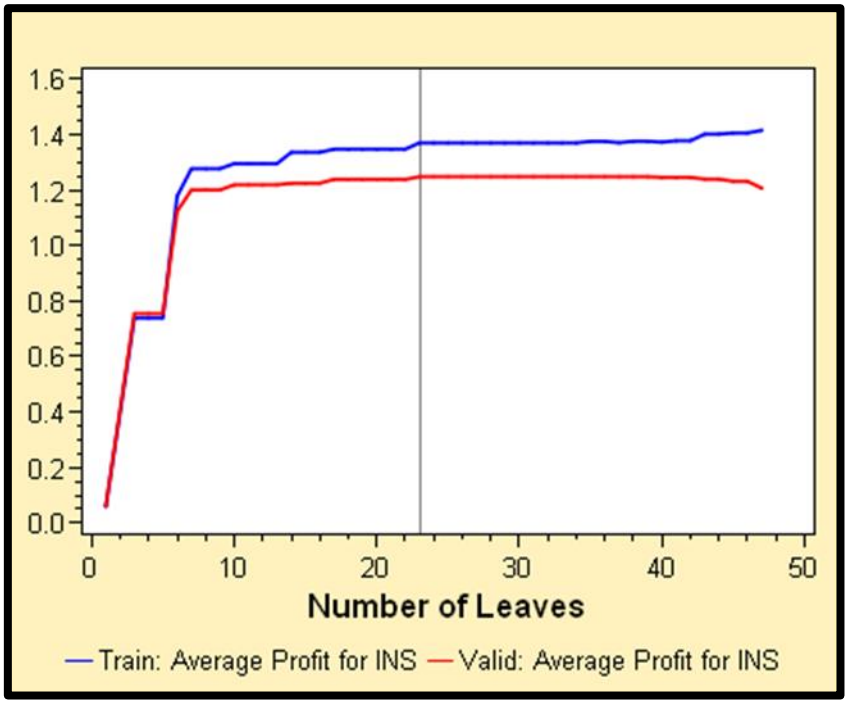
Матрица выигрыша

Training Profit = (.42)(1.18) + (.58)(0) = .50

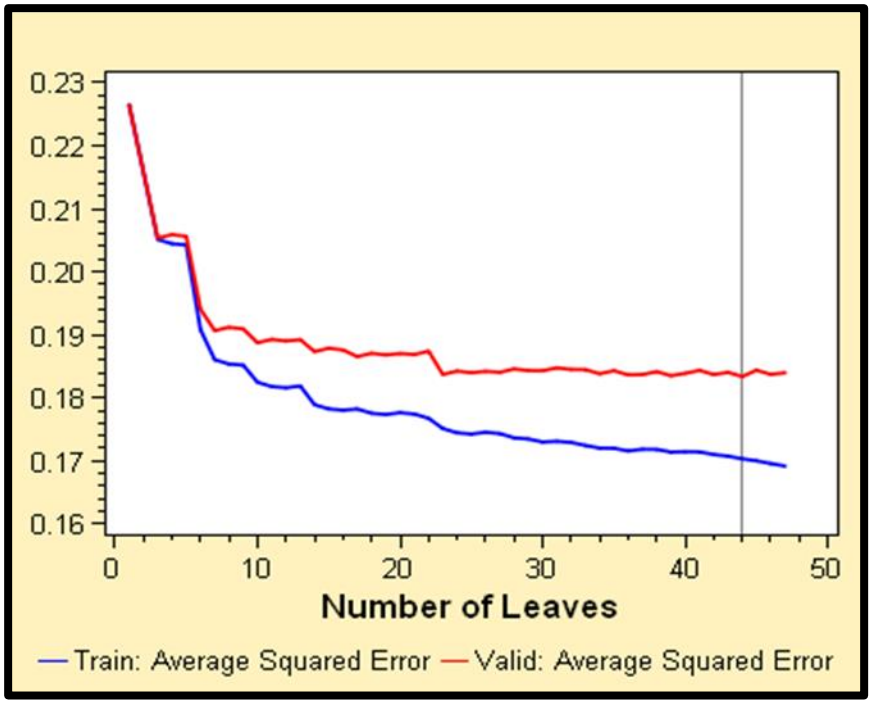
Validation Profit = (.40)(1.11) + (.60)(0) = .44

Вероятностное дерево

Выигрыш



Среднеквадратичная ошибка



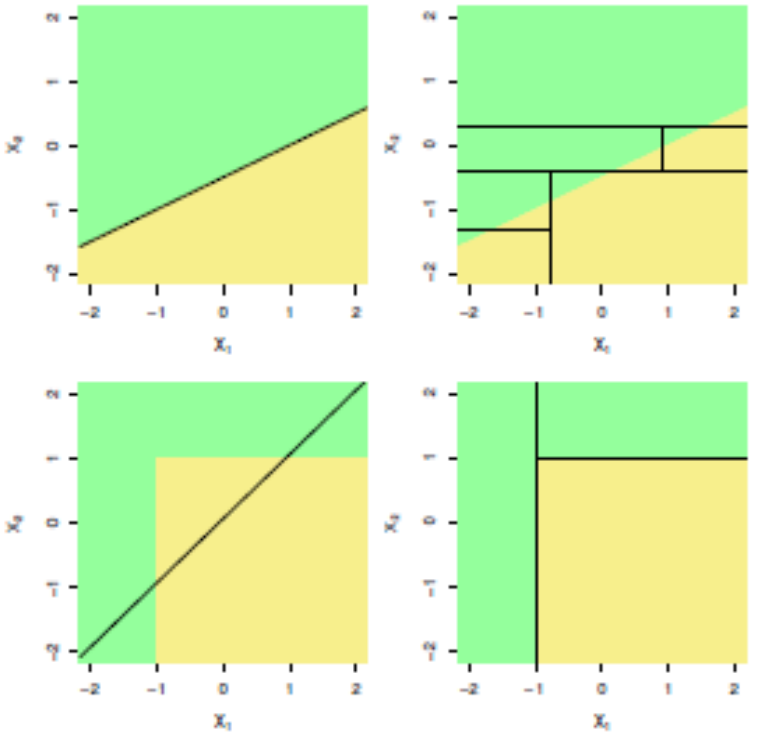
$$ASE_{nominal} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}(y_i - p_k(x_i))^2$$

$$ASE_{interval} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

Особенности популярных алгоритмов построения деревьев решений

Свойства	CHAID (Kass)	CART (Breiman)	C5 (Quinlan)
Критерий для числ. отклика	Фишер	Вариация	нет
Критерий для симв. отклика	Хи-квадрат	Джини	Энтропия
Работа с пропусками	Отдельная ветвь	Подстановка	Пропорция по ветвям (игнор)
Особенности	Корректировка Бонферрони и глубина	Линейные комбинации при разбиении	Алгоритм «сокращения» правил
Обрубание вервей	НЕТ	По точности	По точности

Деревья решений vs линейные модели



Верхний ряд: истинная линейная граница; Нижний ряд: истинная нелинейная граница.

