



Data Mining in Action

Лекция 4. Решающие деревья и ансамбли



Индустриальный партнер курса



jet.su

На прошлой лекции

- Линейная классификация
- Стохастический градиентный спуск
- Регуляризация
- Линейная регрессия

Напоминание: часто используемые методы

- Линейные модели
- Решающие деревья
- Ансамбли решающих деревьев

План лекции

1. Решающие деревья

2. Ансамбли деревьев

3. Общие идеи построения ансамблей

4. Извлечение и простые преобразования признаков

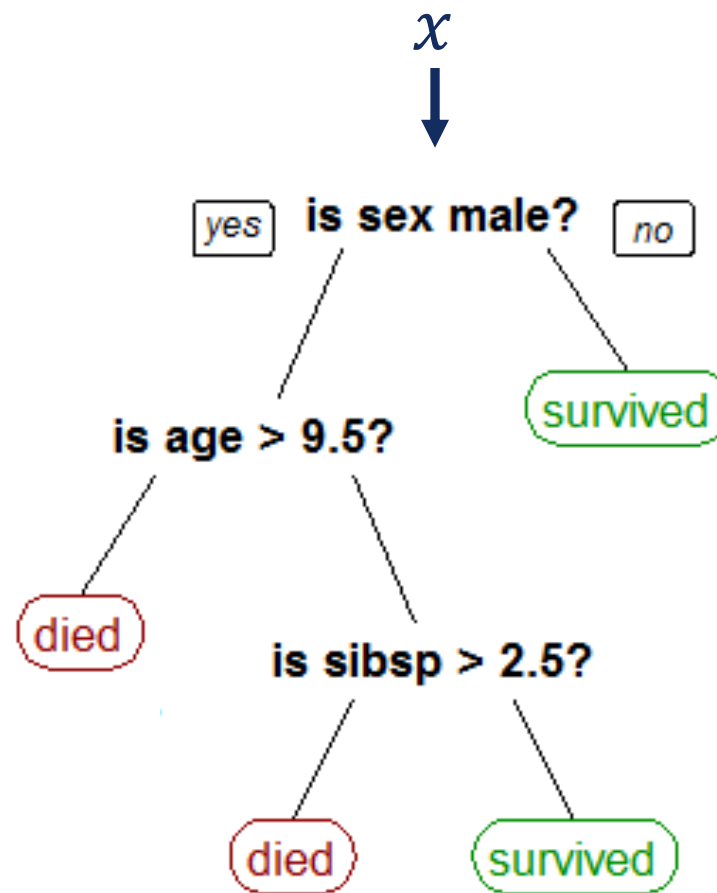
5. Отбор признаков

1. Решающие деревья

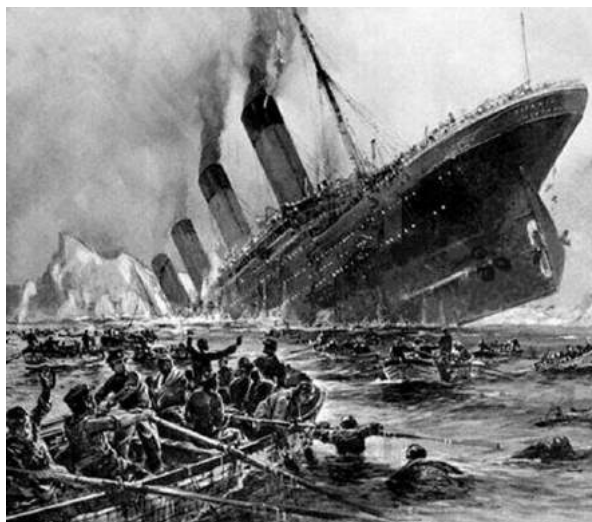
План

1. Что такое решающие деревья
2. Решающие деревья в классификации и регрессии
3. Как строить решающие деревья
4. Дополнительные темы

Решающее дерево



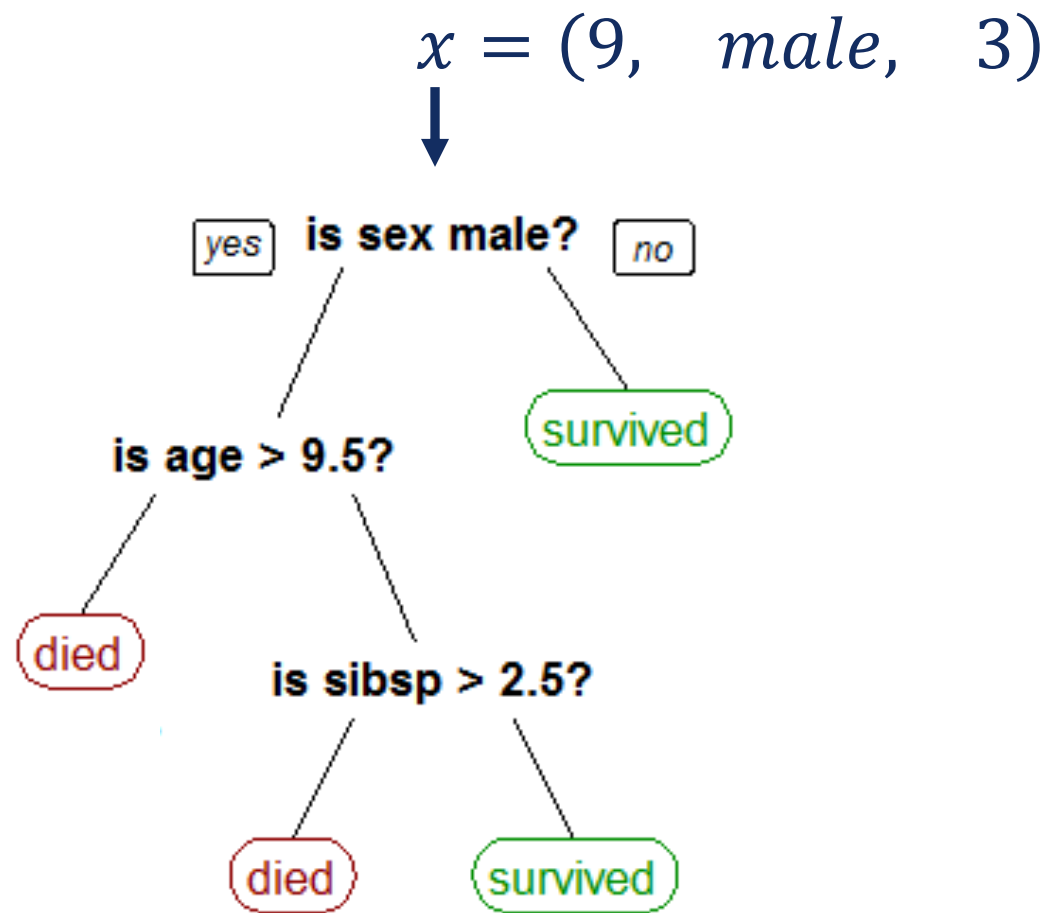
Датасет



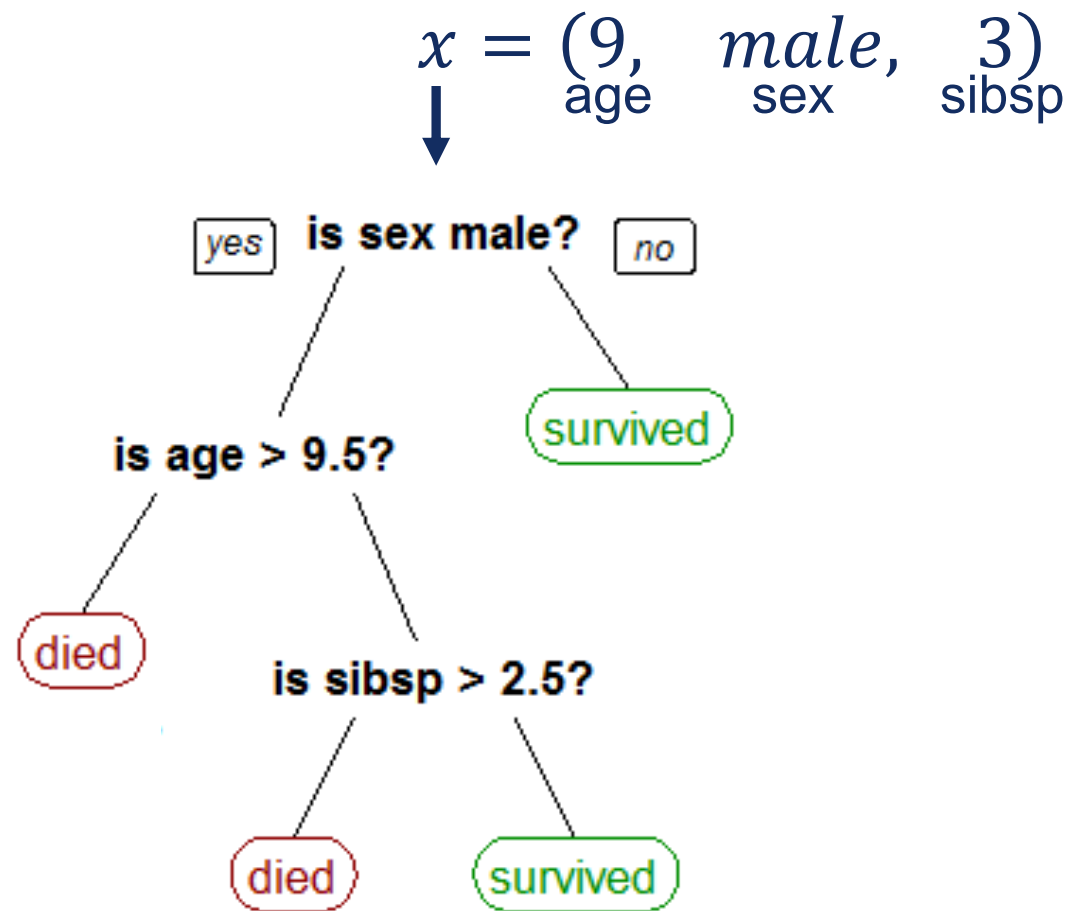
«Titanic Dataset» - список пассажиров Титаника, для которых даны возраст, пол, количество членов семьи на борту и другие признаки.

Целевые значения: выжил пассажир или нет (задача классификации)