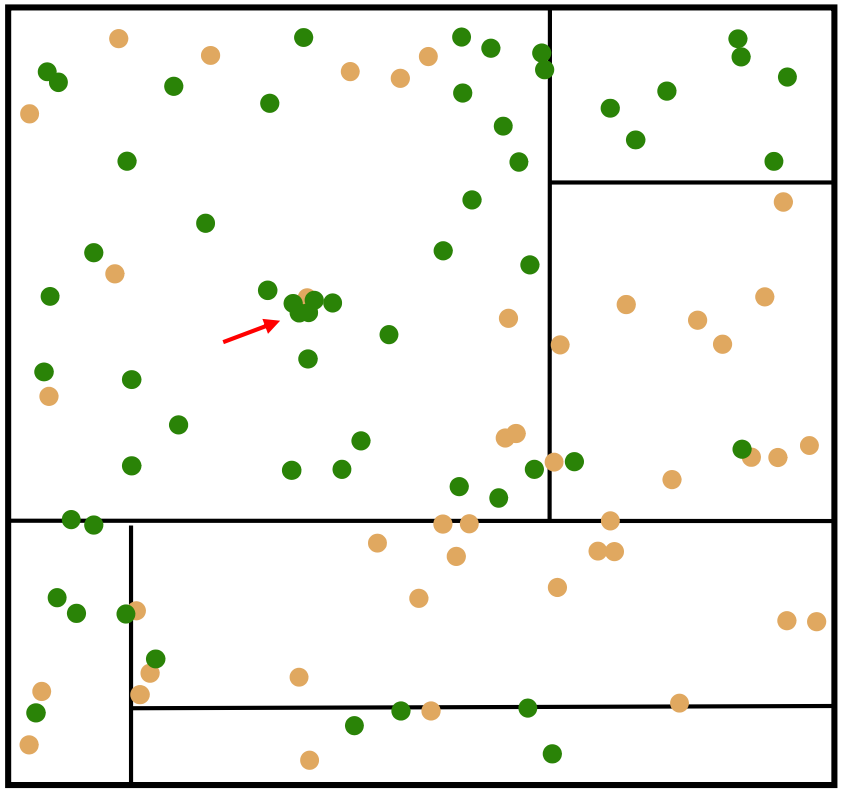
Левый столбец: линейная модель; Правый столбец: модель на основе деревьев решений

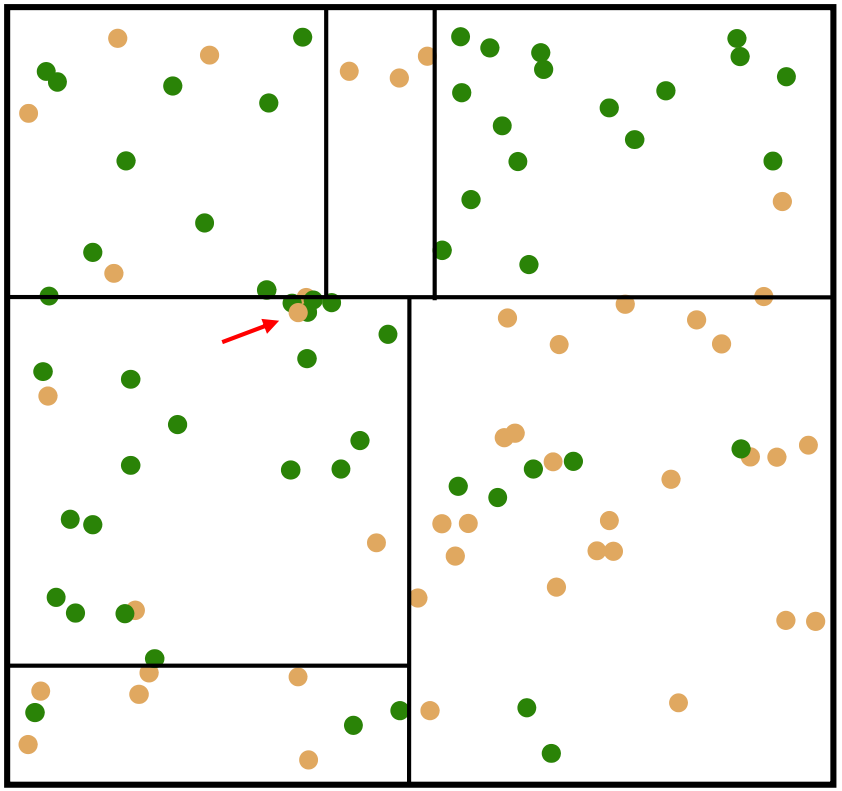
Преимущества и недостатки деревьев решений

- Деревья имеют очень понятную интерпретацию. Даже проще, чем линейная регрессия!
- Существует мнение, что деревья решений отражают процесс принятия решений людьми лучше, чем подходы для решения задач регрессии и классификации, рассмотренные в предыдущих главах.
- Деревья можно наглядно отобразить графически и легко интерпретировать даже не специалистам (особенно, для небольших деревьев).
- Деревья могут легко обрабатывать качественные переменные и пропуски без необходимости создания фиктивных переменных.
- К сожалению, деревья обычно не дают такую же точность прогнозирования, как некоторые другие подходы для решения задач регрессии и классификации.