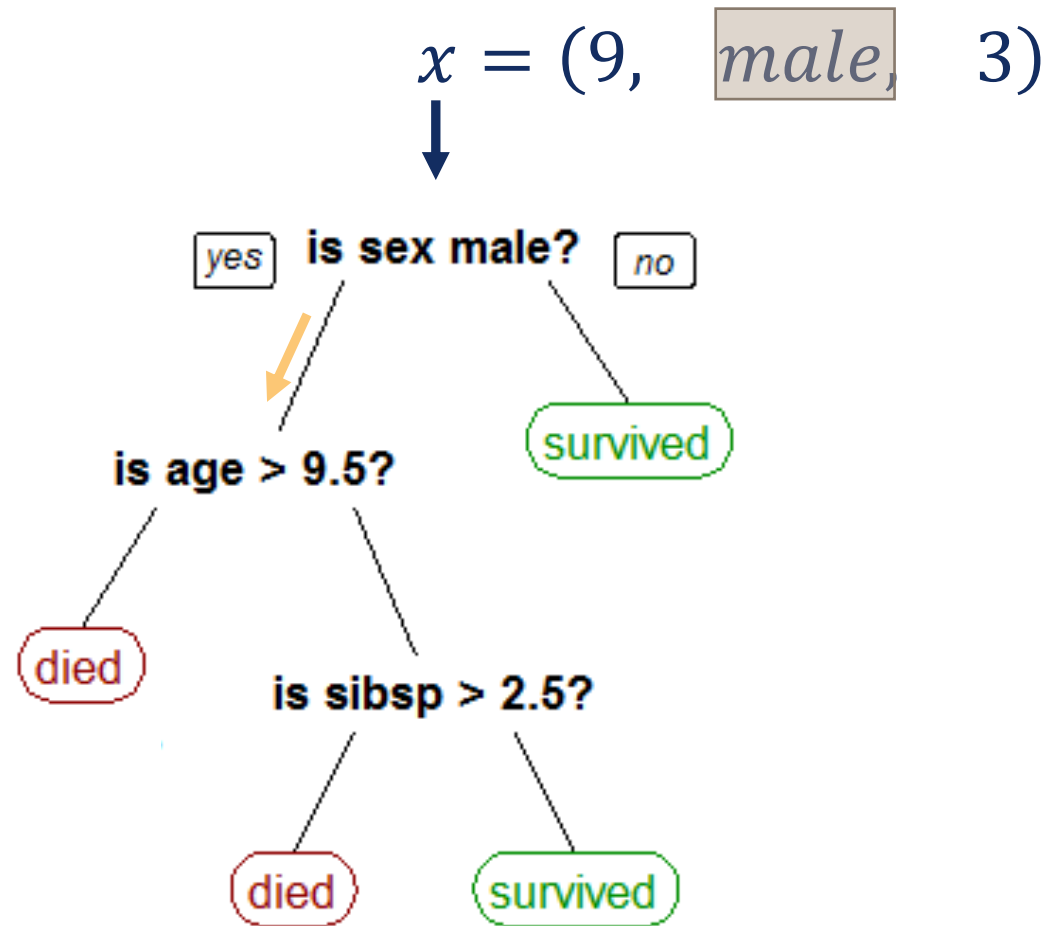
Решающее дерево



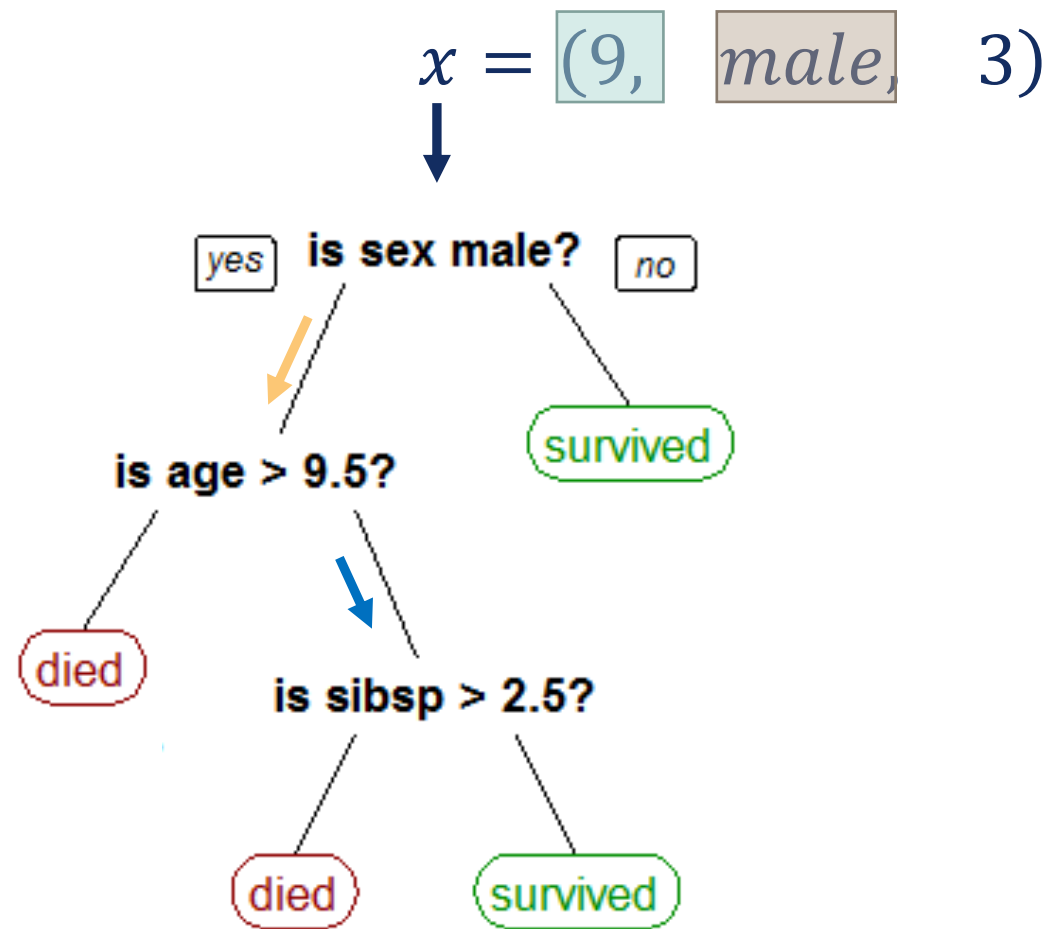
Решающее дерево



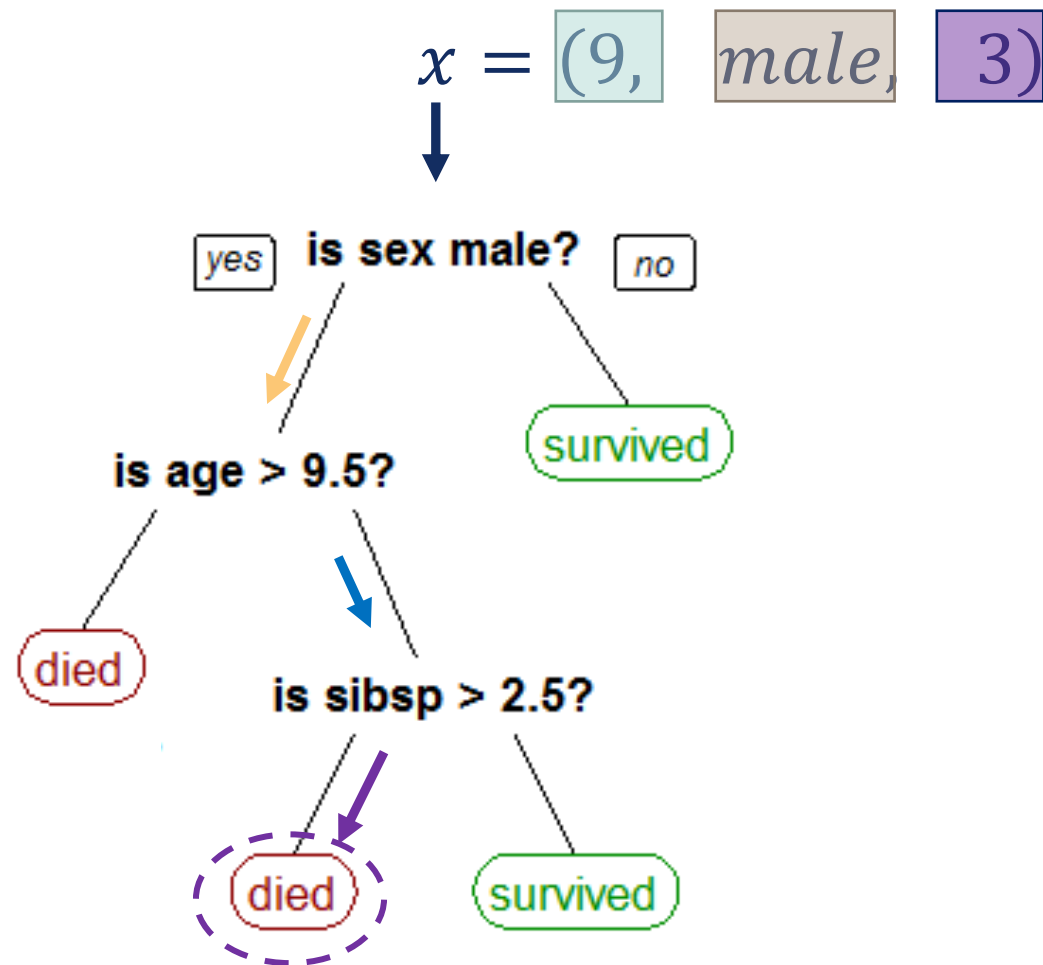
Решающее дерево



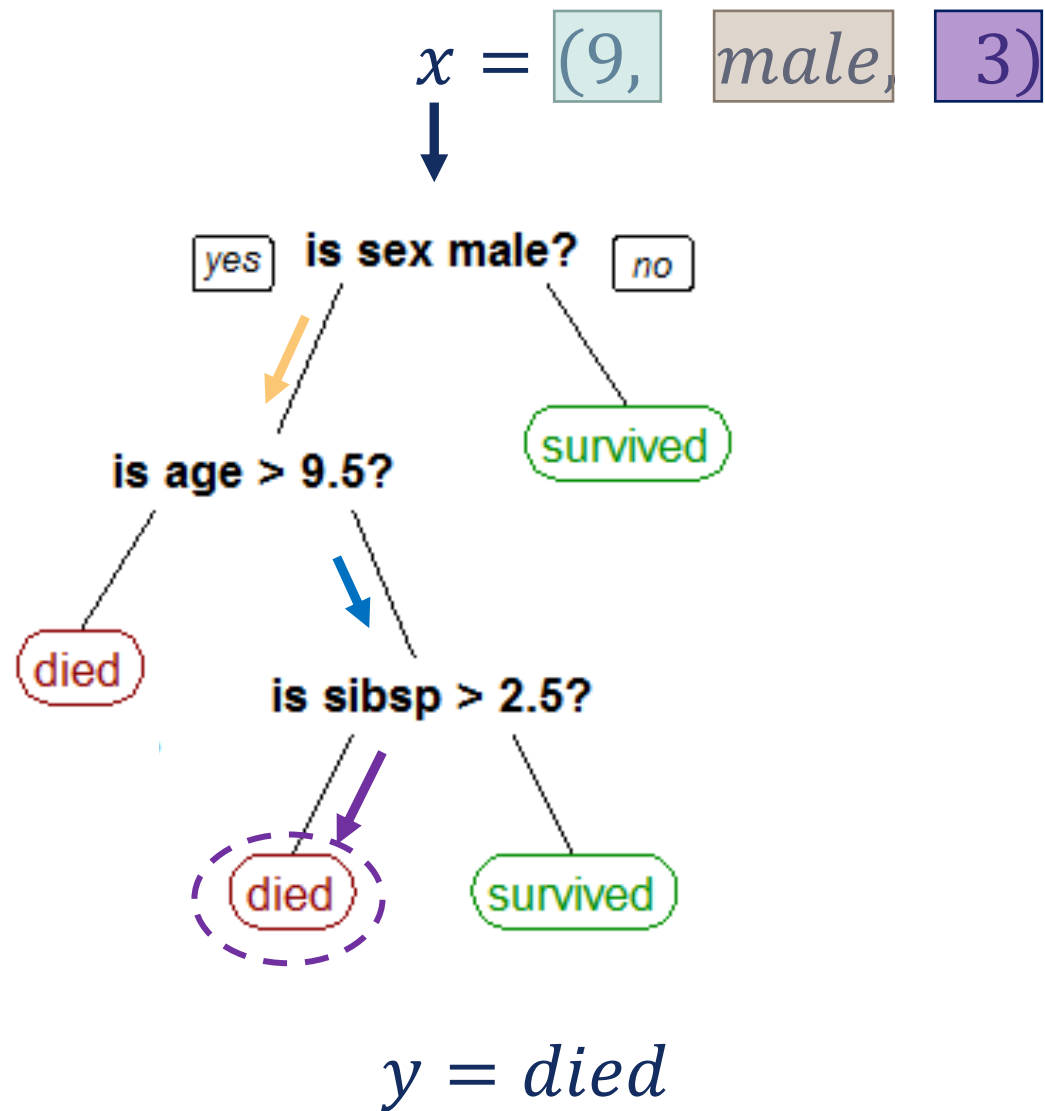
Решающее дерево



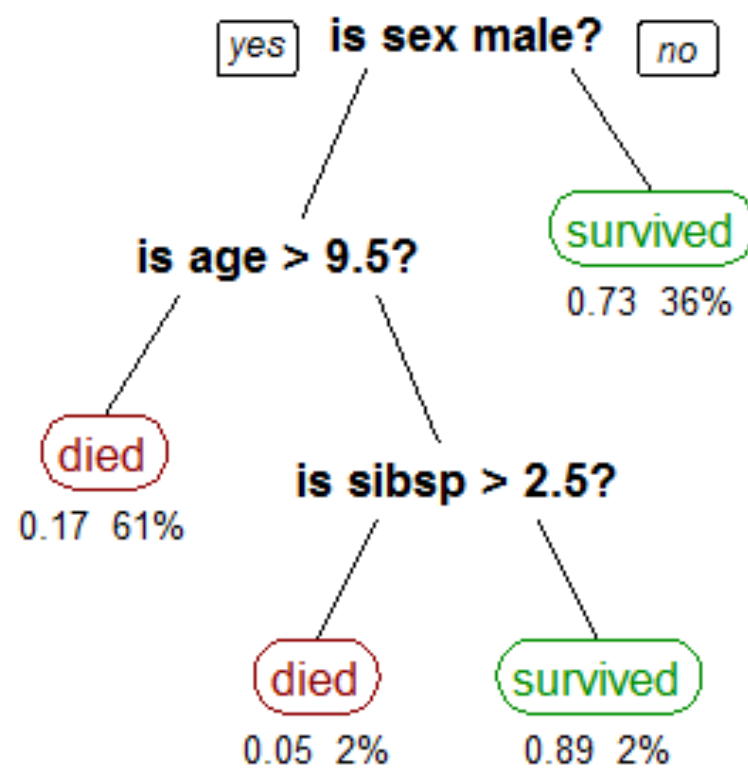
Решающее дерево



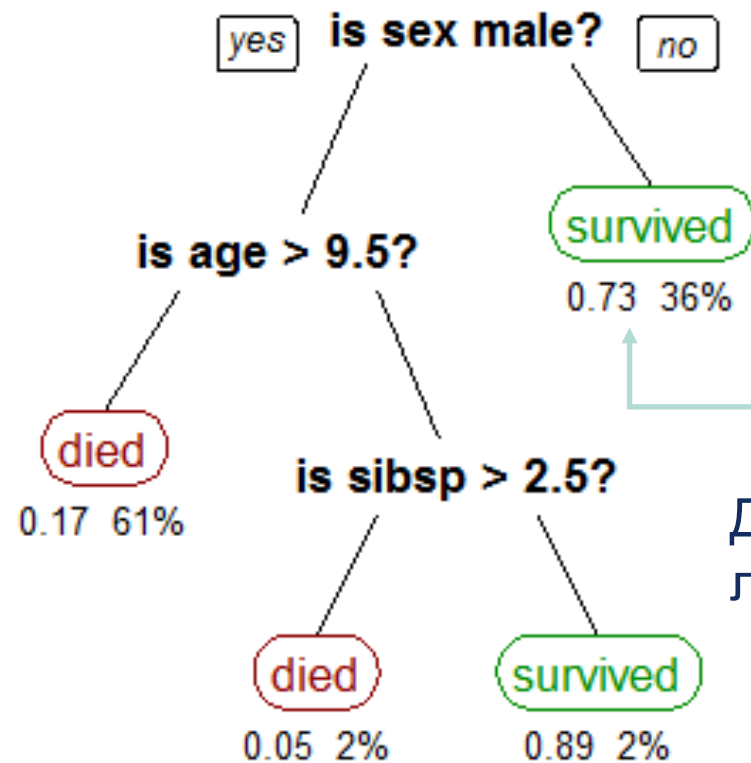
Решающее дерево



Решающее дерево: классификация



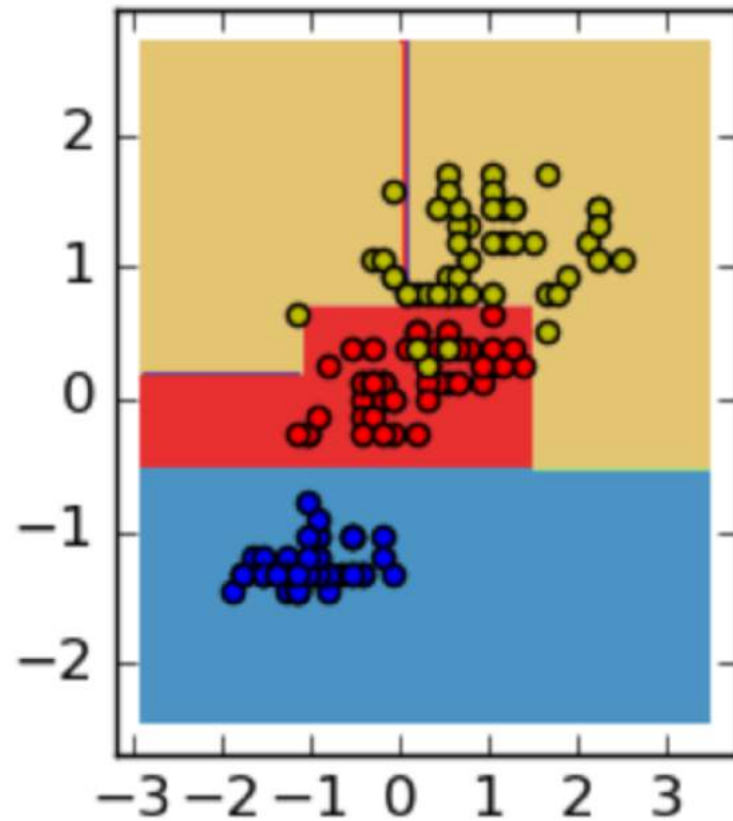
Решающее дерево: классификация



Доля выживших среди попавших в лист

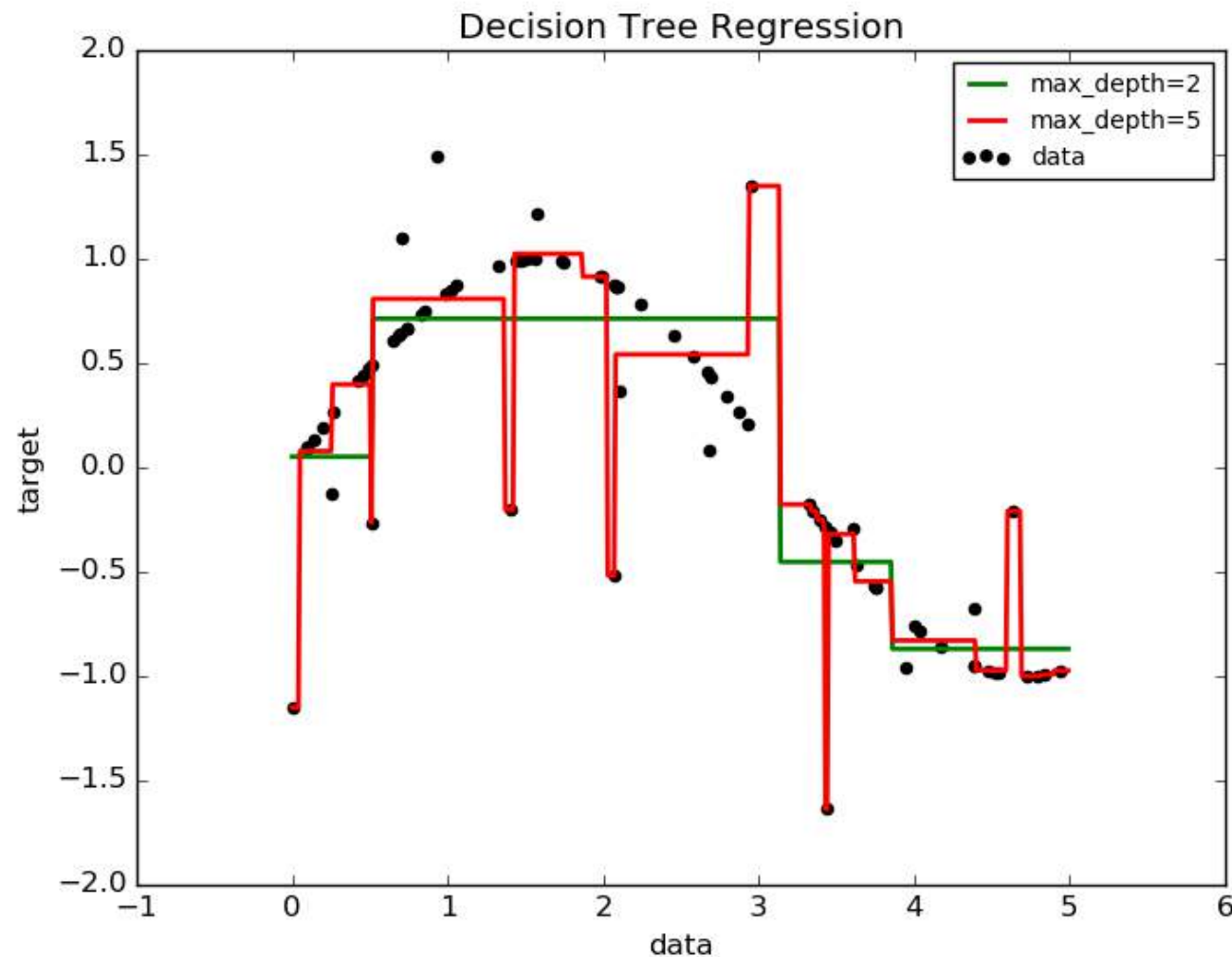
В каждом листе дерево отвечает преобладающим классом

Решающее
дерево:
классификация



Пример: 3 класса и
2 признака

Решающее дерево: регрессия



Пример:
восстановление
зависимости y от x
с помощью
решающих
деревьев глубины
2 и глубины 5

В каждом листе
дерево отвечает
некоторой
константой

Рекурсивное построение

Строим
разбиение
выборки по
значению одного
из признаков

$$x^{(j)} < t$$

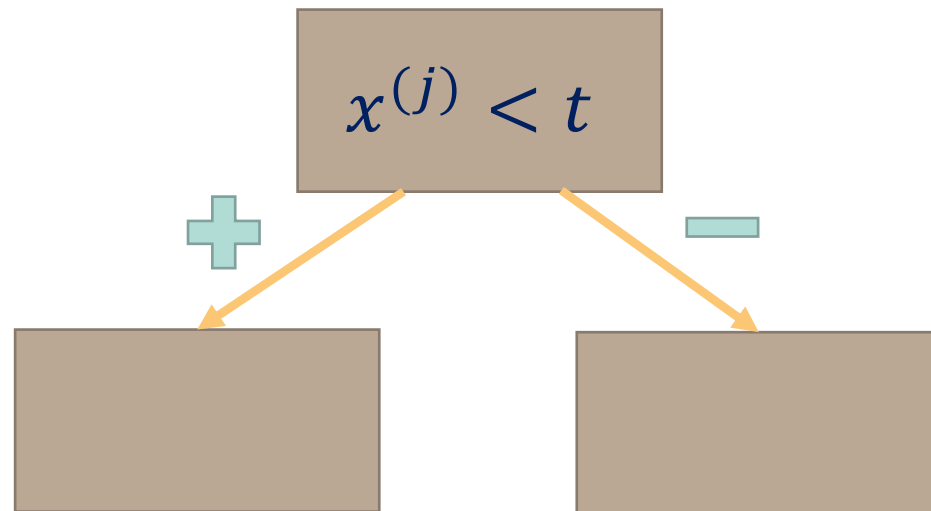
Рекурсивное построение

Строим
разбиение
выборки по
значению одного
из признаков

$$x^{(j)} < t$$

Фактически нужно
только выбрать j и
 t наилучшим
образом

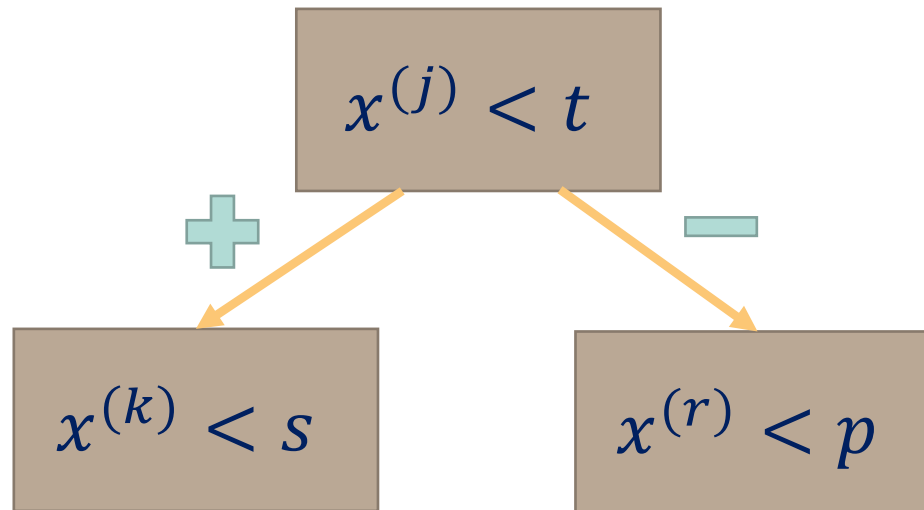
Рекурсивное построение



Выборка
делится по
этому
условию на
две части

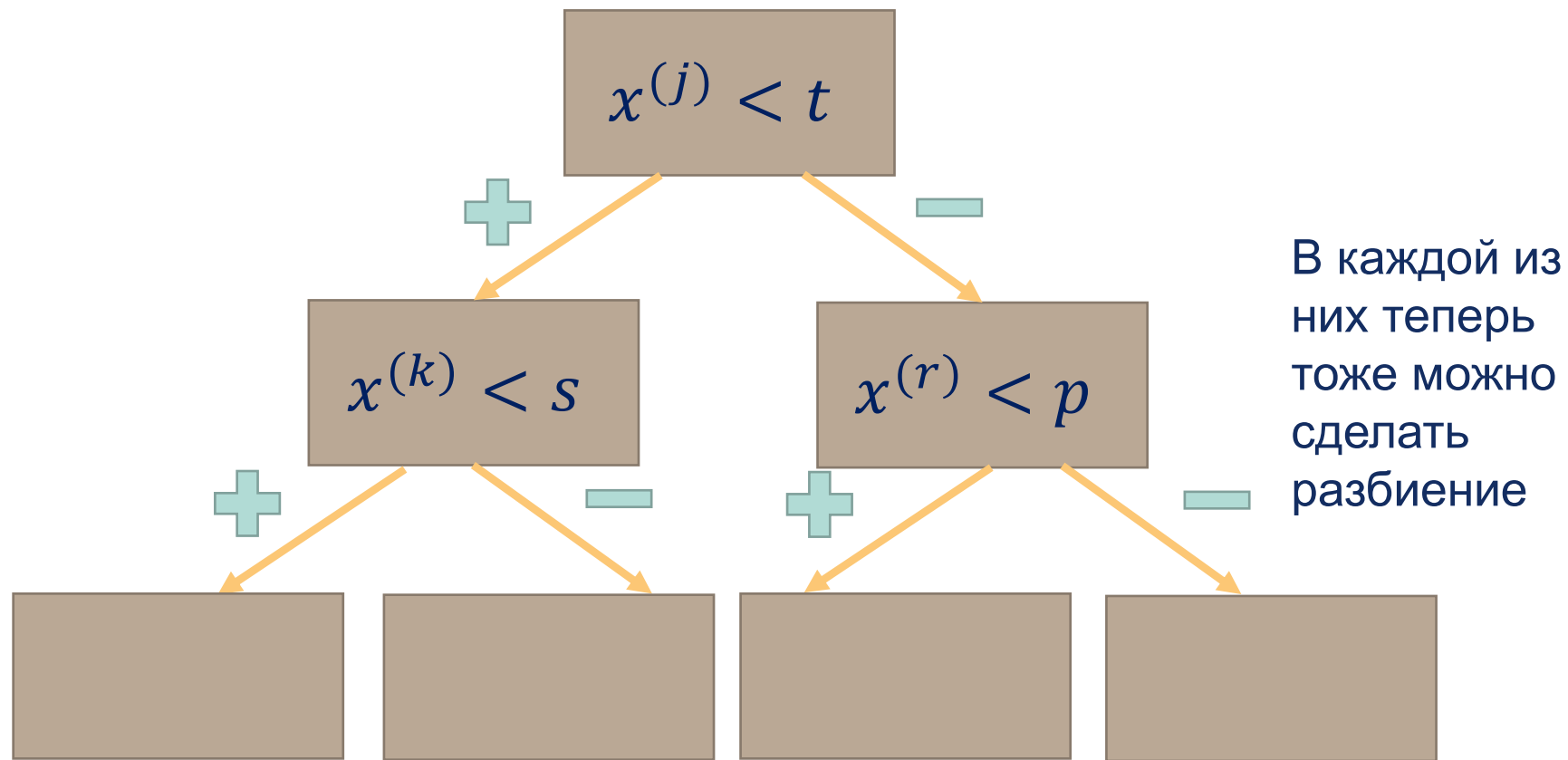
Рекурсивное построение

В каждой из них теперь тоже можно сделать разбиение

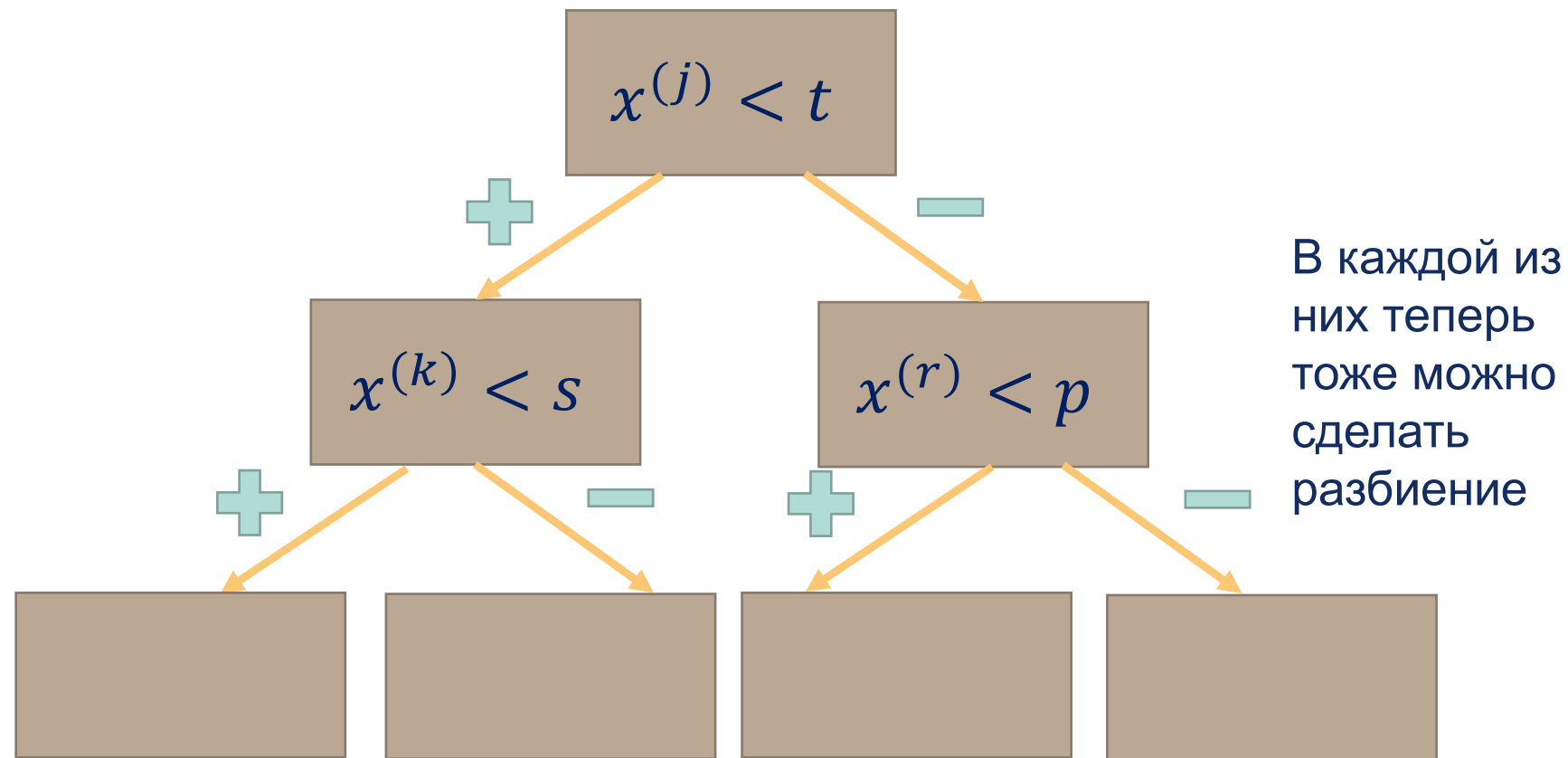


Выборка делится по этому условию на две части

Рекурсивное построение

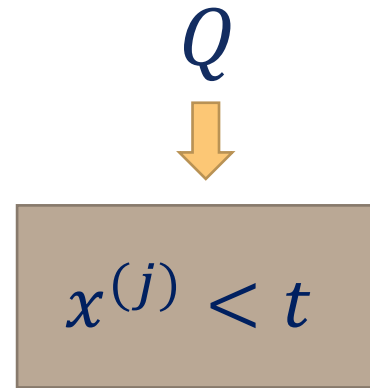


Рекурсивное построение

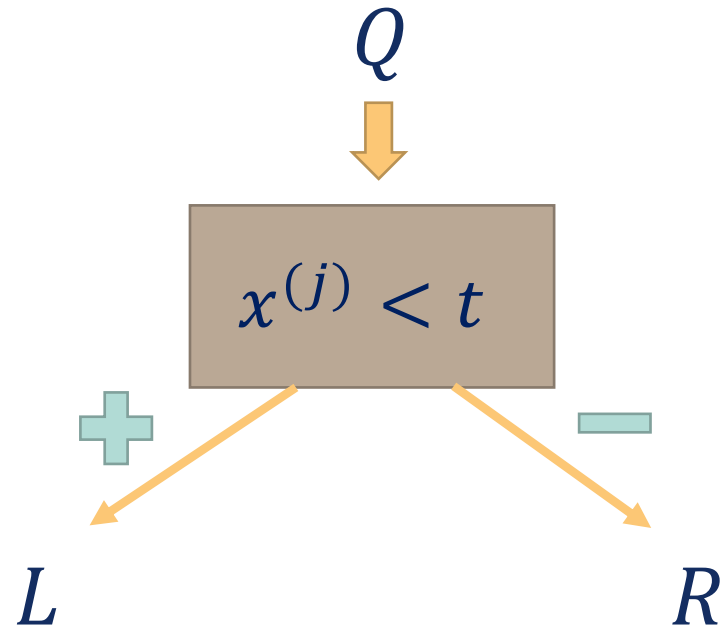


Процесс можно продолжать в тех узлах, в которые попадает достаточно много объектов

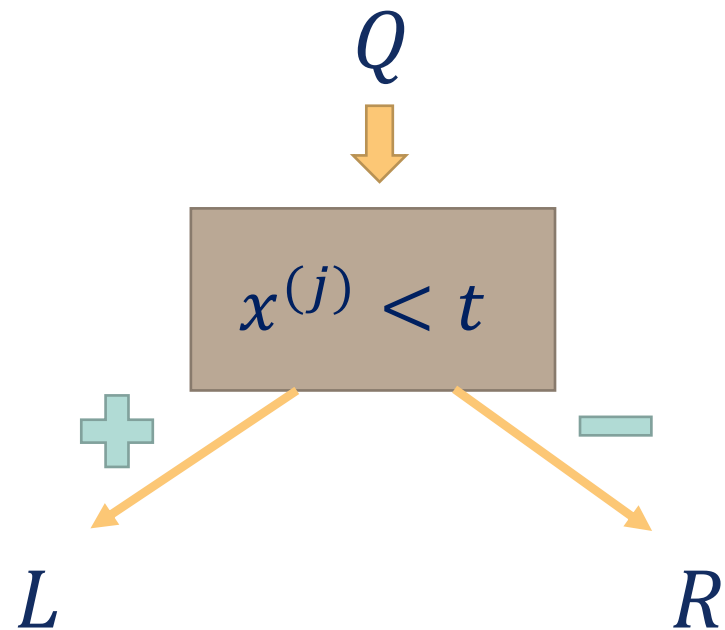
Выбор разбиения



Выбор разбиения

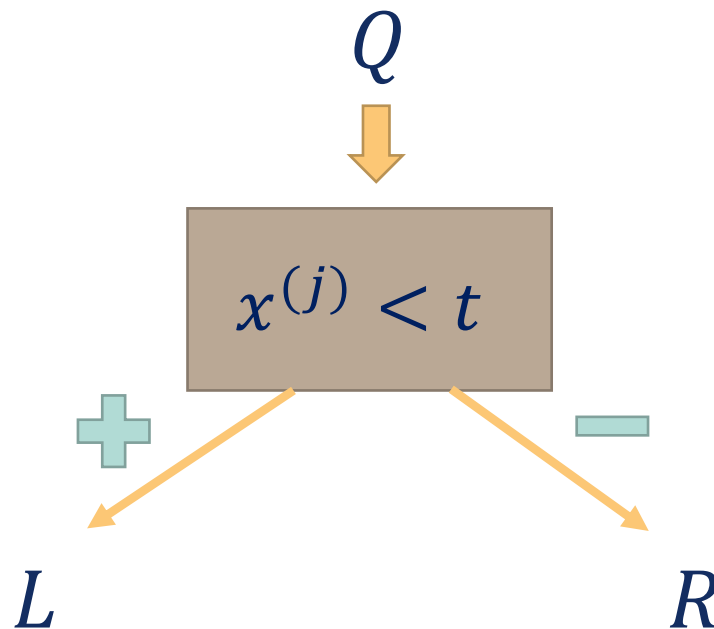


Выбор разбиения



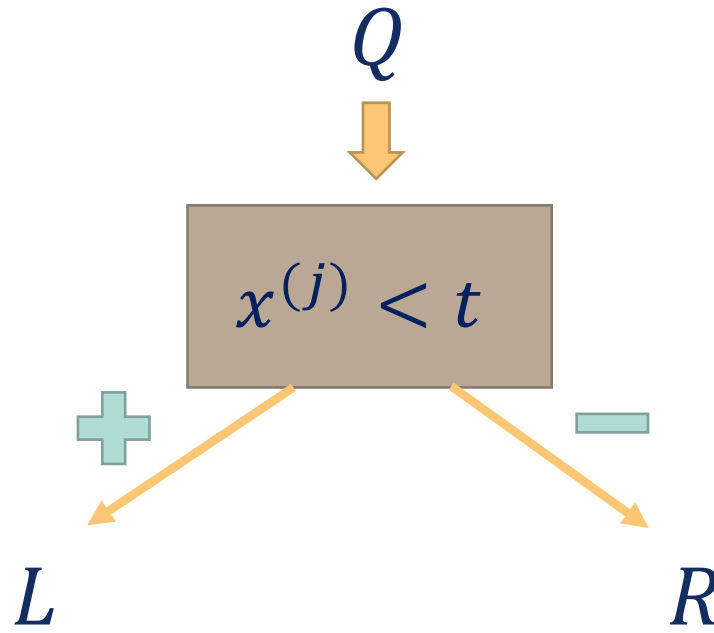
$$G(j, t) = \frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R)$$

Выбор разбиения



$$G(j, t) = \frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R) \rightarrow \min_{j, t}$$

Выбор разбиения



$$G(j, t) = \frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R) \rightarrow \min_{j, t}$$

$H(R)$ - мера «неоднородности» множества R

Критерии построения разбиений

$H(R)$ — мера «неоднородности» множества R

Пусть мы решаем задачу классификации на 2 класса,

p_0, p_1 — доли объектов классов 0 и 1 в R

1) Misclassification criteria: $H(R) = 1 - \max\{p_0, p_1\}$

2) Entropy criteria: $H(R) = -p_0 \ln p_0 - p_1 \ln p_1$

3) Gini criteria: $H(R) = 1 - p_0^2 - p_1^2 = 2p_0p_1$

Критерии построения разбиений

$H(R)$ – мера «неоднородности» множества R

Пусть мы решаем задачу классификации на K классов,

p_1, \dots, p_K – доли объектов классов $1, \dots, K$ в R

1) Misclassification criteria: $H(R) = 1 - p_{max}$

2) Entropy criteria:

$$H(R) = - \sum_{k=1}^K p_k \ln p_k$$

3) Gini criteria:

$$H(R) = \sum_{k=1}^K p_k (1 - p_k)$$

Критерии построения разбиений

$H(R)$ – мера «неоднородности» множества R

Чтобы решать задачу регрессии, достаточно взять среднеквадратичную ошибку в качестве $H(R)$:

$$H(R) = \frac{1}{|R|} \sum_{x_i \in R} (y_i - \bar{y})^2$$

Критерии построения разбиений

$H(R)$ – мера «неоднородности» множества R

Чтобы решать задачу регрессии, достаточно взять среднеквадратичную ошибку в качестве $H(R)$:

$$H(R) = \frac{1}{|R|} \sum_{x_i \in R} (y_i - \bar{y})^2$$

$$\bar{y} = \frac{1}{|R|} \sum_{x_i \in R} y_i$$

Prunning

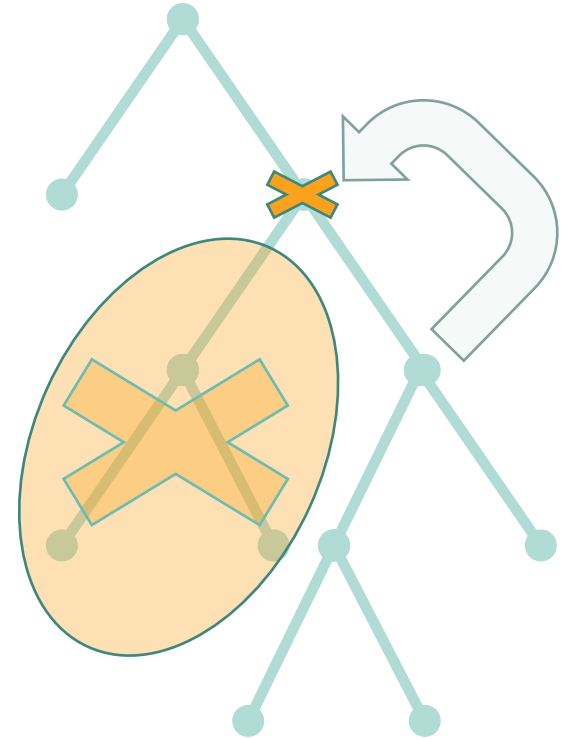
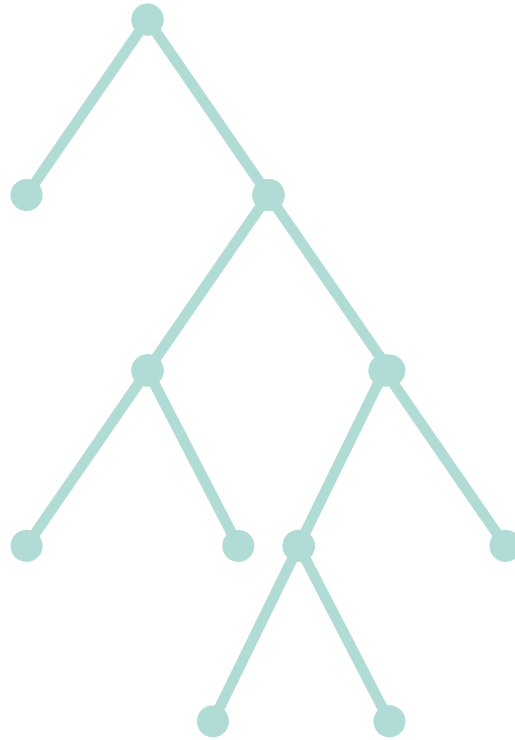
Pre-prunning:

- Ограничиваем рост дерева до того как оно построено
- Если в какой-то момент информативность признаков в разбиении меньше порога – не разбиваем вершину

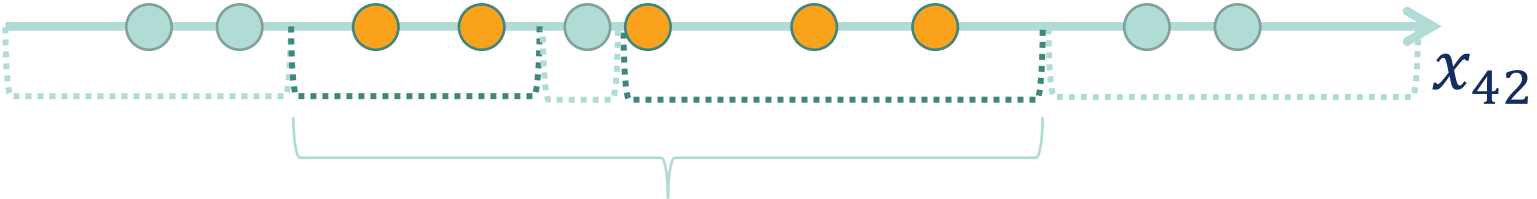
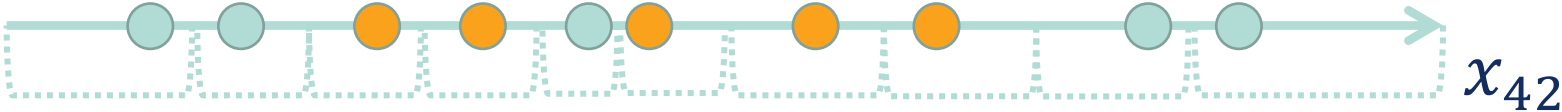
Post-prunning:

- Упрощаем дерево после того как дерево построено

Post-pruning



Бинаризация



Вариации алгоритма построения

- C4.5
- C5.0
- CART

Итог

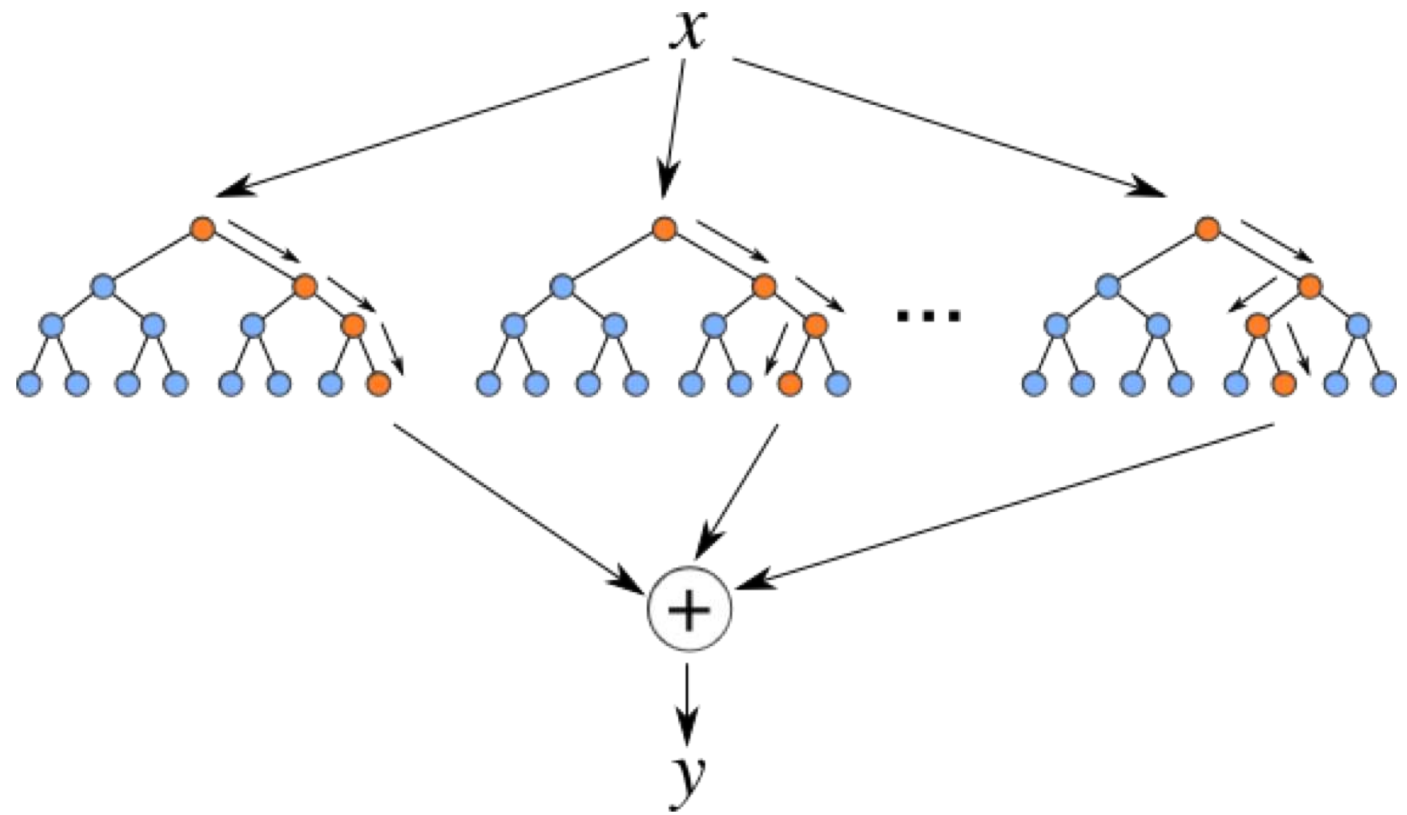
1. Что такое решающие деревья
2. Решающие деревья в классификации и регрессии
3. Как строить решающие деревья
4. Дополнительные темы