Нестабильность модели

Один новый пример

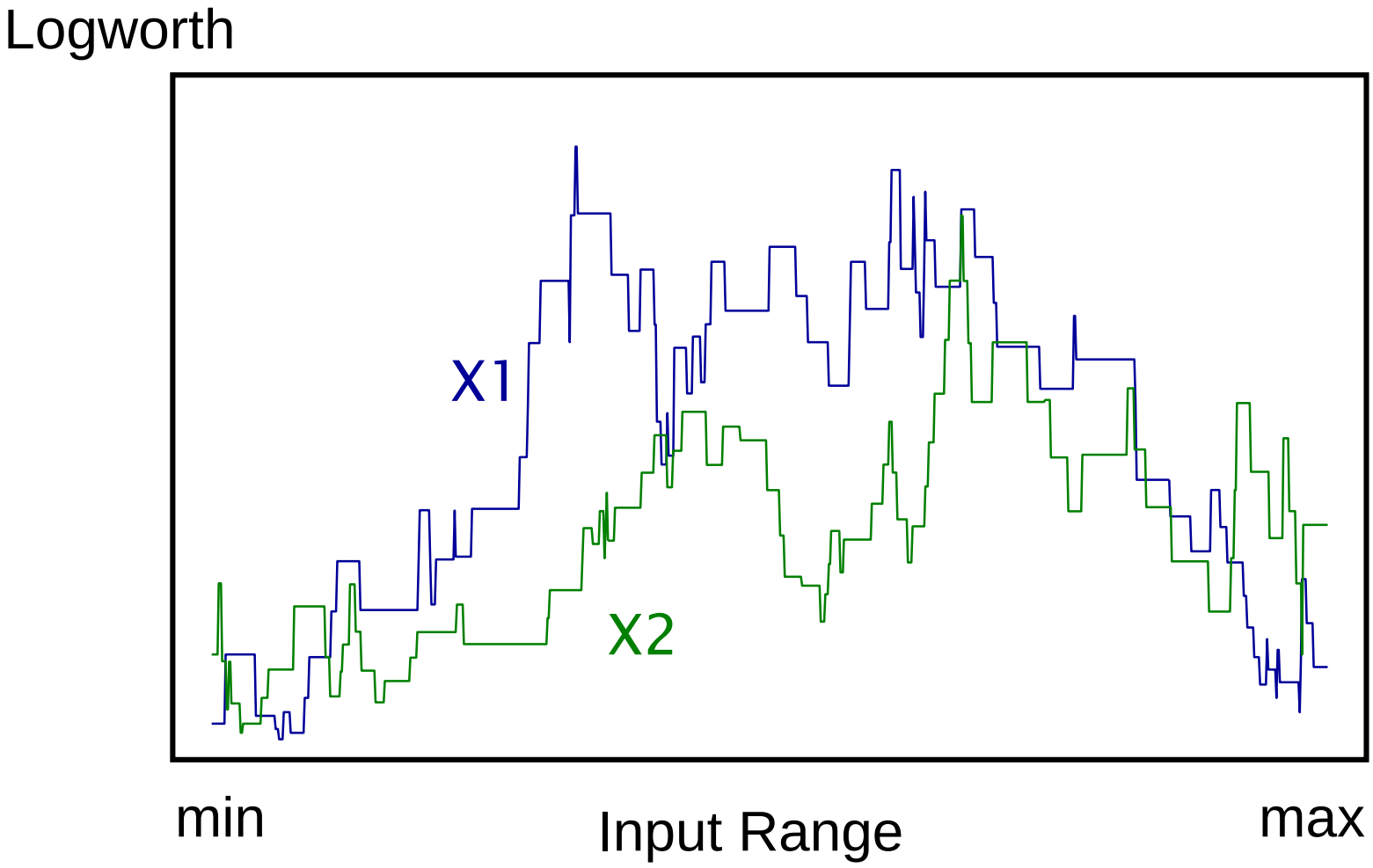




Точность = 81%

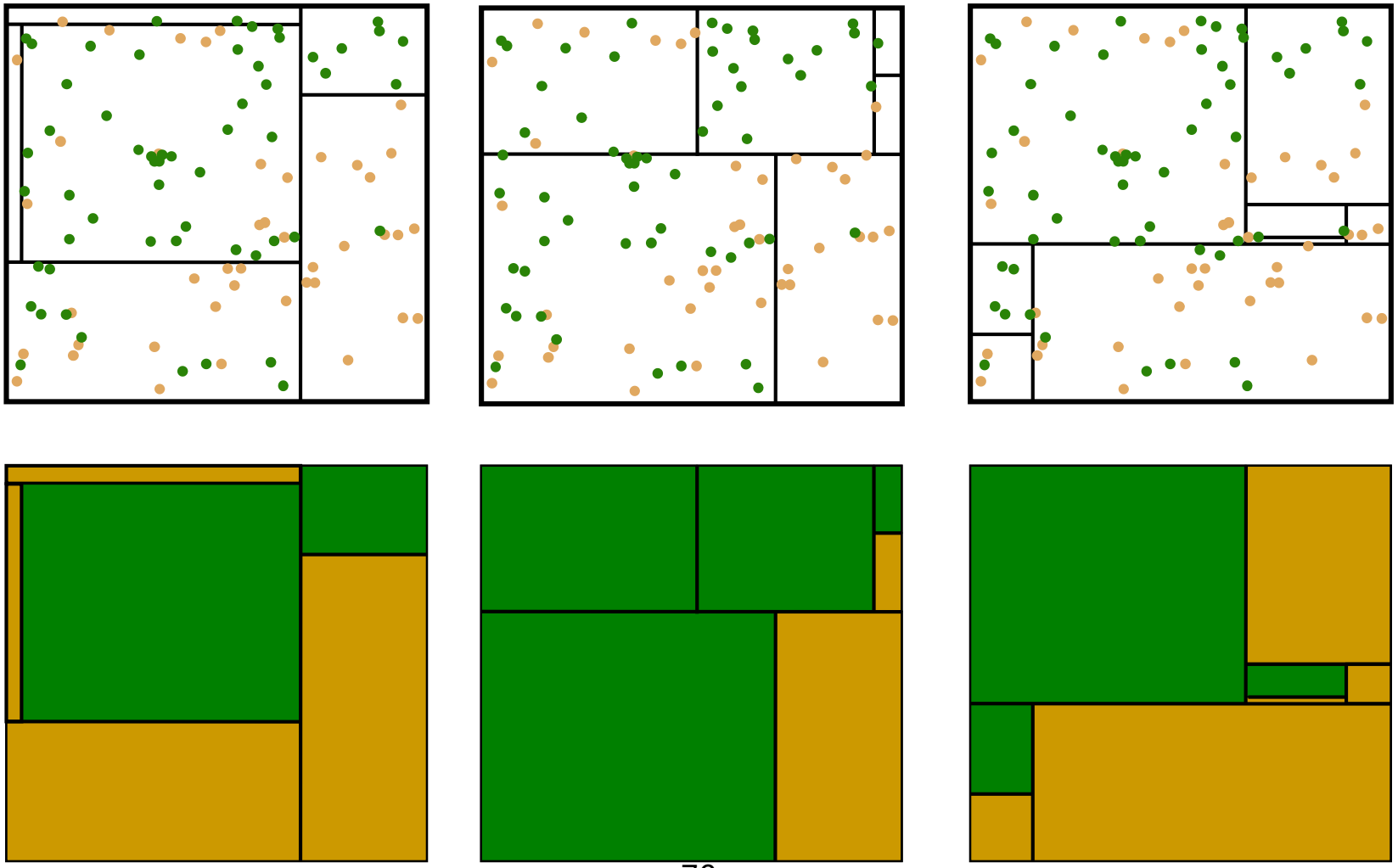
Точность = 80%

Альтернативные точки разбиения

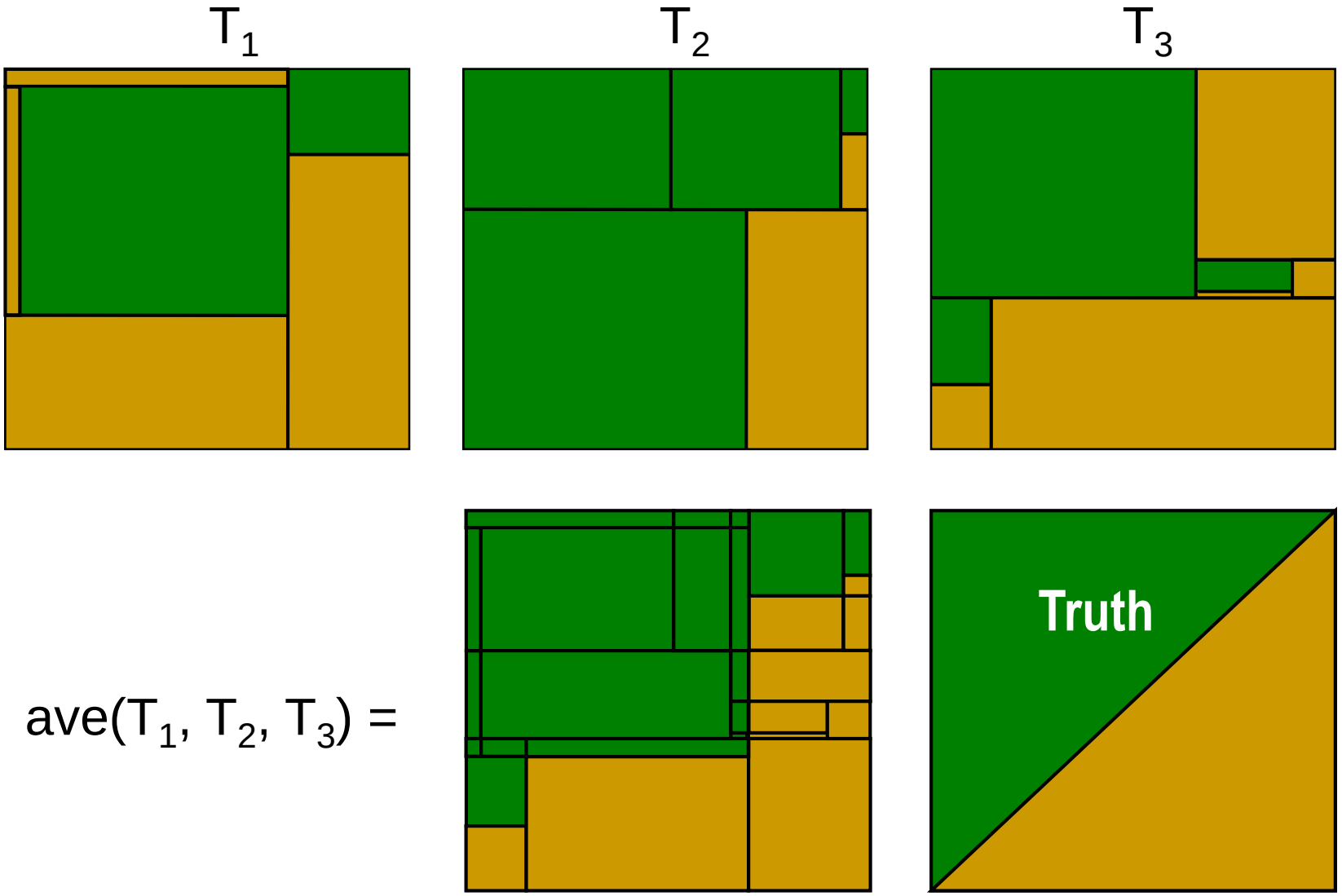


НЕСКОЛЬКО моделей построенных в разных условиях

T_1 T_2 T_3

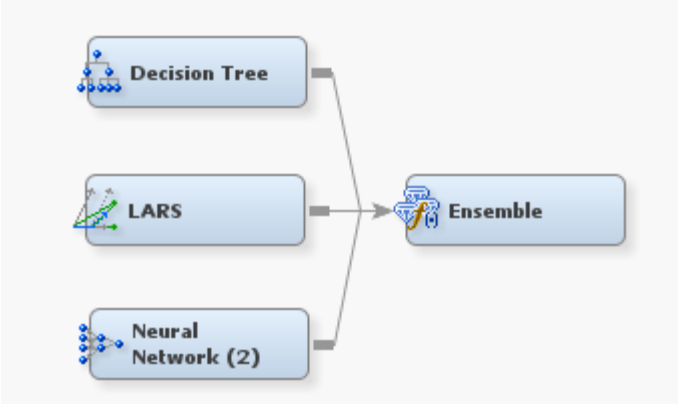


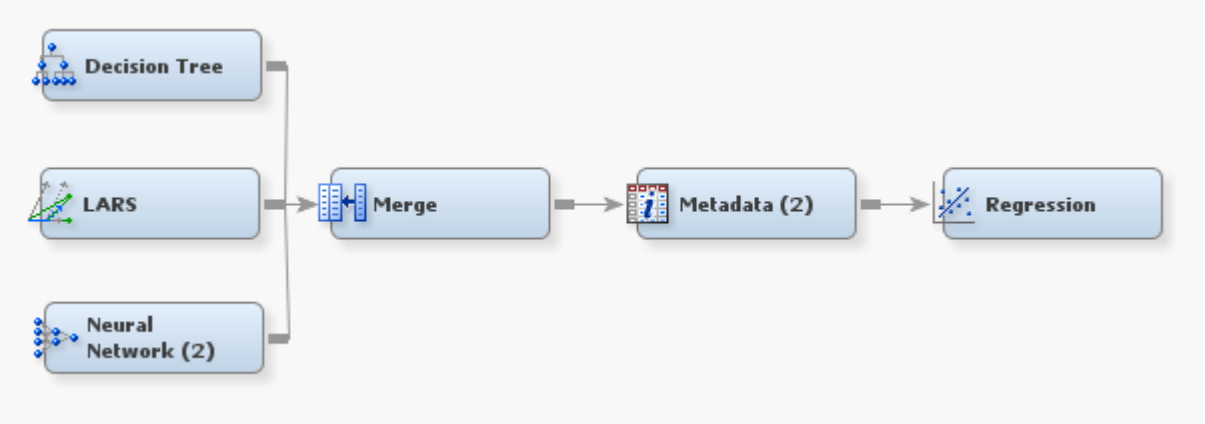
Ансамбль = Комбинация моделей



Типы ансамблей

- Простое «голосование»
  - Усреднение отклика, максимум отклика, «пропорция» голосов
- Взвешенное «голосование»
  - Строится новая простая модель (например регрессия или простая нейронная сеть) на основе откликов моделей ансамбля

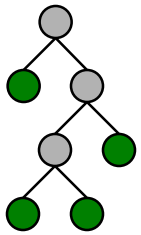