2. Ансамбли решающих деревьев

План

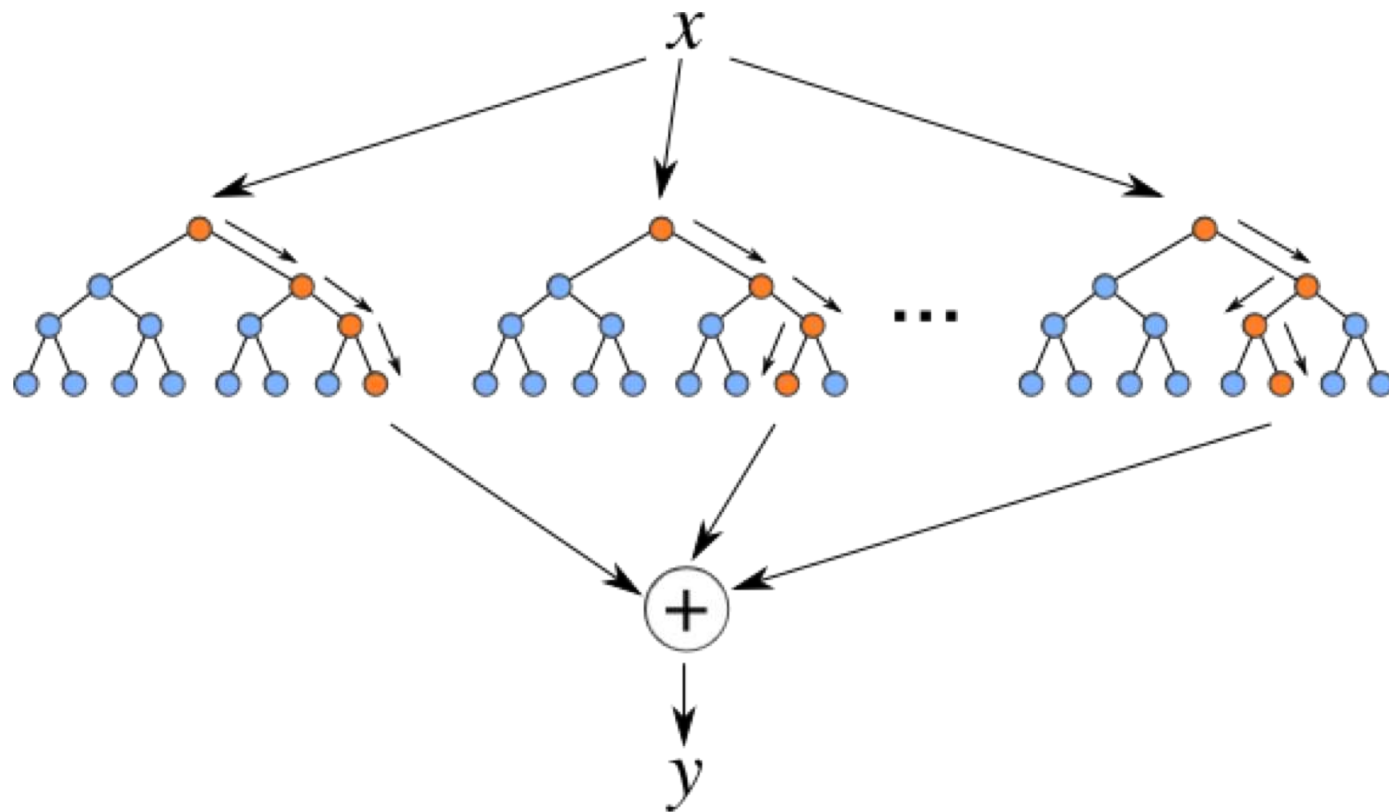
1. Random Forest
2. Идея Gradient Boosted Decision Trees (GBDT)
3. Библиотеки
4. Подробное обсуждение GBDT

Random Forest



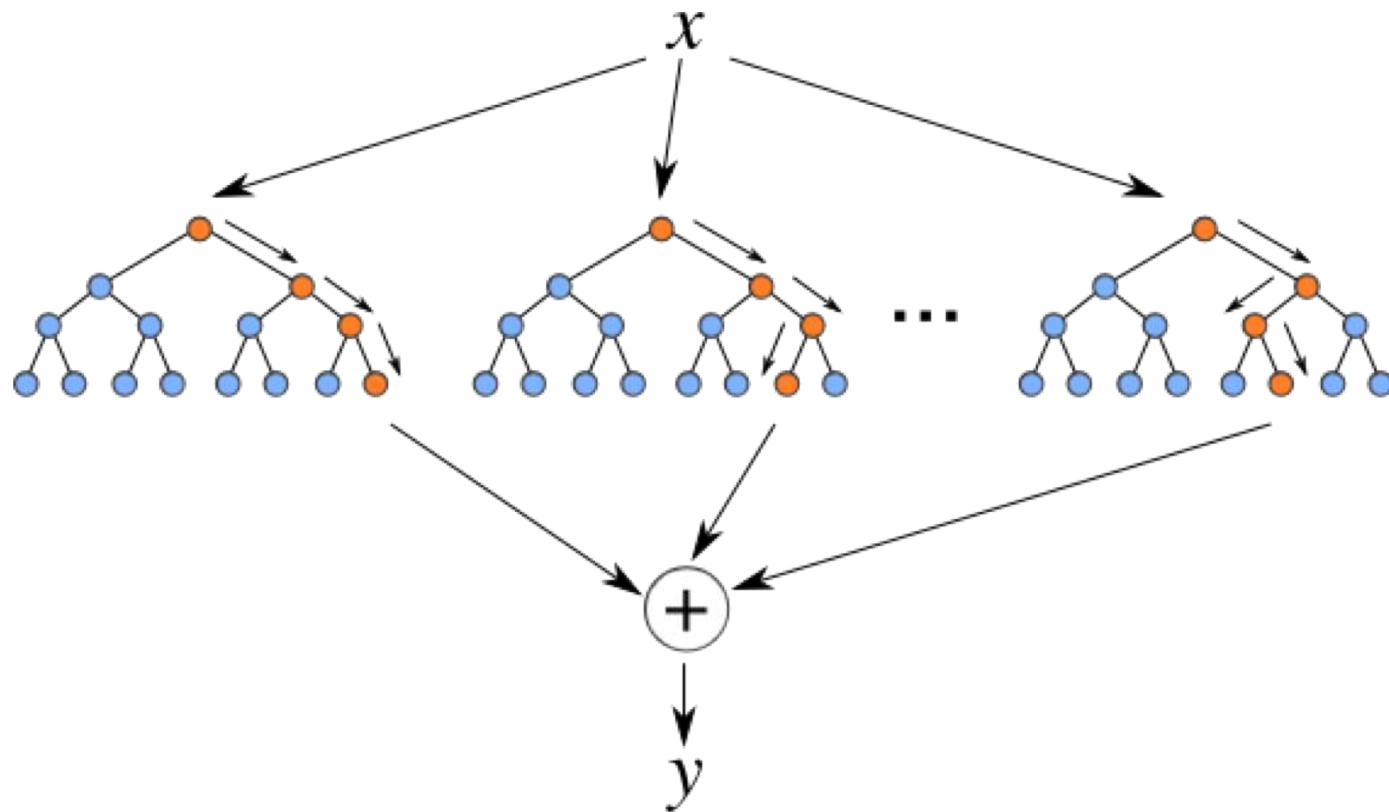
1. Генерируем M выборок на основе имеющейся

Random Forest



1. Генерируем M выборок на основе имеющейся
2. Строим на них деревья с рандомизированными разбиениями в узлах: выбираем k случайных признаков и ищем наиболее информативное разбиение по ним

Random Forest



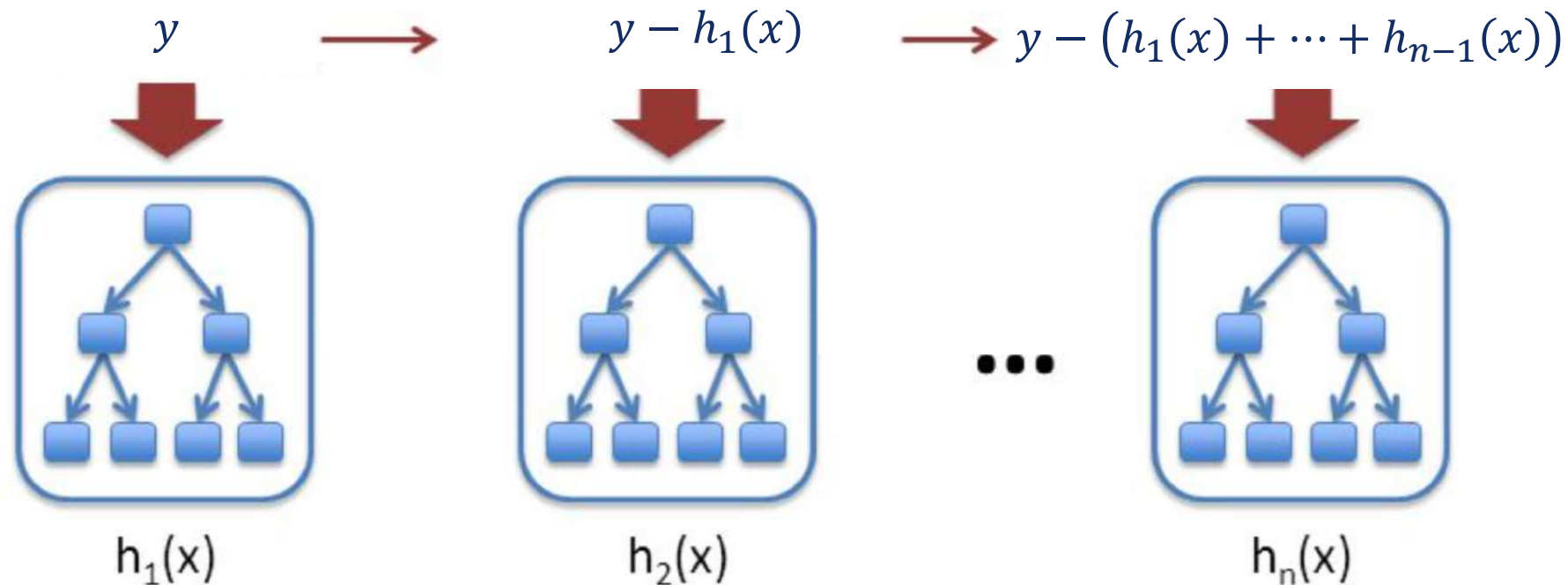
1. Генерируем M выборок на основе имеющейся
2. Строим на них деревья с рандомизированными разбиениями в узлах: выбираем k случайных признаков и ищем наиболее информативное разбиение по ним
3. При прогнозе усредняем ответ всех деревьев

**Идея
Gradient
Boosted
Decision
Trees (GBDT)**

$$h(x) = h_1(x) + \dots + h_n(x)$$

**Идея
Gradient
Boosted
Decision
Trees (GBDT)**

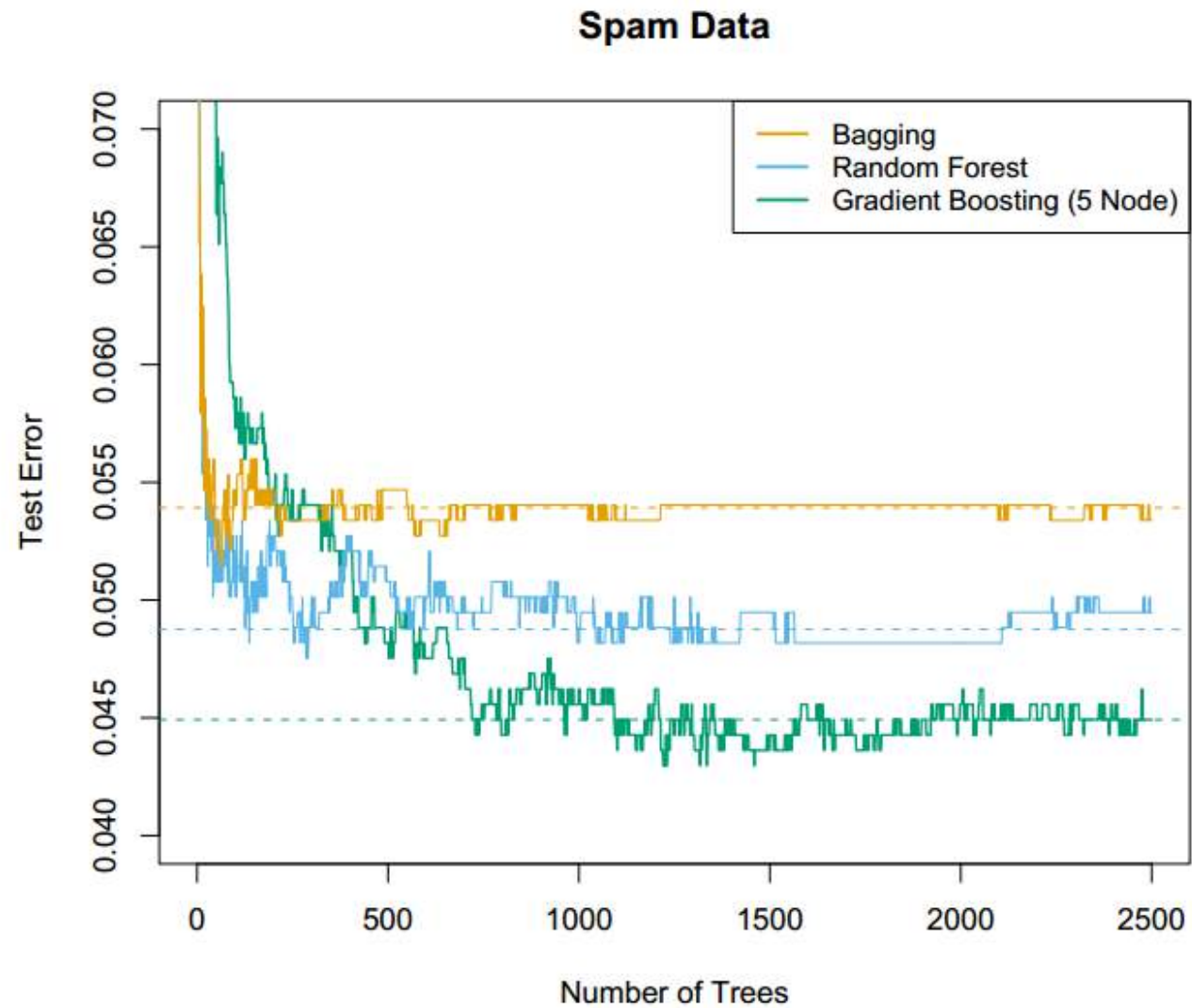
$$h(x) = h_1(x) + \dots + h_n(x)$$



Gradient Boosted Decision Trees

- Каждое новое дерево $h_k(x)$ обучаем на ответы $y_i - h_i$
 h_i - прогноз всей композиции на i -том объекте на предыдущей итерации
- Коэффициент α_k перед новым деревом подбираем с помощью численной оптимизации ошибки

GBDT \gg RF

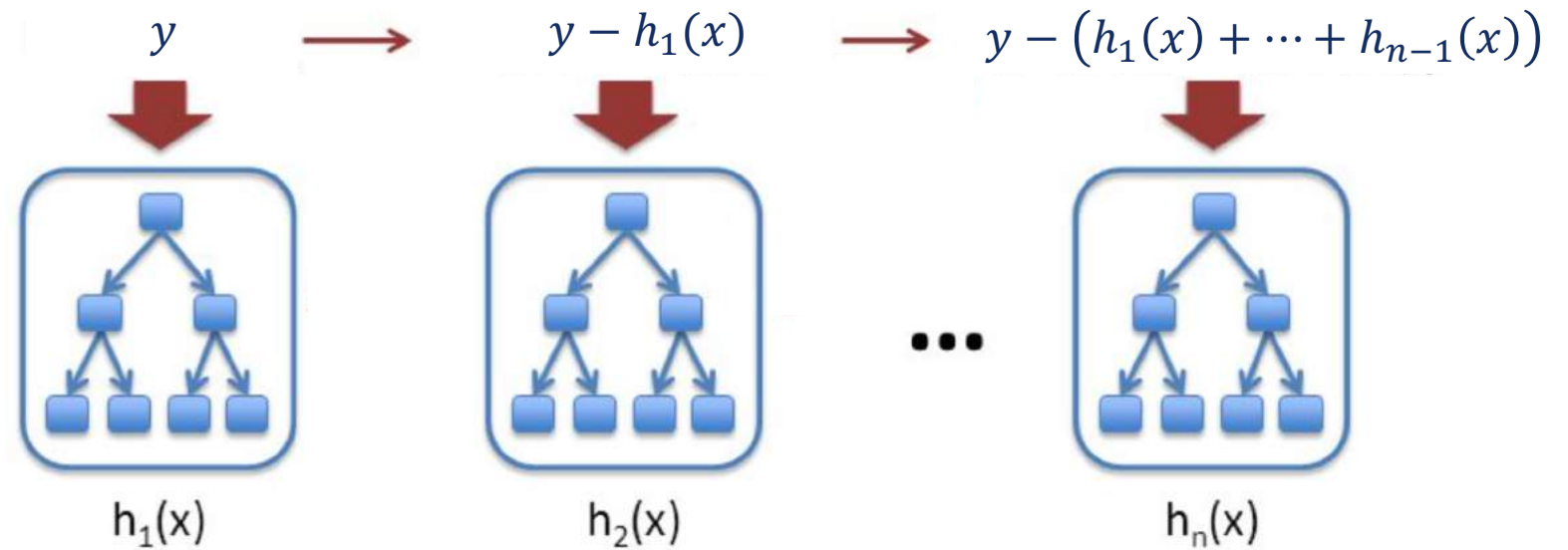


Библиотеки

- Scikit-learn:
 - `sklearn.ensemble.RandomForestClassifier`
 - `sklearn.ensemble.RandomForestRegressor`
- XGBoost
- LightGBM

Идея Gradient Boosted Decision Trees

$$a_n(x) = h_1(x) + \dots + h_n(x)$$



Аналогия с численной оптимизацией

Нам нужно минимизировать ошибку:

$$Q(\hat{y}, y) = \sum_{i=1}^l (\hat{y}_i - y_i)^2 \rightarrow \min \qquad \hat{y}_i = a(x_i)$$

Аналогия с численной оптимизацией

Нам нужно минимизировать ошибку:

$$Q(\hat{y}, y) = \sum_{i=1}^l (\hat{y}_i - y_i)^2 \rightarrow \min \quad \hat{y}_i = a(x_i)$$

Если бы мы подбирали ответы \hat{y} итеративно, можно было бы это делать градиентным спуском