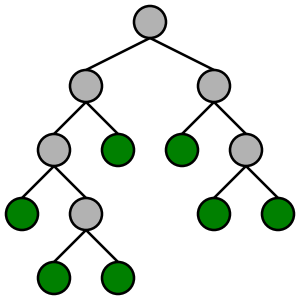


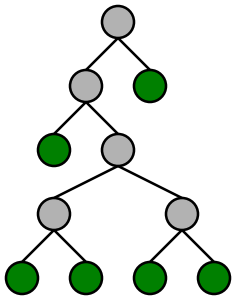
- «Комбинации»
  - Bagging - «усреднение» прогноза на разных выборках
  - Boosting– «усиление» прогноза на разных выборках

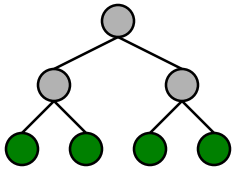
Bagging

	k=1	k=2	k=3	k=4	…
case	freq	freq	freq	freq	
1	1	0	3	1	
2	0	1	1	1	
3	2	0	0	2	
4	0	2	2	0	
5	2	2	0	1	
6	1	1	0	1	





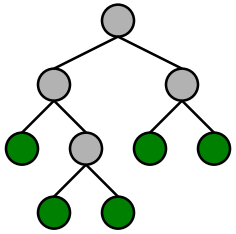


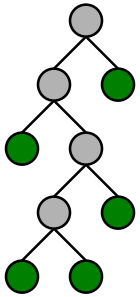


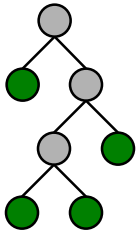
…

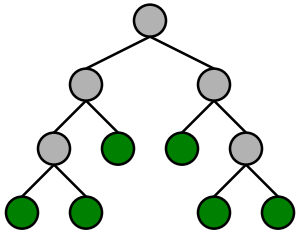
Arc-x4

	k=1		k=2		k=3		k=4 …
case	freq	m	freq	m	freq	m	freq
1	1	1	1.5	1	.5	2	.97
2	1	0	.75	0	.25	0	.06
3	1	1	1.5	2	4.25	3	4.69
4	1	0	.75	1	.5	1	.11
5	1	0	.75	0	.25	0	.06
6	1	0	.75	0	.25	1	.11