Аналогия с численной оптимизацией

Нам нужно минимизировать ошибку:

$$Q(\hat{y}, y) = \sum_{i=1}^l (\hat{y}_i - y_i)^2 \rightarrow \min \quad \hat{y}_i = a(x_i)$$

Если бы мы подбирали ответы \hat{y} итеративно, можно было бы это делать градиентным спуском

Но нам нужно подобрать не ответы, а функцию $a(x)$

Градиентный бустинг и градиент

В бустинге

$$a(x) = \sum_{t=1}^T \beta_t h_t(x)$$

Идея: будем каждый следующий алгоритм выбирать так, чтобы он приближал антиградиент ошибки

$$h_t(x) \approx -\frac{\partial Q(\hat{y}, y)}{\partial \hat{y}}$$

Градиентный бустинг и градиент

Если $h_t(x) \approx -\frac{\partial Q(\hat{y}, y)}{\partial \hat{y}}$ и $Q(\hat{y}, y) = \sum_{i=1}^l (\hat{y}_i - y_i)^2$

$$h_t(x_i) \approx -\frac{\partial Q(\hat{y}_i, y_i)}{\partial \hat{y}_i} = -2(\hat{y}_i - y_i) \propto y_i - \hat{y}_i$$

**GBM с
квадратичными
потерями**

1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

обучаем h_t на ответы $y_i - a_{t-1}(x_i)$

выбираем β_t

GBM с квадратичными потерями

1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

обучаем h_t на ответы $y_i - a_{t-1}(x_i)$

выбираем β_t

Стратегии выбора β_t :

- всегда равен небольшой константе
- как в методе наискорейшего спуска
- уменьшая с ростом t

GBM с квадратичными потерями

1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

обучаем h_t на ответы $y_i - a_{t-1}(x_i)$

выбираем β_t

Стратегии выбора β_t :

- всегда равен небольшой константе
- как в методе наискорейшего спуска
- уменьшая с ростом t

**GBM с
произвольными
потерями**

1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

обучаем h_t на $-\frac{\partial Q(\hat{y}_i, y_i)}{\partial \hat{y}_i} = -\frac{\partial L(\hat{y}_i, y_i)}{\partial \hat{y}_i}$

выбираем β_t

Здесь $Q(\hat{y}, y) = \sum_{i=1}^l L(\hat{y}_i, y_i)$

$$\hat{y}_i = a_{t-1}(x_i)$$

GBM в наиболее общем виде

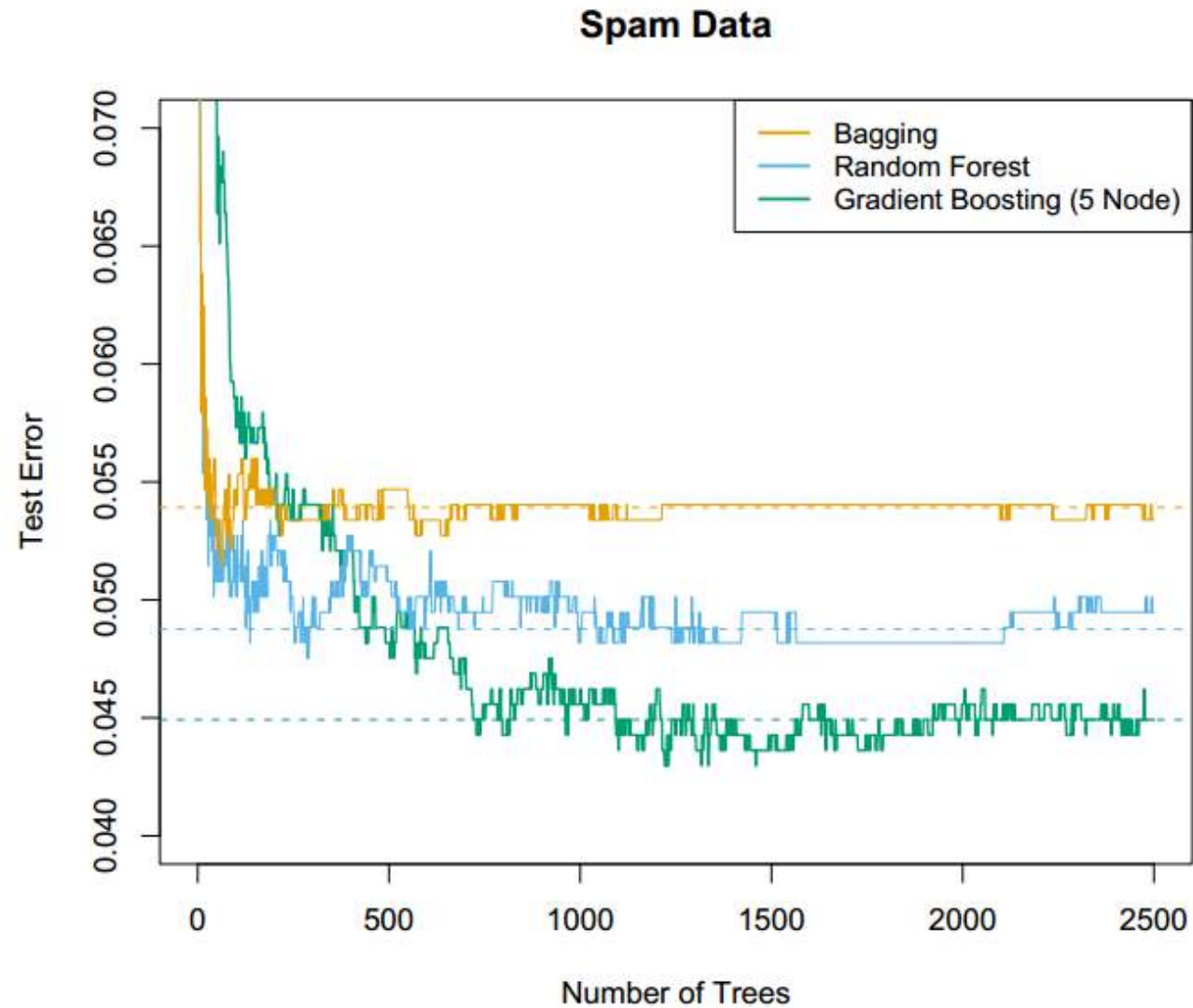
1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

$$h_t = \operatorname{argmin}_h \sum_{i=1}^l \tilde{L} \left(h(x_i), -\frac{\partial L(\hat{y}_i, y_i)}{\partial \hat{y}_i} \right)$$

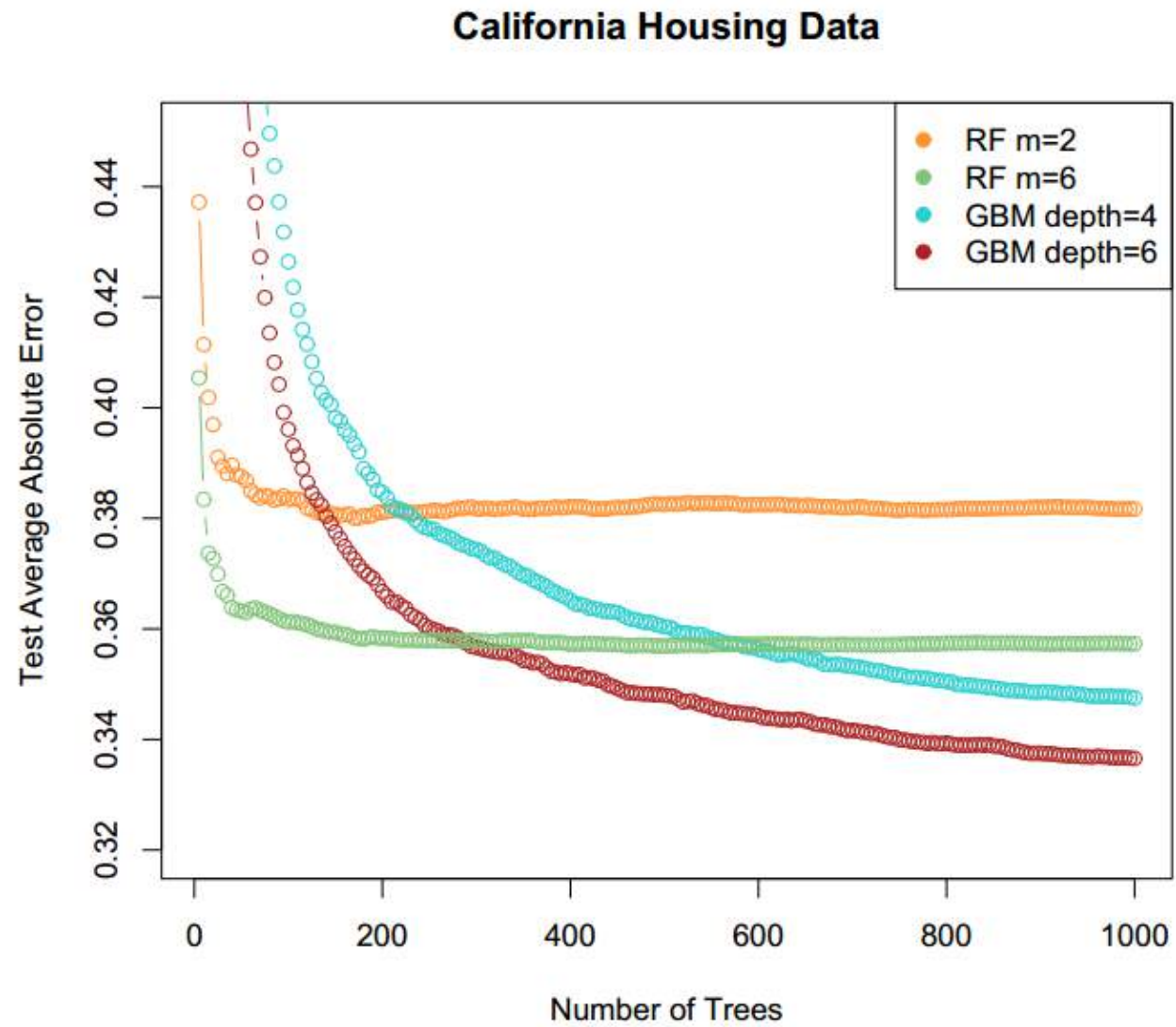
выбираем β_t

$$\text{Здесь } Q(\hat{y}, y) = \sum_{i=1}^l L(\hat{y}_i, y_i) \qquad \hat{y}_i = a_{t-1}(x_i)$$

Bagging, Random Forest, GBDT



GTBM и RF



Параллельная реализация

Вопрос для обсуждения:

Какой из ансамблей деревьев больше подходит для распараллеливания? Как это делать в одном и в другом случае?

Итог

1. Random Forest
2. Gradient Boosted Decision Trees (GBDT)
3. Библиотеки

3. Общие идеи построения ансамблей

Bagging

Bagging = Bootstrap aggregation

Бутстреп

Выборка:

№	X
1	3.4
2	2.9
3	3.7
N	3.1

Бутстреп

Выборка:

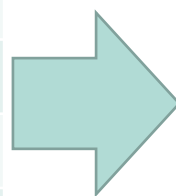
№	X
1	3.4
2	2.9
3	3.7
N	3.1

$$\mathbb{E} X = 3.3 \pm ?$$

Бутстреп

Выборка:

№	X
1	3.4
2	2.9
3	3.7
N	3.1



№	X
3	3.7
1	3.4
2	2.9
2	2.9

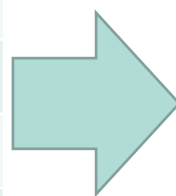
Генерируем новую (искусственную) выборку, отобрав в нее объекты из исходной выборки по схеме выбора с возвращением

$$\mathbb{E} X = 3.3 \pm ?$$

Бутстреп

Выборка:

№	X
1	3.4
2	2.9
3	3.7
N	3.1



№	X	№	X
3	3.7	1	3.4
1	3.4	3	3.7
2	2.9	2	2.9
2	2.9	M	3.0

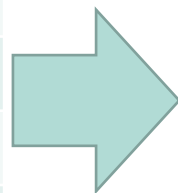
$$\mathbb{E} X = 3.3 \pm ?$$

Продолжаем
генерировать
такие выборки

Бутстреп

Выборка:

№	X
1	3.4
2	2.9
3	3.7
N	3.1



№	X
3	3.7
1	3.4
2	2.9
2	2.9

№	X
1	3.4
3	3.7
2	2.9
M	3.0

...

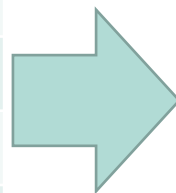
№	X
3	3.7
2	2.9
1	3.4
1	3.4

$$\mathbb{E} X = 3.3 \pm ?$$

Бутстреп

Выборка:

№	X
1	3.4
2	2.9
3	3.7
N	3.1



№	X
3	3.7
1	3.4
2	2.9
2	2.9

№	X
1	3.4
3	3.7
2	2.9
M	3.0

...

№	X
3	3.7
2	2.9
1	3.4
1	3.4

$$\mathbb{E} X = 3.3 \pm ?$$

$$\mathbb{E} X = 3.25$$

$$\mathbb{E} X = 3.27$$

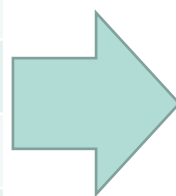
...

$$\mathbb{E} X = 3.39$$

Бутстреп

Выборка:

№	X
1	3.4
2	2.9
3	3.7
N	3.1



№	X
3	3.7
1	3.4
2	2.9
2	2.9

№	X
1	3.4
3	3.7
2	2.9
M	3.0

...

№	X
3	3.7
2	2.9
1	3.4
1	3.4

$$\mathbb{E} X = 3.3 \pm ?$$

$$\mathbb{E} X = 3.25 \quad \mathbb{E} X = 3.27 \quad \dots \quad \mathbb{E} X = 3.39$$

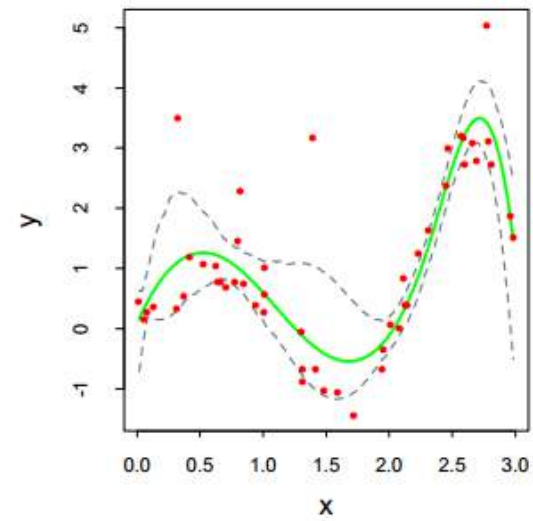
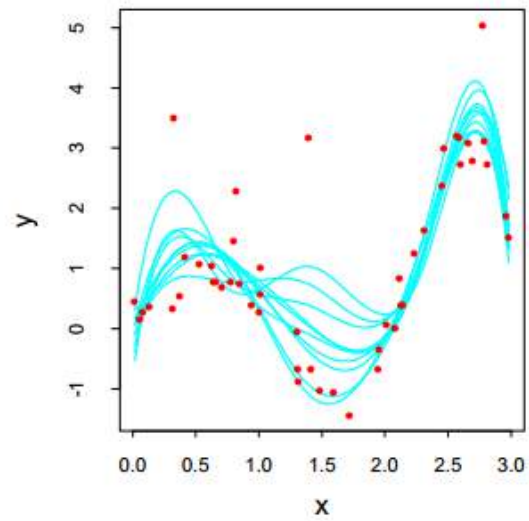
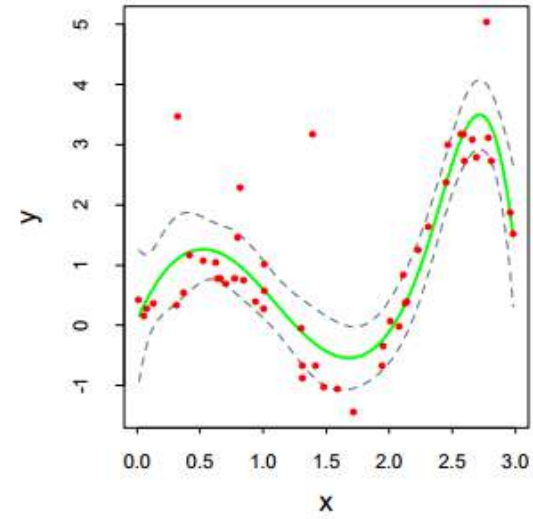
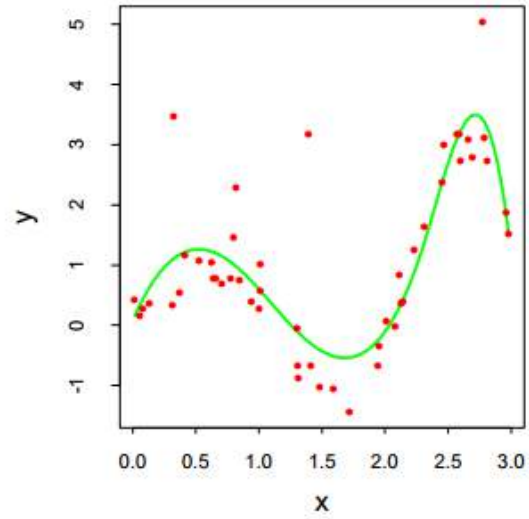
$$\mathbb{E} X = 3.32 \pm 0.06$$

Bagging

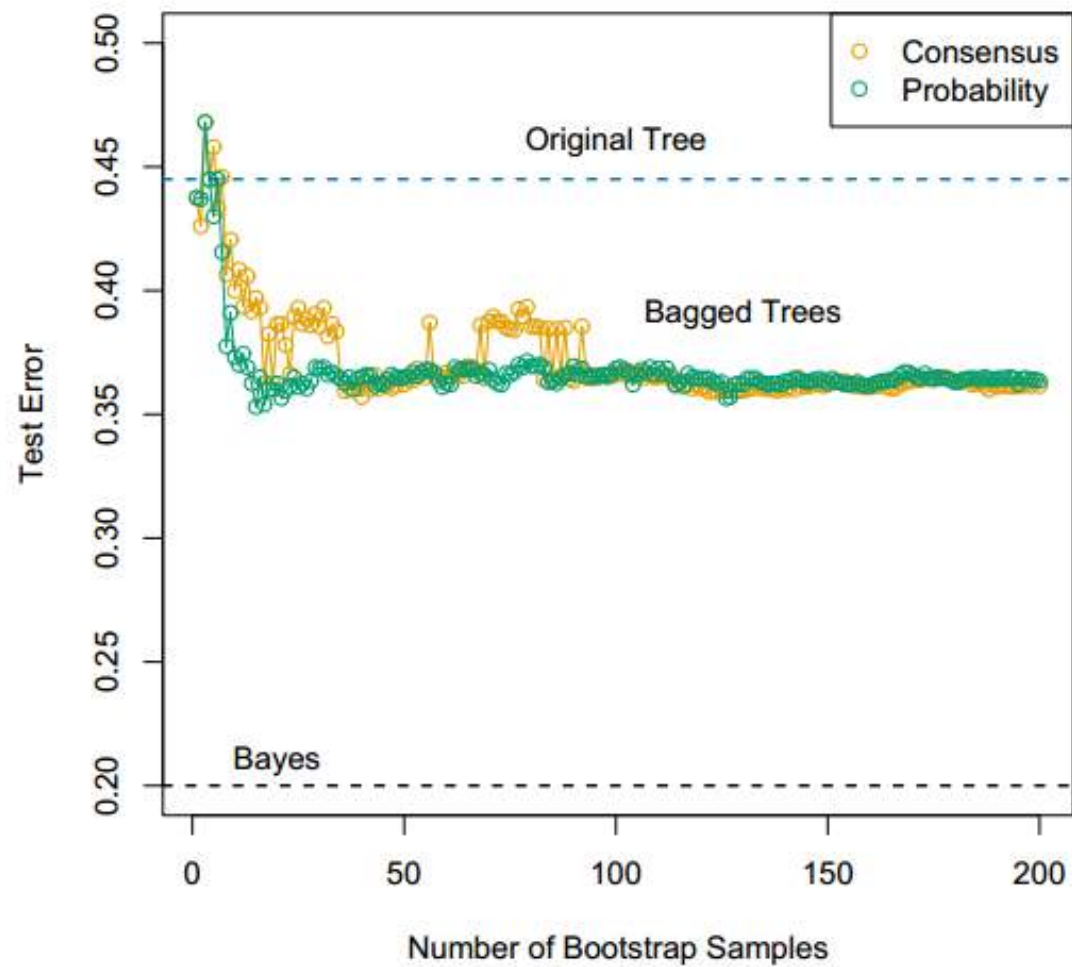
Bagging = Bootstrap aggregation

По схеме выбора с возвращением, генерируем M обучающих выборок такого же размера, обучаем на них модели и усредняем