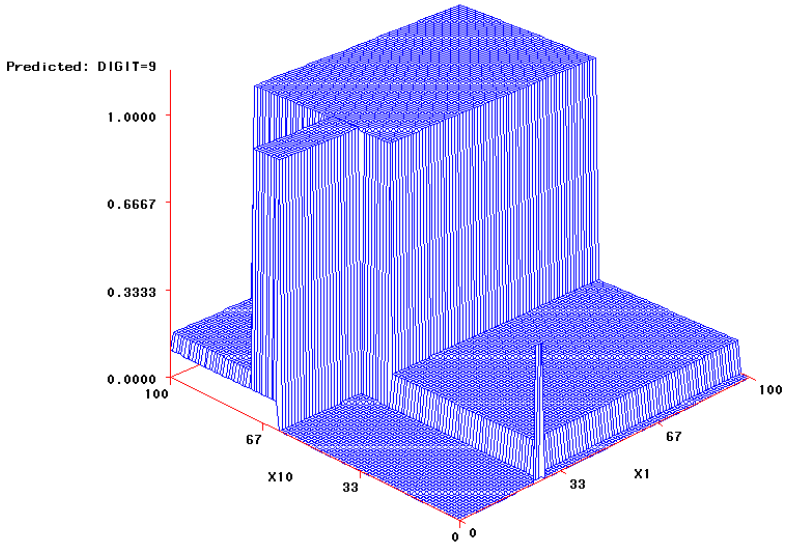


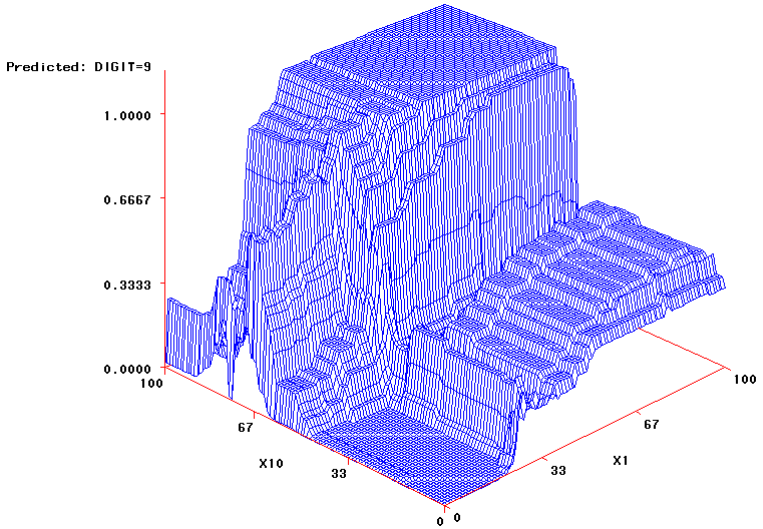


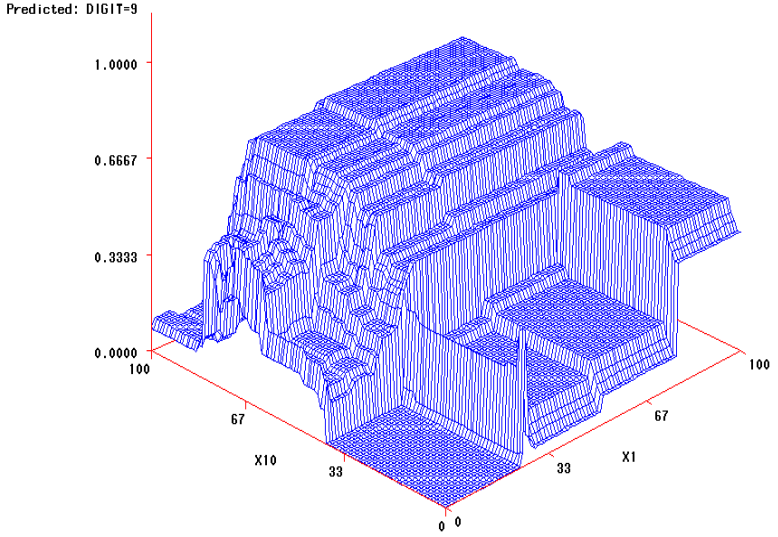
$$p(i) = \frac{1 + m(i)^4}{\sum \left(1 + m(i)^4\right)}$$

...

Обычное, Bagged и Boosted Деревья







Логистическая регрессия

Почему нельзя моделировать вероятность отклика p как непрерывный отклик с помощью линейно регрессии?

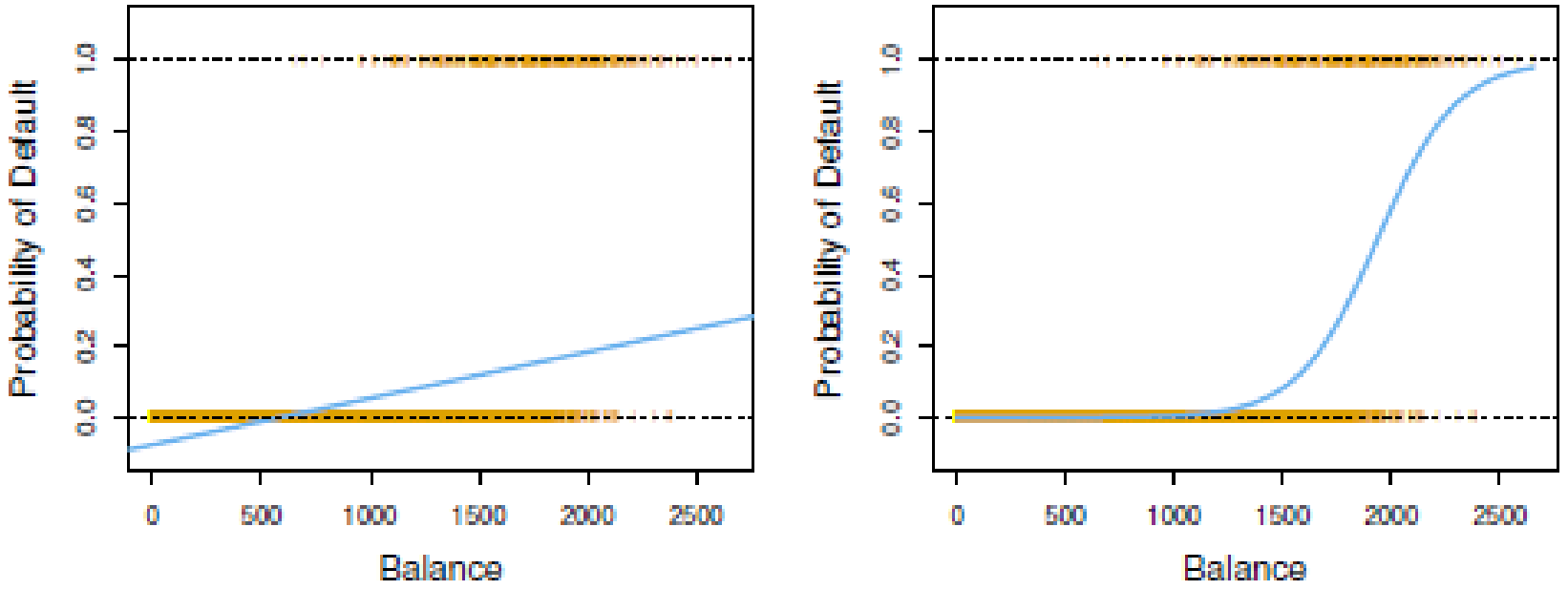
OLS Reg: $Y_i=\beta_0+\beta_1X_{1i}+\varepsilon_i$

- Если целевая переменная категориальная, как представить ее в виде числовой?
- Если целевая закодирована (1=Yes and 0=No) а результат модели 0.5 или 1.1 или -0.4, что это означает?
- Если переменная имеет только два значения (или несколько), имеет ли смысл требовать постоянства дисперсии или нормальности ошибок?

Linear Prob. Model: $p_i=\beta_0+\beta_1X_{1i}$

- Вероятность ограничена, а линейная функция принимает любые значения.
- Принимая во внимание ограниченность вероятности, можно ли предполагать линейную связь между X и *p*?
- Можно ли предполагать ошибку с постоянной дисперсией?
- Что такое наблюдаемая вероятность для конкретного наблюдения? 0 и 1?

Линейная vs логистическая регрессия



Логистическая регрессия гарантирует, что оценка $p(X)$ находится в диапазоне между 0 и 1.

Логистическая регрессия

Уравнение логистической регрессии:

вероятность

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}$$

параметр

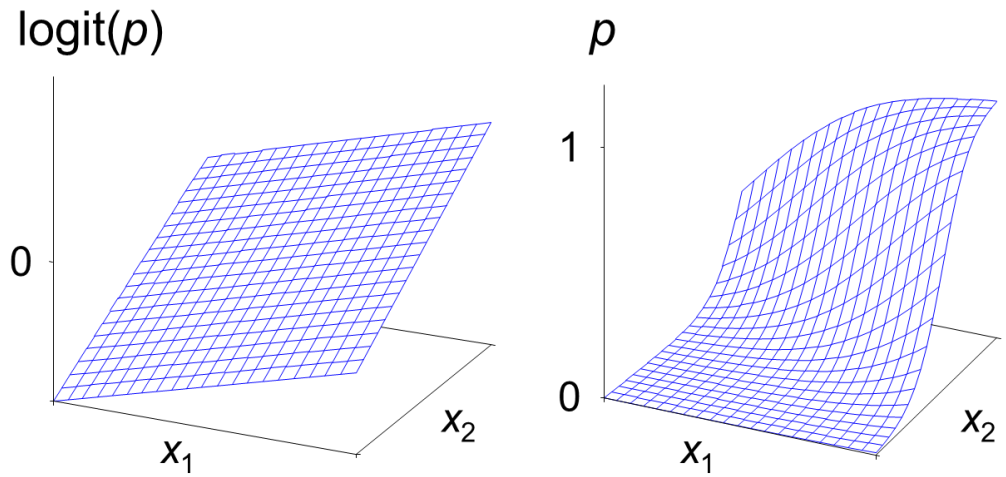
предиктор

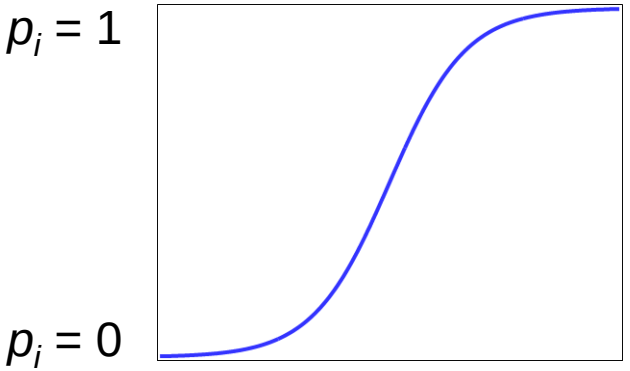
Функция связи (логит) и обратная ей (логистическая):

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \eta$$

$$\Leftrightarrow \quad p_i = \frac{1}{1+e^{-\eta}}$$

Основное предположение линейной логистической регрессии (линейная зависимость логита от предикторов):





меньше ← η → больше

Ограничивает значение отклика

Максимальное правдоподобие

Максимальное правдоподобие для оценки параметров.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

Это правдоподобие дает вероятность наблюдаемых нулей и единиц в данных. Мы выбираем параметры модели, чтобы максимизировать вероятность наблюдаемых данных.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Отношение шансов

- Показывает как изменится отношение шансов при изменении i-ой переменной на 1 unit (равно exp от коэф.)

$$\text{logit}(\hat{p}) = \log(odds) = \beta_0 + \beta_i * x_i + \sum_{j \neq i} \beta_j * x_j$$

$$odds = \exp(\beta_0 + \beta_i * x_i + \sum_{j \neq i} \beta_j * x_j)$$

$$\text{logit}(\hat{p}') = \log(odds) = \beta_0 + \beta_i * (x_i + 1) + \sum_{j \neq i} \beta_j * x_j$$

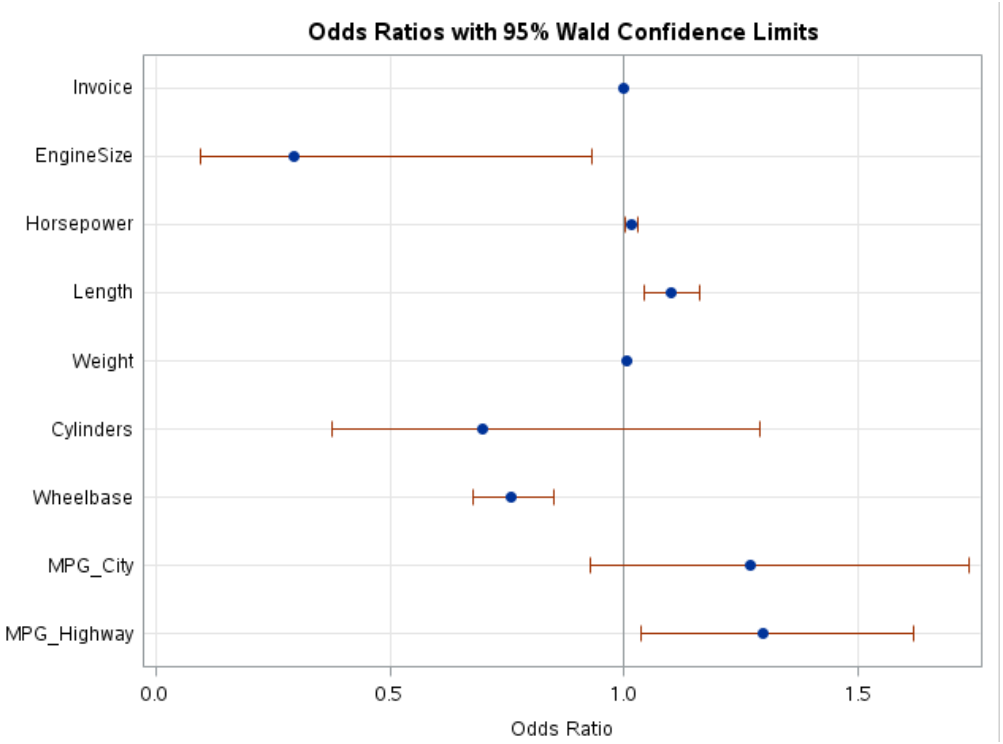
$$odds' = \exp(\beta_0 + \beta_i * (x_i + 1) + \sum_{j \neq i} \beta_j * x_j)$$

$$odds\ ratio = odds' / odds = \exp(\beta_i)$$

Больше 1 – отношение шансов увеличивается, если меньше, то уменьшается

Отношение шансов (пример)

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Invoice	1.000	1.000	1.000
EngineSize	0.295	0.094	0.931
Horsepower	1.016	1.003	1.029
Length	1.100	1.044	1.160
Weight	1.005	1.004	1.007
Cylinders	0.696	0.376	1.289
Wheelbase	0.757	0.676	0.849
MPG_City	1.270	0.929	1.736
MPG_Highway	1.295	1.036	1.618



Категориальные предикторы

- Схемы кодировки:
  - Effect coding (относительно «среднего»)

CLASS	Value	Label	1	2
IncLevel	1	Low Income	1	0
	2	Medium Income	0	1
	3	High Income	-1	-1

  - Reference coding (относительно «базового»)

CLASS	Value	Label	1	2
IncLevel	1	Low Income	1	0
	2	Medium Income	0	1
	3	High Income	0	0

Effect Coding: Пример

$$\text{logit}(p)=\beta_0+\beta_1*D_{\text{Low income}}+\beta_2*D_{\text{Medium income}}$$

β_0= Средний логит по всем категориям

β_1= Разница между логитом для Low income и средним логитом

β_2= разница между Medium income и средним логитом

Analysis of Maximum Likelihood Estimates						
Parameter		**DF**	**Estimate**	**Standard Error**	**Wald Chi-Square**	**Pr > ChiSq**
Intercept		1	-0.5363	0.1015	27.9143	<.0001
IncLevel	**1**	1	-0.2259	0.1481	2.3247	0.1273
IncLevel	**2**	1	-0.2200	0.1447	2.3111	0.1285