Bagging



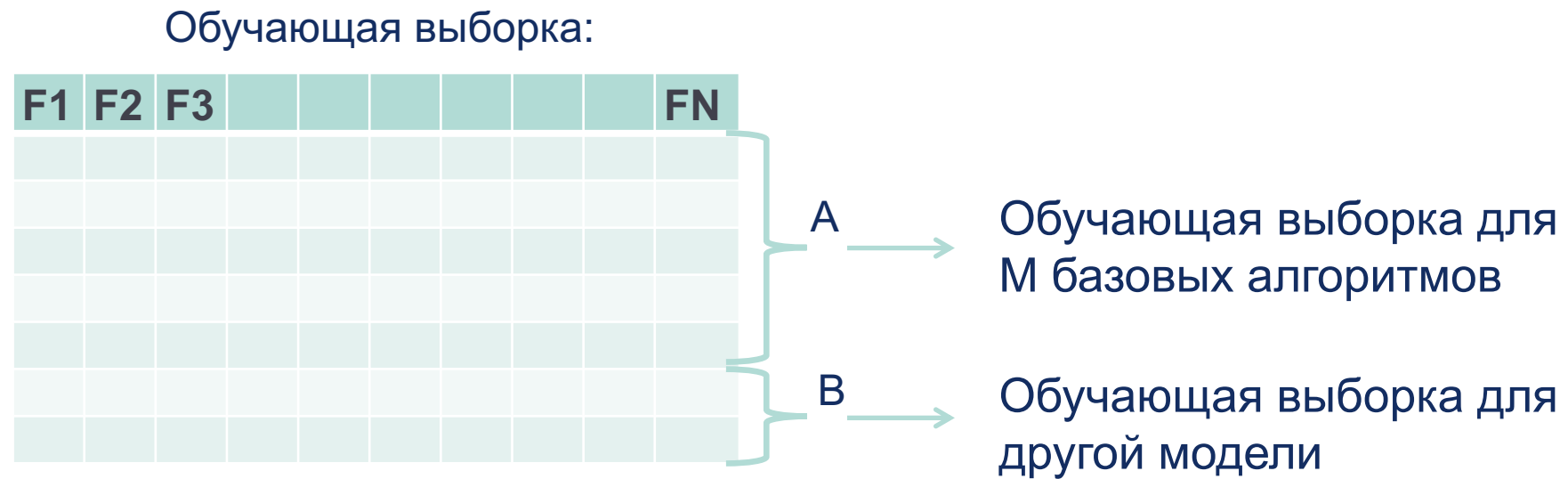
Бэггинг в классификации



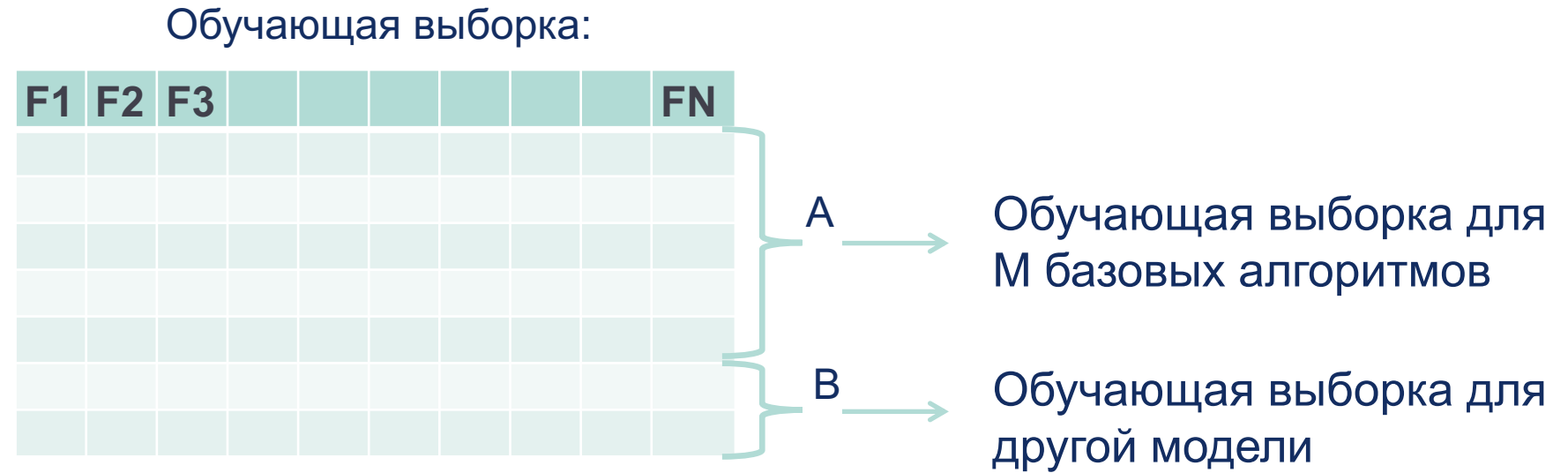
Вариации: Pasting, RSM

- RSM – Random Subspace Method, выбираем не объекты, а признаки
- Pasting – выбираем объекты без возвращения

Stacking



Stacking



Обучаем М базовых алгоритмов на выборке А

Stacking

Обучающая выборка:

F1	F2	F3							FN

A

Обучающая выборка для
М базовых алгоритмов

B

Обучающая выборка для
другой модели

Обучаем М
базовых
алгоритмов
на выборке А

Считаем их
прогнозы на
выборке В

B1	B2			BM

Stacking

Обучающая выборка:

F1	F2	F3							FN

A

Обучающая выборка для
М базовых алгоритмов

B

Обучающая выборка для
другой модели

Обучаем М
базовых
алгоритмов
на выборке А

Считаем их
прогнозы на
выборке В

B1	B2			BM

Обучаем другую
модель (например,
линейную регрессию с
 $w_0 = 0$)

Stacking

Обучающая выборка:

F1	F2	F3							FN

A

Обучающая выборка для
М базовых алгоритмов

B

Обучающая выборка для
другой модели

Обучаем М
базовых
алгоритмов
на выборке А

Считаем их
прогнозы на
выборке В

B1	B2			BM

Обучаем другую
модель (например,
линейную регрессию с
 $w_0 = 0$)

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

Blending

Смесь нескольких сильных классификаторов:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

+ веса неотрицательны и дают в сумме единицу

Blending

Смесь нескольких сильных классификаторов:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

+ веса неотрицательны и дают в сумме единицу

Преимущества и недостатки:

- Очень прост идейно, хорошо работает, логичен

Blending

Смесь нескольких сильных классификаторов:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

+ веса неотрицательны и дают в сумме единицу

Преимущества и недостатки:

- Очень прост идейно, хорошо работает, логичен
- Иногда надо перебирать веса или использовать дискретную оптимизацию

Blending

Смесь нескольких сильных классификаторов:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

+ веса неотрицательны и дают в сумме единицу

Преимущества и недостатки:

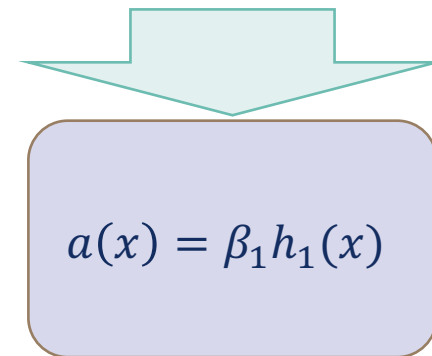
- Очень прост идейно, хорошо работает, логичен
- Иногда надо перебирать веса или использовать дискретную оптимизацию
- Не всегда композиция в виде взвешенной суммы – то, что надо. Иногда нужна более сложная композиция

Boosting

Бустинг – жадное построение взвешенной суммы базовых алгоритмов $h_k(x)$

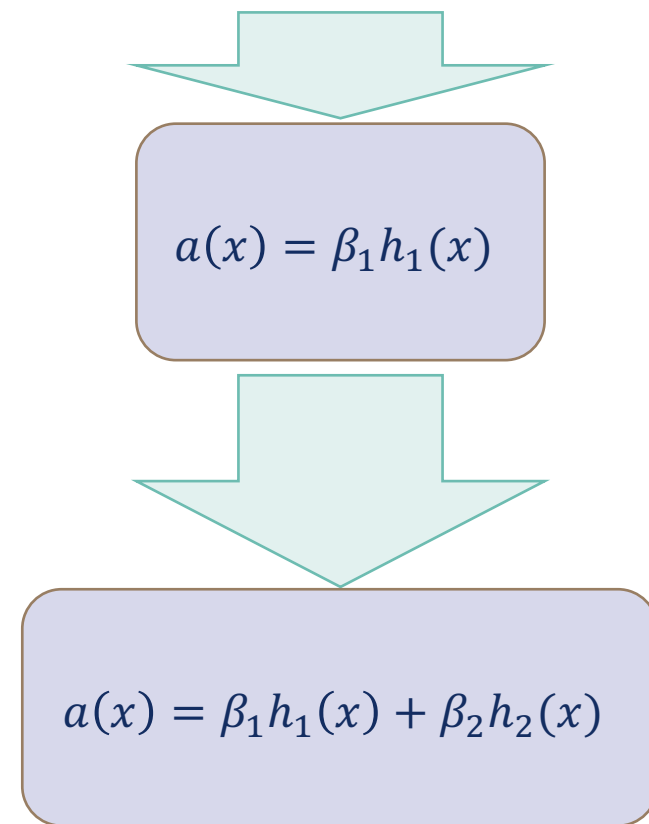
Boosting

Бустинг – жадное
построение взвешенной
суммы базовых
алгоритмов $h_k(x)$


$$a(x) = \beta_1 h_1(x)$$

Boosting

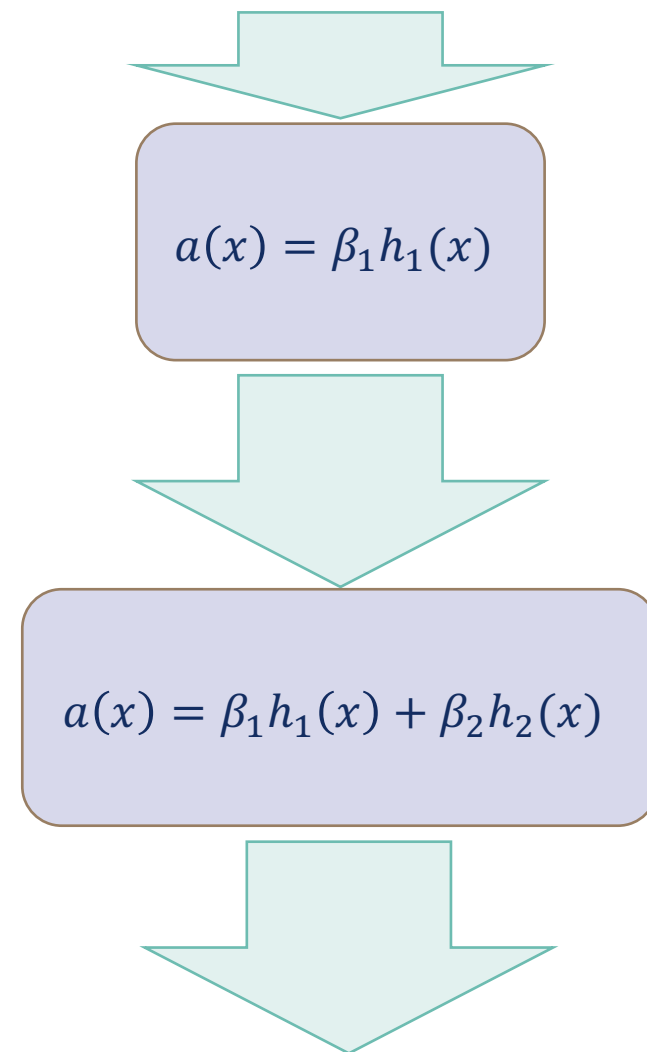
Бустинг – жадное
построение взвешенной
суммы базовых
алгоритмов $h_k(x)$



Boosting

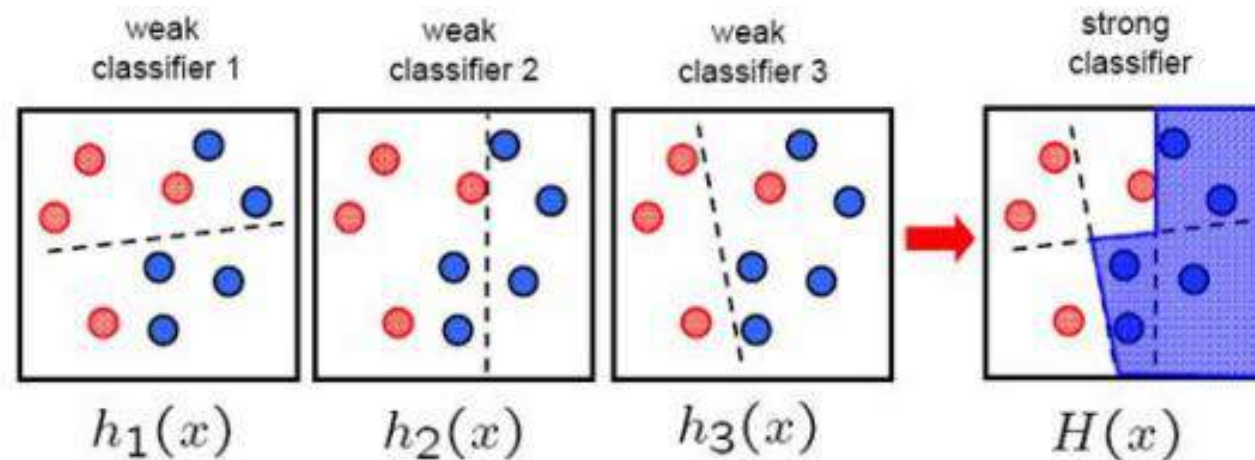
Бустинг – жадное
построение взвешенной
суммы базовых
алгоритмов $h_k(x)$

$$a(x) = \sum_{t=1}^T \beta_t h_t(x)$$



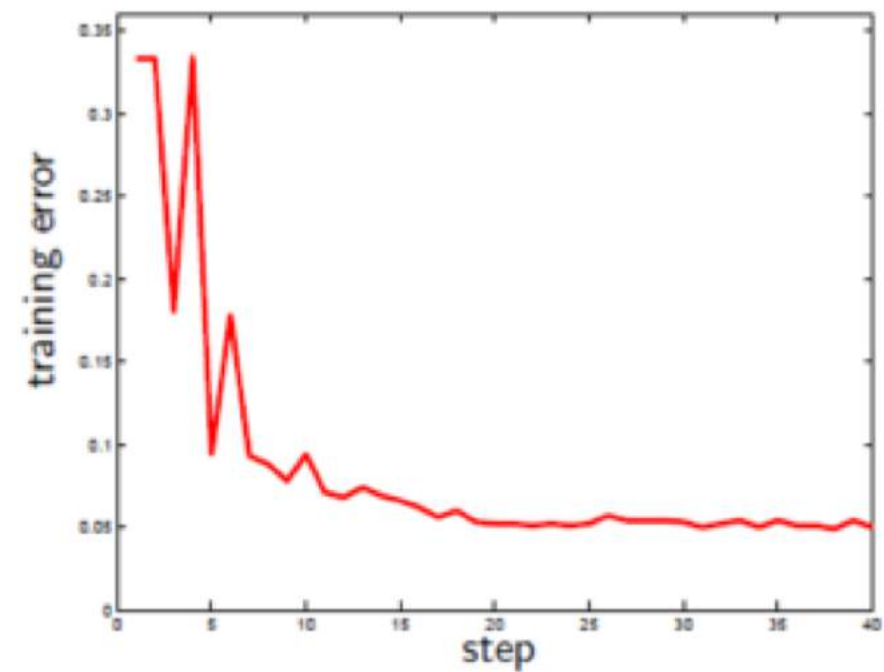
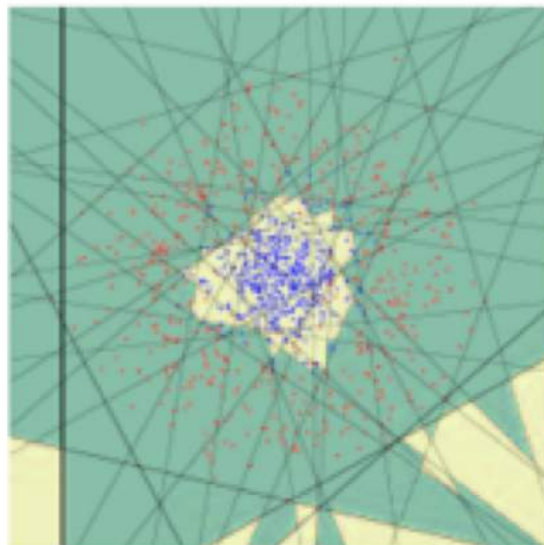
«Слабые» алгоритмы

$h_k(x)$ – как правило, решающие деревья
небольшой глубины или линейные
модели



Пример:
бустинг над
линейными
моделями

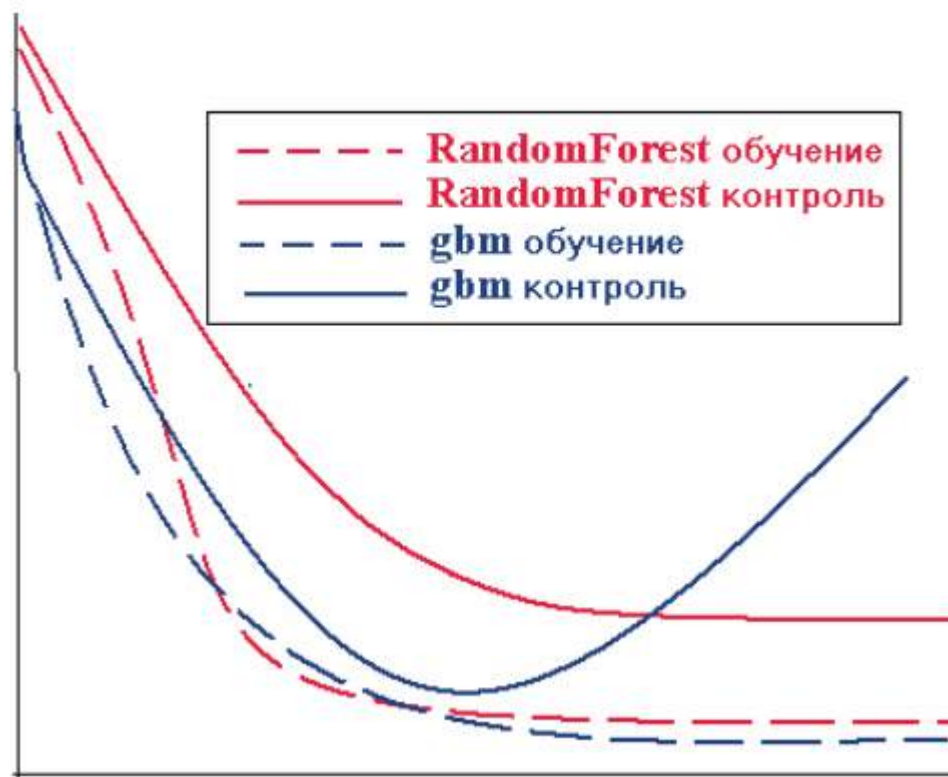
$t = 40$



Алгоритмы бустинга

- Основные алгоритмы:
 - Градиентный бустинг
 - Адаптивный бустинг (AdaBoost)
- Вариации AdaBoost:
 - AnyBoost (произвольная функция потерь)
 - BrownBoost
 - GentleBoost
 - LogitBoost
 -

Бэггинг и бустинг: переобучение



Преимущества и недостатки бустинга

- Позволяет очень точно приблизить восстанавливаемую функцию или разделяющую поверхность классов
- Плохо интерпретируем
- Композиции могут содержать десятки тысяч базовых моделей и долго обучаться
- Переобучение на выбросах при избыточном количестве классификаторов

4.Извлечение и простые преобразования признаков

Виды признаков

Какие бывают признаки:

1. Числовые
2. Порядковые
3. Категориальные
4. Даты и время
5. Координаты

Даты и время

1. Количество прошедших секунд
например, с 00:00:00 UTC, 1 January 1970
2. Использование периодичности
 - а. номер дня в году, в месяце, в неделе
 - б. час, минута, секунда
3. Время до/после важных событий
Например, количество дней, оставшихся до
ближайшего праздника

Координаты

1. Повороты системы координат на 45 градусов, 22.5 градусов, etc
2. Добавление расстояний до:
 - a. Других объектов из выборки
 - b. Центров кластеров
 - c. Инфраструктурных зданий - магазинов, школ, больниц

Категориальные признаки (строки)

Из колонок “name”, “ticket”, “cabin” можно
сгенерировать новые признаки

	A	B	C	D	E	F	G	H	I	J	K
1	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
2	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
4	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2.	7.925		S
5	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
8	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
9	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
10	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
11	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
12	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
13	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S

Категориальные признаки

Бинаризация

feature
a
b
c
b



feature == a	feature == b	feature == c
1		
	1	
		1
	1	

Категориальные признаки

Hashing trick

feature
a
b
c
b



feature == a or feature == c	feature == b
1	
	1
1	
	1

Категориальные
признаки

№ склада	Город	...	Продано вина (ящиков)
2343	Москва	...	56000
185	Самара	...	10500
121	Ростов	...	11300
...