Reference Coding: Пример

$$\text{logit}(p)=\beta_0+\beta_1*D_{\text{Low income}}+\beta_2*D_{\text{Medium income}}$$

β_0= Логит для High

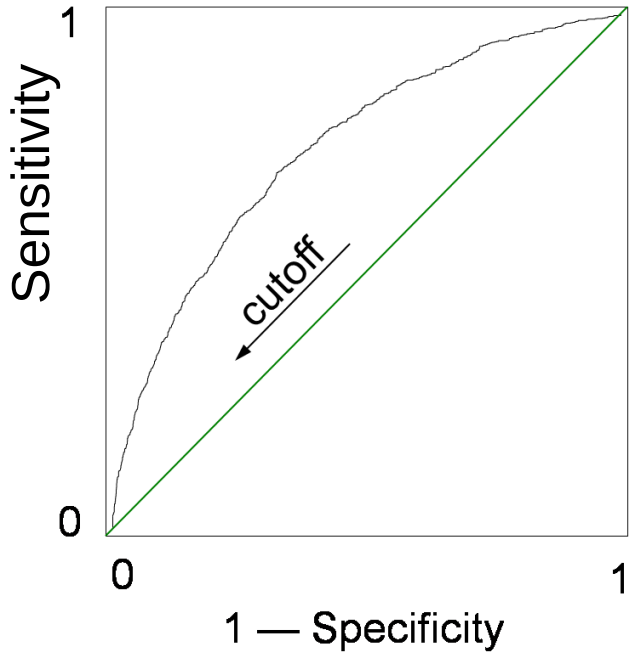
β_1= Разница логитов между Low и High

β_2= Разница логитов между Medium и High

Analysis of Maximum Likelihood Estimates						
Parameter		**DF**	**Estimate**	**Standard Error**	**Wald Chi-Square**	**Pr > ChiSq**
Intercept		1	-0.0904	0.1608	0.3159	0.5741
IncLevel	**1**	1	-0.6717	0.2465	7.4242	0.0064
IncLevel	**2**	1	-0.6659	0.2404	7.6722	0.0056

Оценка модели

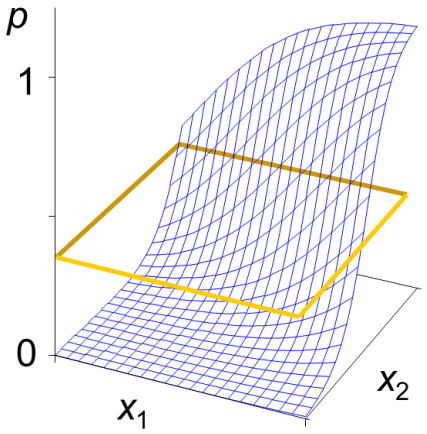
Actual Class	Predicted Class: 0	Predicted Class: 1	
0	**True Negative**	**False Positive**	Actual Negative
1	**False Negative**	**True Positive**	Actual Positive
	Predicted Negative	Predicted Positive	

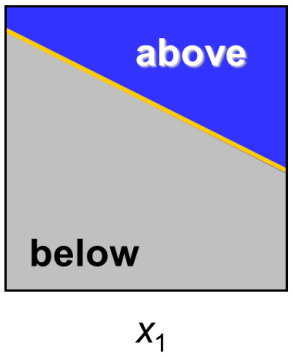


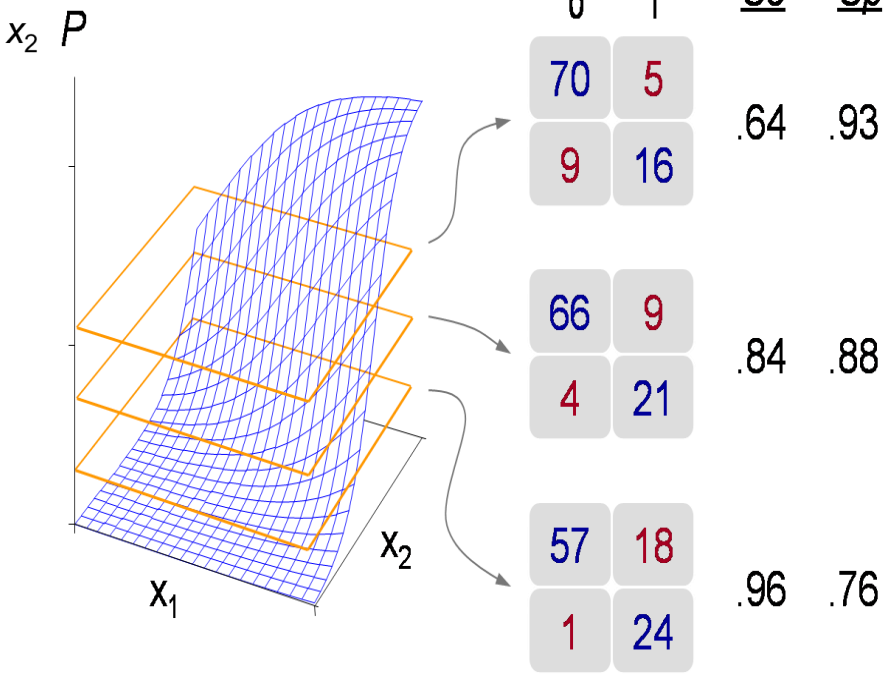
SENSITIVITY (true positive rate (TPR), hit rate, recall)
TPR = TP / (TP+FN)

SPECIFICITY (SPC) (true negative rate (TNR))
SPC = TN / (FP + TN)

Оценка модели







Матрица выигрыша-проигрыша:

Actual Class \ Decision	0	1
0	δ_{TN}	δ_{FP}
1	δ_{FN}	δ_{TP}

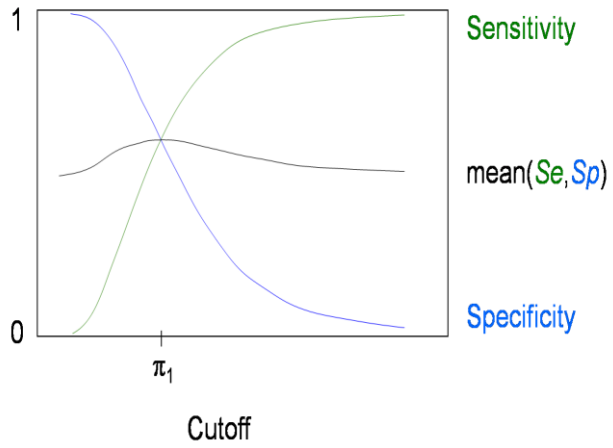
Bayes Rule:

Decision 1 if

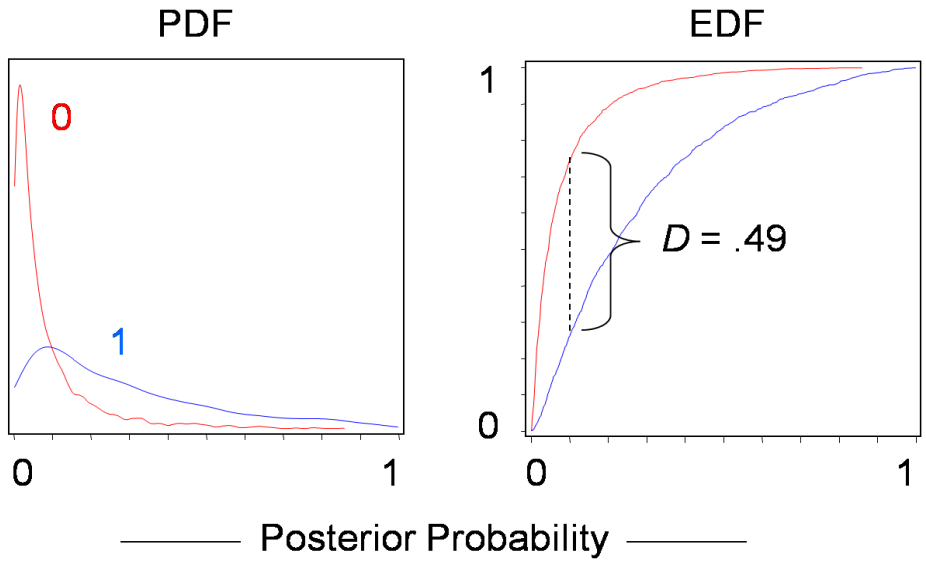
$$P > \frac{1}{1+\left(\frac{\delta_{TP}-\delta_{FN}}{\delta_{TN}-\delta_{FP}}\right)}$$

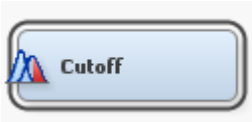
Выбор порога

Усредненная (иногда взвешенная) чувствительность и специфичность



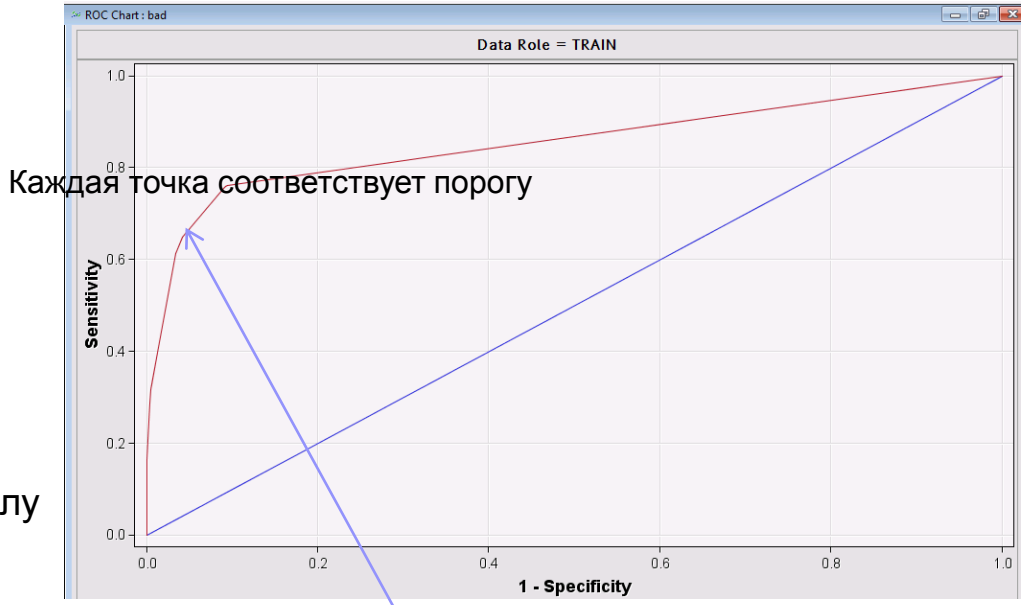
Критерий Колмогорова-Смирнова





ROC кривая и AUC

- Процедура построения:
  - Сортируем набор по убыванию спрогнозированной оценки (вероятности положительного отклика)
  - Идем порогом отсечения по отсортированному набору
  - Для каждого положения порога считаем:
    1. отношение числа положительных примеров «слева» от порога к числу всех положительных примеров – detection rate
    2. отношение числа отрицательных примеров «слева» от порога к числу всех отрицательных примеров – false positive
  - Ставим точку на графике



Каждая точка соответствует порогу

Ошибки 1 и второго рода

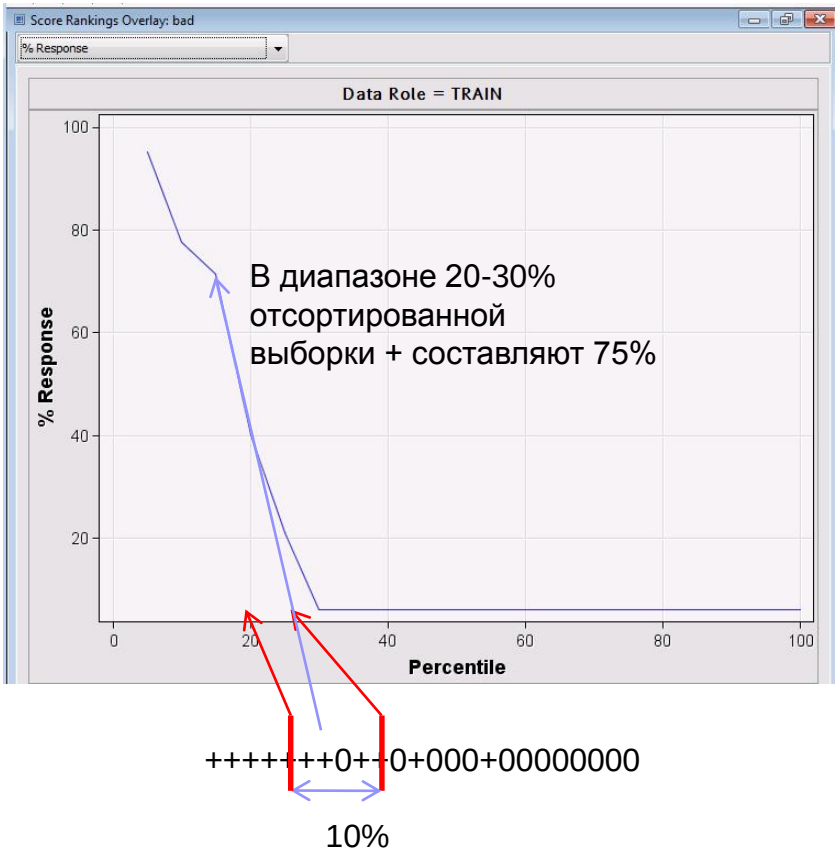
+++++++0++0+000+00000000

+? 0?

Оценка на основе согласованности всевозможных пар наблюдений (правильной упорядоченности наблюдений в паре), принадлежащих разным классам.

Графические средства сравнения моделей: Response (отклик)

- Процедура построения:
  - Сортируем (например, слева направо) набор по убыванию спрогнозированной оценки (вероятности положительного отклика)
  - Идем порогом отсечения по отсортированному набору (слева направо) с некоторым диапазоном (как правило кратно 5%)
  - Для каждого положения диапазона считаем отношение числа положительных примеров к числу всех примеров внутри диапазона
  - Ставим точку на графике
- Агрегированный отклик (cumulative response):
  - Диапазон всегда с 0



Графические средства сравнения моделей: Lift (подъем)

- Процедура построения:
  - Сортируем (например, слева направо) набор по убыванию спрогнозированной оценки (вероятности положительного отклика)
  - Идем порогом отсечения по отсортированному набору (слева направо) с некоторым диапазоном (как правило кратно 5%)
  - Для каждого положения диапазона считаем отношение числа положительных примеров к числу положительных примеров, которые могли бы быть выбраны «случайно» - без модели
  - Ставим точку на графике
- Агрегированный подъем(cumulative lift):
  - Диапазон всегда с 0