Категориальные
признаки

№ склада	В среднем продают в городе	...	Продано вина (ящиков)
2343	59000	...	56000
185	11200	...	10500
121	12100	...	11300
...

Метапризнаки

Использование ответов других алгоритмов

	xgb_prediction	knn_prediction	svm_prediction	target
train	0.192	0.293	0.122	0
train	0.789	0.890	0.670	1
test	0.542	0.310	0.173	?

Осторожно с переобучением: используйте KFold, LOO

Feature engineering

- Выделение признаков (feature extraction) – генерация признаков по известным данным
- Отбор признаков (feature selection) – ранжирование признаков по «полезности» и выкидывание наименее полезных (или даже наоборот «вредных»)
- Преобразование признаков (feature transform) – создание новых признаков на основе имеющихся

Генерация признаков

Для решения задачи нужно использовать разные типы данных

Пример: задача рекомендации музыки

1. Музыкальные треки
2. Тексты песен
3. Плейлисты

Проблема: нужно преобразовать к одному формату - матрице “объекты-признаки”

Пример 1: текстовые признаки

- Dataset: 20news_groups
- Электронные письма, разбитые по 20 темам (классам)
- Попробуем придумать классификатор, который различает две темы:
- auto и politics.mideast

Извлечение текстовых признаков

Пример письма 1:

From: carl_f_hoffman@cup.portal.com
Newsgroups: rec.autos
Subject: 1993 Infiniti G20
Message-ID: <78834@cup.portal.com>

Date: Mon, 5 Apr 93 07:36:47 PDT
Organization: The Portal System (TM)
Lines: 26

I am thinking about getting an Infiniti G20. In consumer reports it is ranked high in many catagories including highest in reliability index for compact cars. Mitsubishi Galant was second followed by Honda Accord).

A couple of things though:

- 1) In looking around I have yet to see anyone driving this car. I see lots of Honda's and Toyota's.
- 2) There is a special deal where I can get an Infinity G20, fully loaded, at dealer cost (I have check this out and the numbers match up). They are doing this because they are releasing and update mid-1993 version (includes dual air-bags) and want to get rid of their old 1993's.

I guess my question is: Is this a good deal?
Also, Can anyone give me any feedback on Infiniti?

Thanks,
Carl Hoffman

P.S.

The other cars that I have test driven and which are in the running are: Mitsubishi Galant, Honda Accord, and Toyota Camary

Извлечение текстовых признаков

Пример письма 2:

From: Bob.Waldrop@f418.n104.z1.fidonet.org (Bob Waldrop)
Subject: Celebrate Liberty! 1993
Message-ID: <1993Apr5.201336.16132@dsd.es.com>
Followup-To: talk.politics.misc

Announcing. . . Announcing. . . Announcing. . . Announcing. . .

CELEBRATE LIBERTY!
1993 LIBERTARIAN PARTY NATIONAL CONVENTION
AND POLITICAL EXPO

THE MARRIOTT HOTEL AND THE SALT PALACE
SALT LAKE CITY, UTAH

INCLUDES INFORMATION ON DELEGATE DEALS!
(Back by Popular Demand!)

The convention will be held at the Salt Palace Convention Center and the Marriott Hotel, Salt Lake City, Utah. The business sessions, Karl Hess Institute, and Political Expo are at the Salt Palace; breakfasts, parties, and banquet are at the Marriott Hotel.

Marriott Hotel room rates are \$79.00 night, plus 10.5% tax (\$87.17 total). This rate is good for one to four persons room occupancy. Double is one or two beds; 3 or 4 people is 2 beds. You can make your reservations direct with the hotel (801-531-0800), or you can purchase your room

Текстовые признаки: bag-of-words



the world of

TOTAL

► All About The Company

Global Activities
Corporate Structure
TOTAL's Story
Upstream Strategy
Downstream Strategy
Chemicals Strategy
TOTAL Foundation
Homepage

all about the
company

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

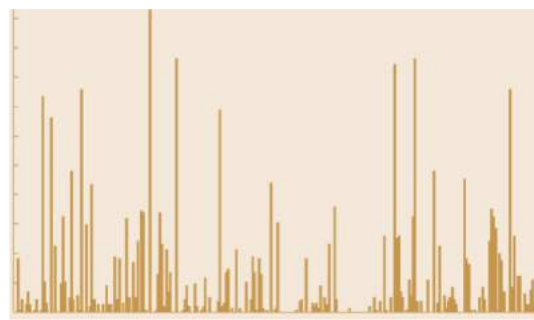
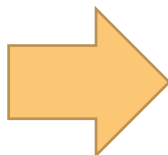
Our expanding refining and marketing operations in Asia and the Mediterranean Basin complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

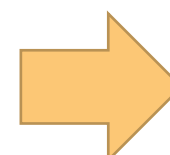


aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
...	
Zaire	0

Простой классификатор текстов



Bag-of-words



Взвешивание частот слов в текстах

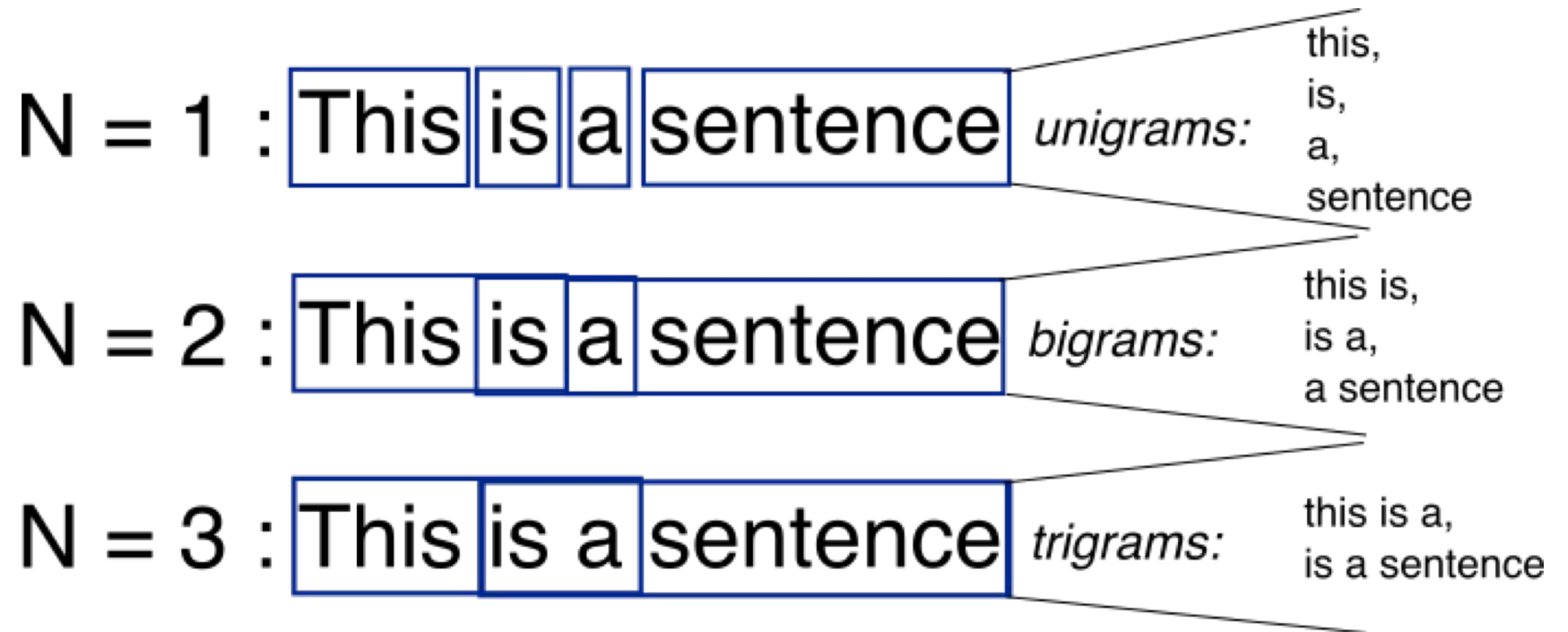
Term Frequency

$$\text{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

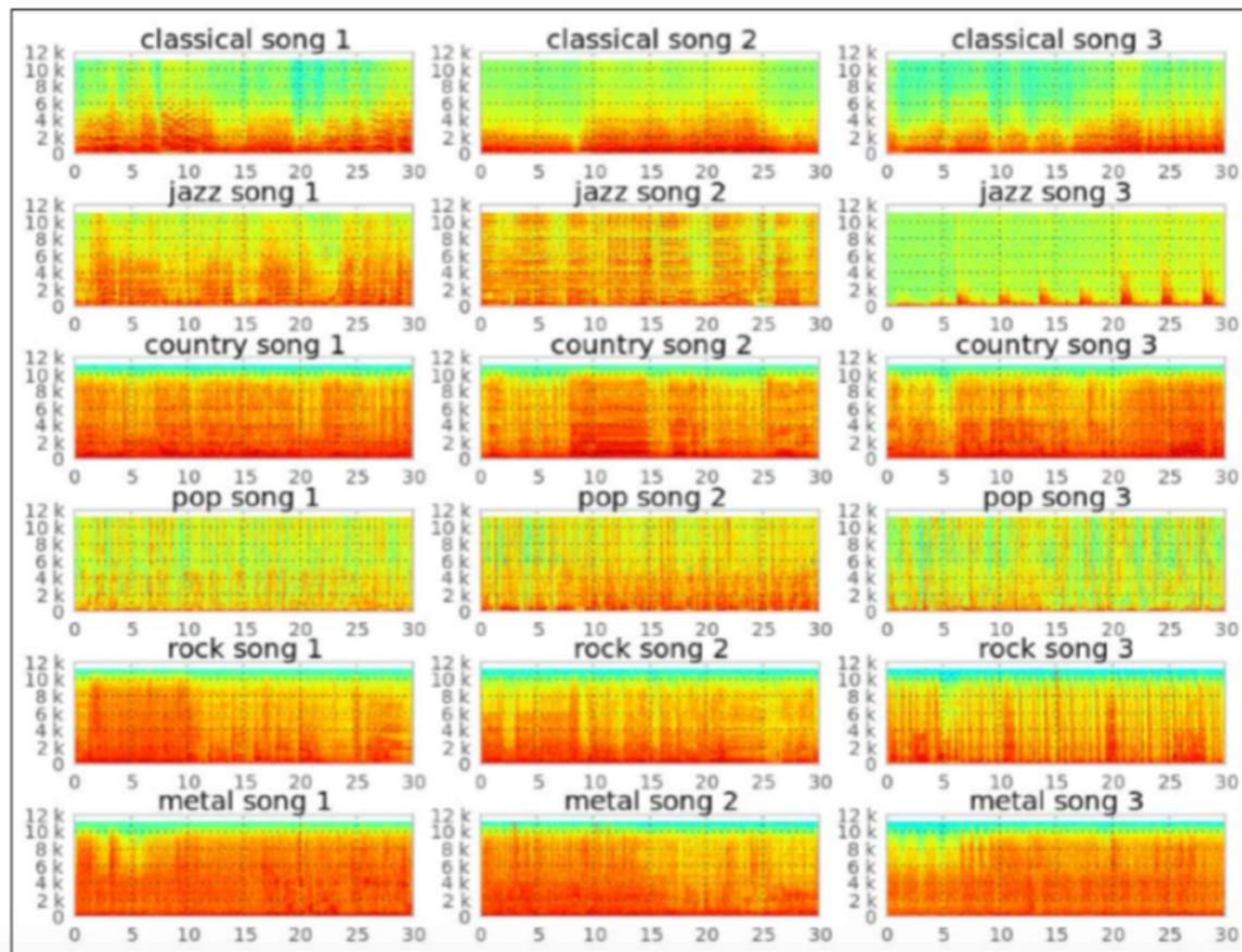
Inverse Document Frequency

$$\text{idf}_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

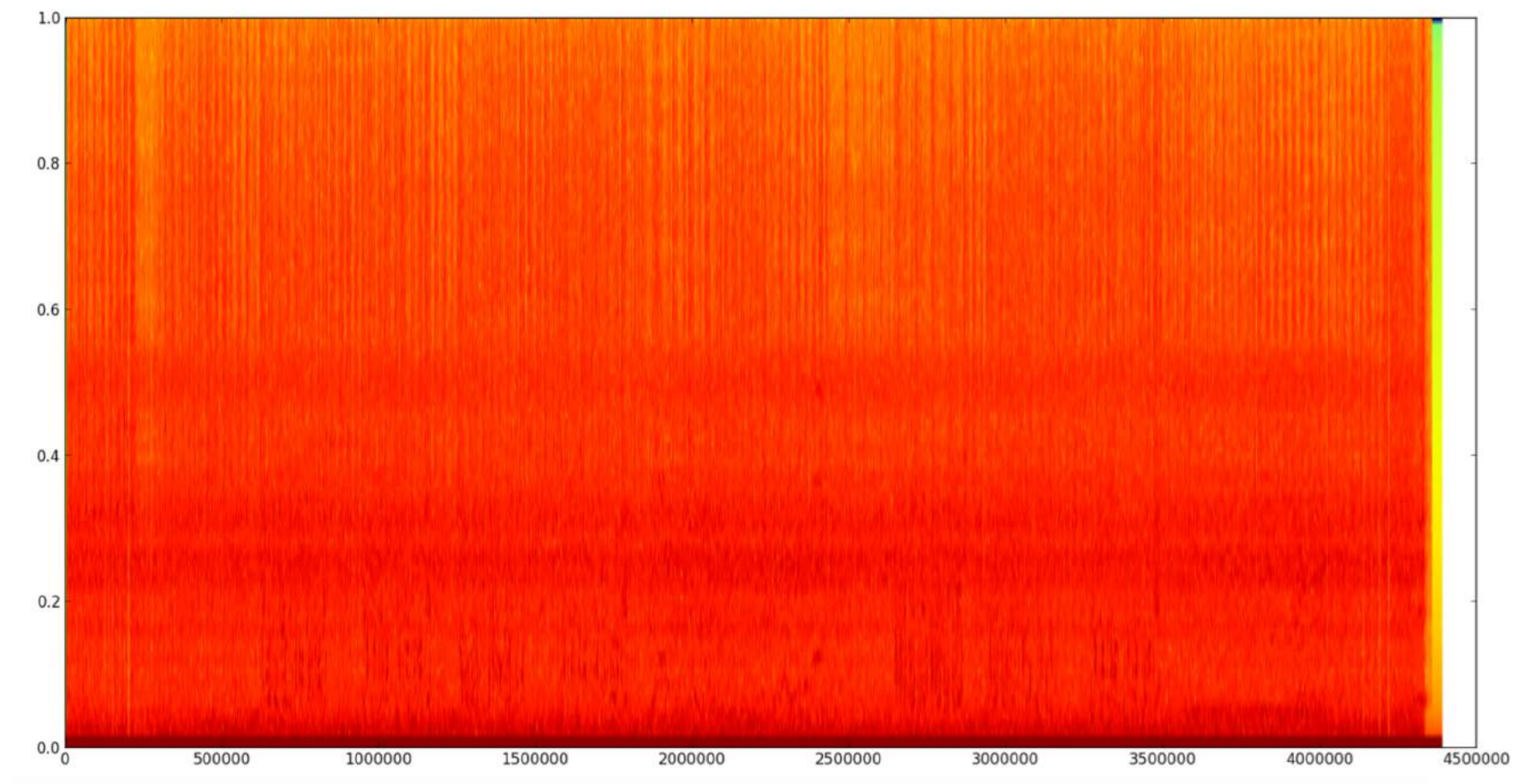
Частоты N-грамм



Пример 2: признаки аудиофайла

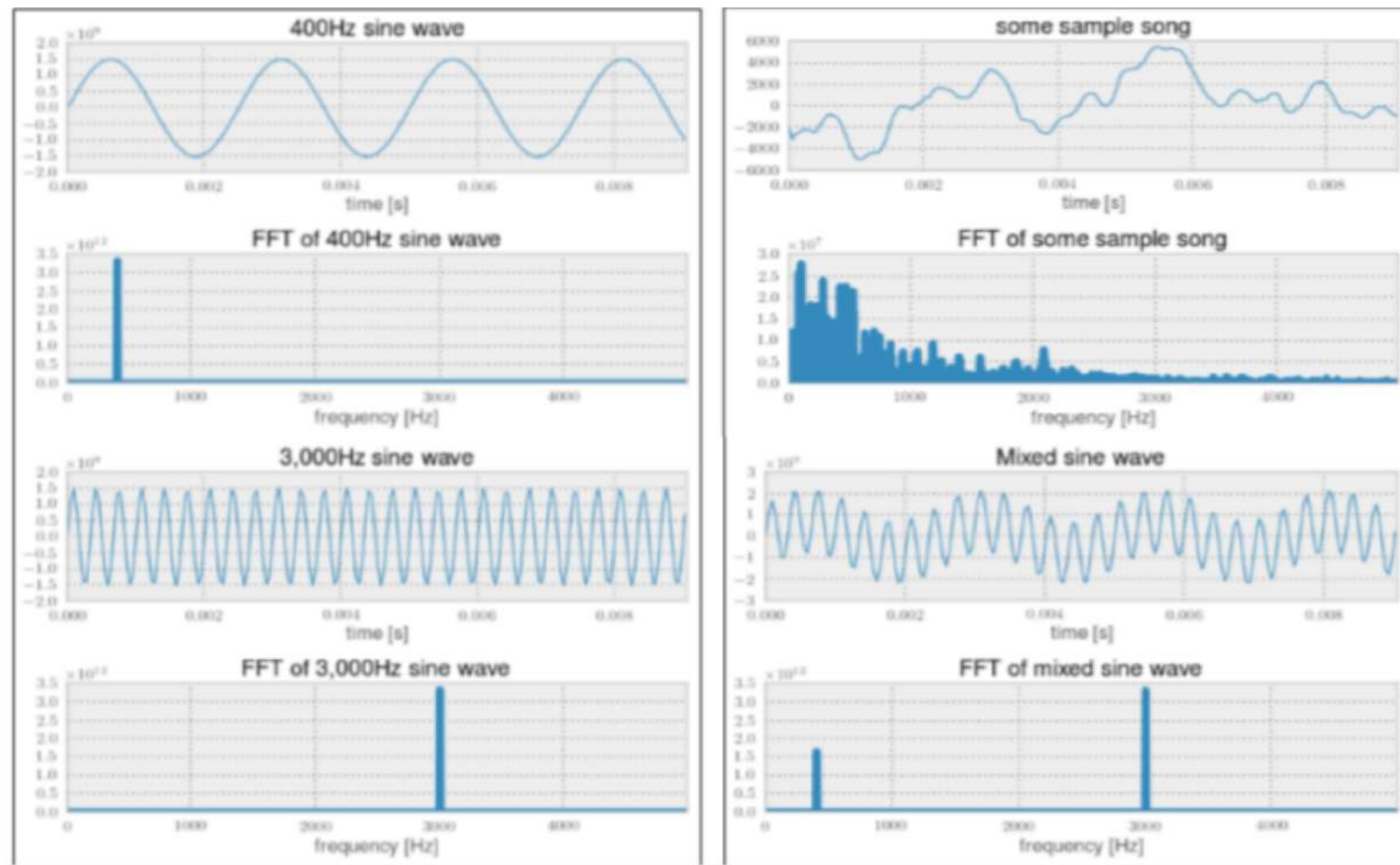


Пример 2: признаки аудиофайла



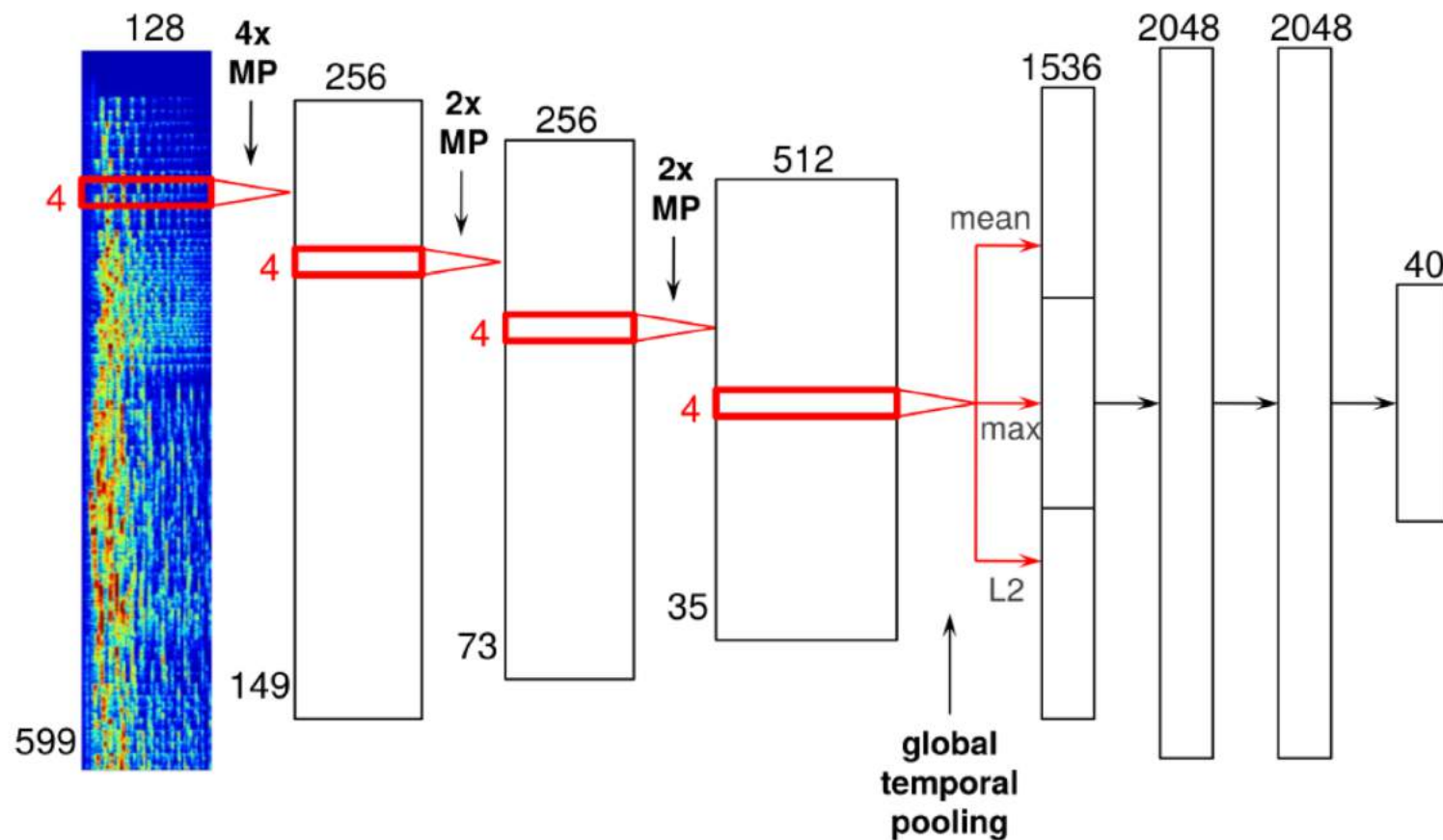
MFCC - преобразование Фурье логарифма спектра

Пример 2: признаки аудиофайла



Пример 2: признаки аудиофайла

Embeddings с помощью нейронных сетей:



Пример 3: признаки изображения

Input image

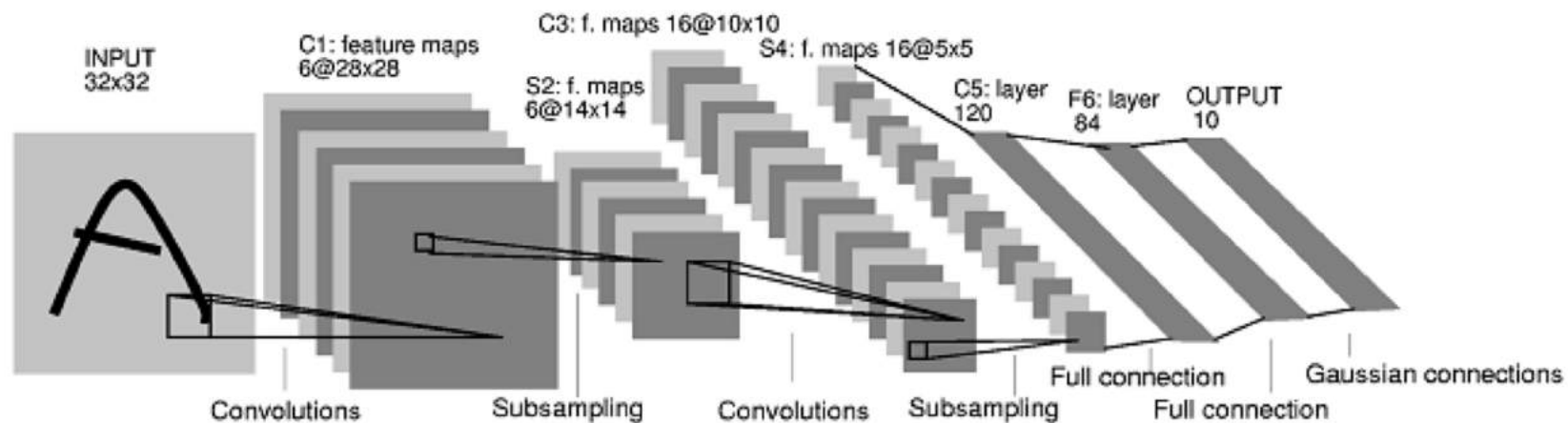


Histogram of Oriented Gradients



Пример 3: признаки изображения

Выходы слоев из нейросети



5.Отбор признаков

Отбор признаков

1. Статистические методы
2. С помощью регуляризации L1
3. Жадный отбор
4. С помощью моделей

**Отбор признаков
по
статистическим
критериям**

Пример: критерий хи-квадрат позволяет отобрать лучшие бинарные признаки для каждого класса

	Значение признака 1	Значение признака 0
Объект принадлежит классу	A	B
Объект не принадлежит классу	C	D

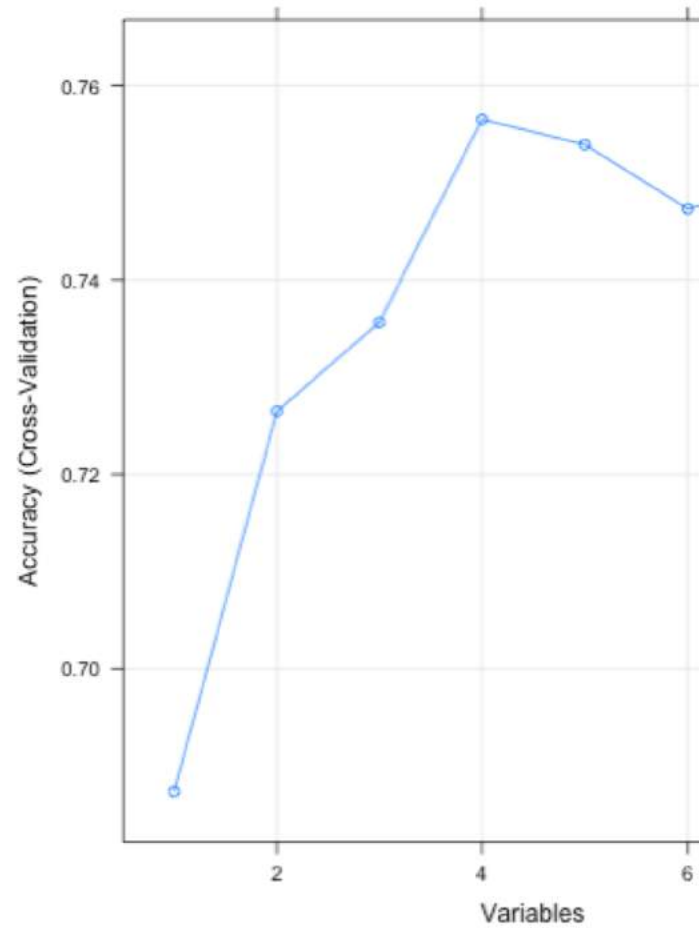
$$\chi^2 = \frac{N(AD - BC)^2}{(A + B)(A + C)(B + D)(C + D)}$$

Отбор
признаков с
помощью l1-
регуляризации

$$\sum_{i=1}^l L(M_i) + \gamma \sum_{k=1}^m |w_k| \rightarrow \min$$

Жадный отбор признаков

Чередование добавления и удаления признаков

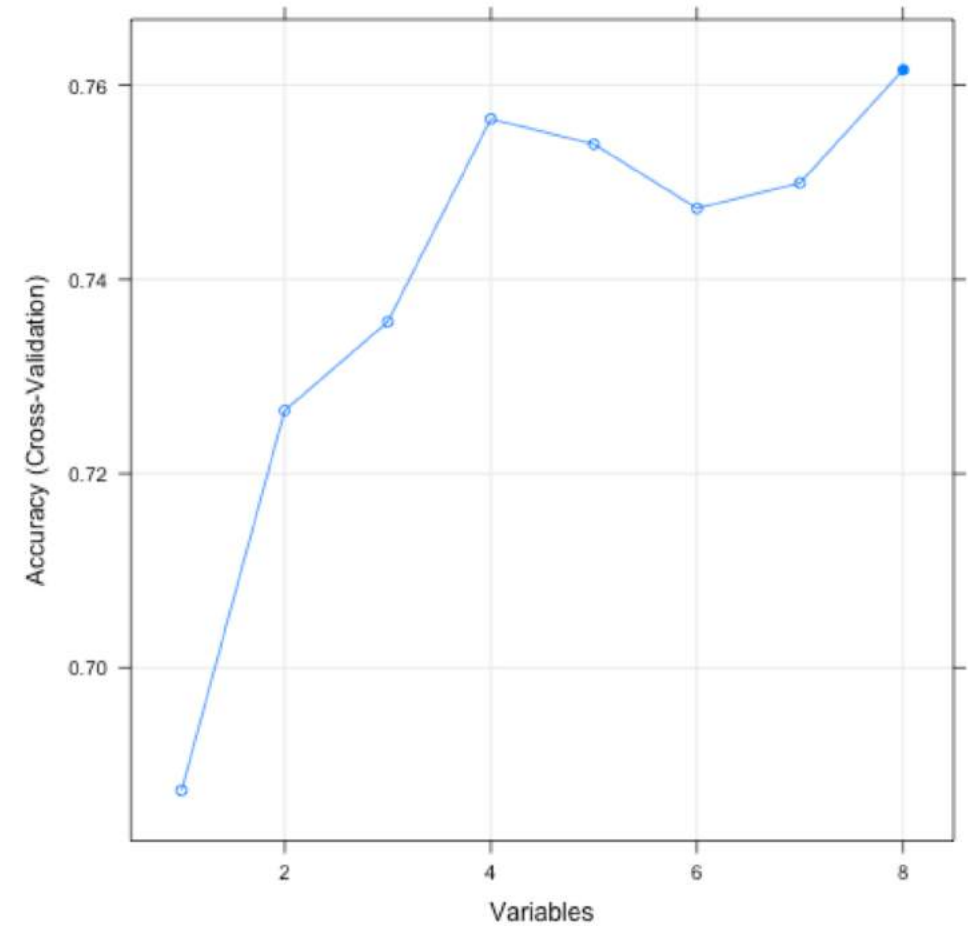


Жадный отбор признаков

Чередование
добавления и
удаления признаков

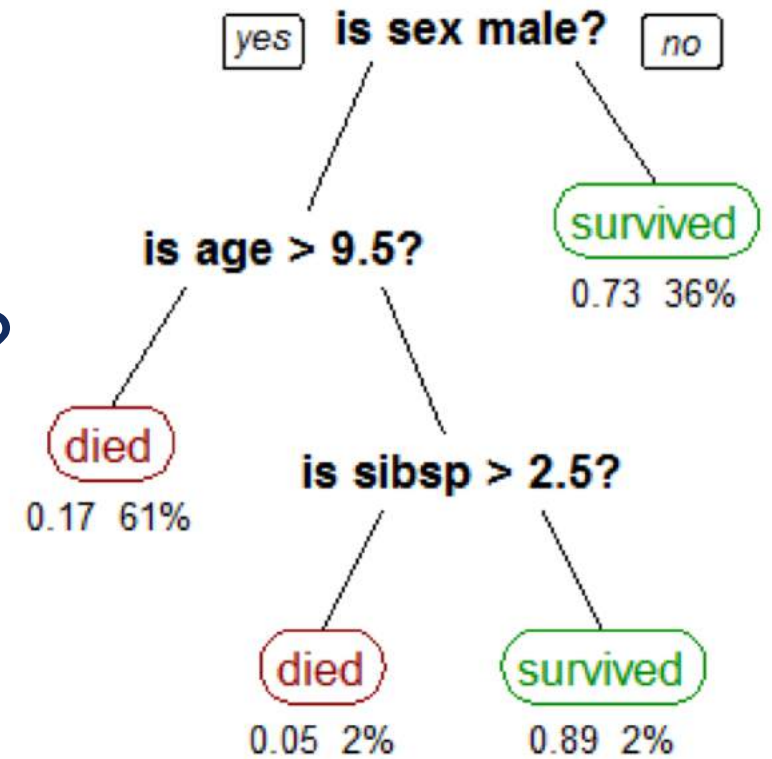
Этап добавления:
добавляем лучшие
признаки

Этап удаления:
удаляем худшие
признаки



Отбор признаков с помощью моделей

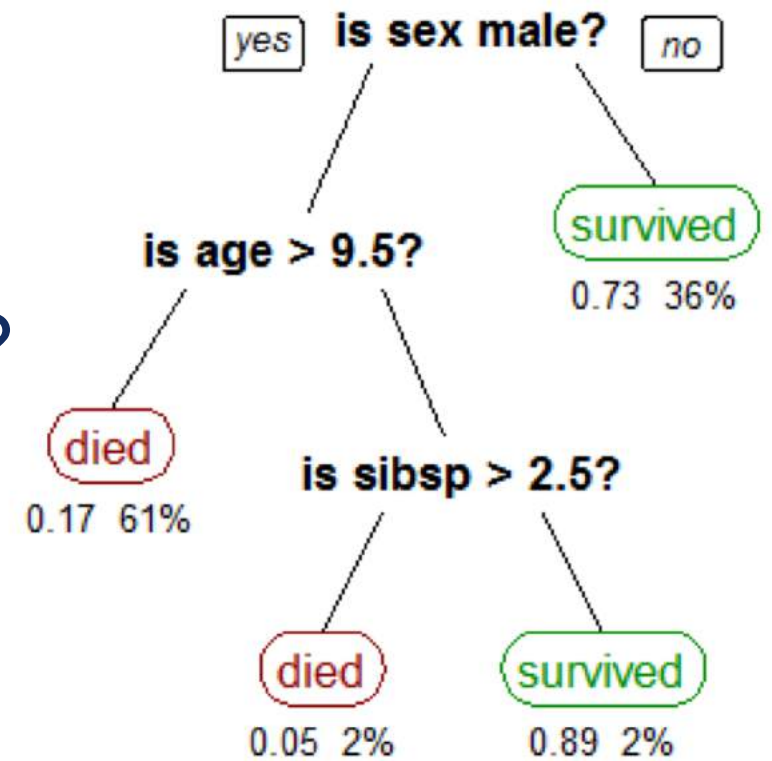
Вопрос: как можно оценивать важность признака в решающих деревьях?



Отбор признаков с помощью моделей

Вопрос: как можно оценивать важность признака в решающих деревьях?

А в линейных моделях?



План лекции

1. Решающие деревья

2. Ансамбли деревьев

3. Общие идеи построения ансамблей

4. Извлечение и простые преобразования признаков

5. Отбор признаков

Data Mining in Action

Лекция 4

Группа курса в Telegram:



<https://t.me/joinchat/B1OITk74nRV56Dp1TDJGNA>

eXtreme Gradient Boosting (XGBoost)

<https://arxiv.org/pdf/1603.02754.pdf>

eXtreme Gradient Boosting (XGBoost)

$$\sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b(x_i)) \rightarrow \min_b$$

$$s = \left(- \frac{\partial L}{\partial z} \Big|_{z=a_{N-1}(x_i)} \right)_{i=1}^{\ell} = -\nabla_s \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + s_i)$$

$$b_N(x) = \arg \min_{b \in \mathcal{A}} \sum_{i=1}^{\ell} (b(x_i) - s_i)^2$$



$$b_N(x) = \arg \min_{b \in \mathcal{A}} \sum_{i=1}^{\ell} \left(b(x_i) - \frac{s_i}{h_i} \right)^2 \quad h_i = \frac{\partial^2 L}{\partial z^2} \Big|_{z=a_{N-1}(x_i)}$$

eXtreme Gradient Boosting (XGBoost)

$$b_N(x) = \arg \min_{b \in \mathcal{A}} \sum_{i=1}^{\ell} \left(b(x_i) - \frac{s_i}{h_i} \right)^2$$

$$b(x) = \sum_{j=1}^J b_j [x \in R_j]$$

$$\sum_{i=1}^{\ell} \left(-s_i b(x_i) + \frac{1}{2} h_i b^2(x_i) \right) + \lambda J + \frac{\mu}{2} \sum_{j=1}^J b_j^2 \rightarrow \min_b$$

$$\sum_{j=1}^J \left\{ \underbrace{\left(-\sum_{i \in R_j} s_i \right)}_{=-S_j} b_j + \frac{1}{2} \left(\mu + \underbrace{\sum_{i \in R_j} h_i}_{=H_j} \right) b_j^2 + \lambda \right\}$$

eXtreme Gradient Boosting (XGBoost)

$$b_j = \frac{S_i}{H_j + \mu}$$

$$H(b) = \frac{1}{2} \sum_{j=1}^J \frac{S_j^2}{H_j + \mu} + \lambda J$$

$$H(b_l) + H(b_r) - H(b) - \lambda \rightarrow \max$$

LightGBM

<https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree>