



Ансамбли и beyond

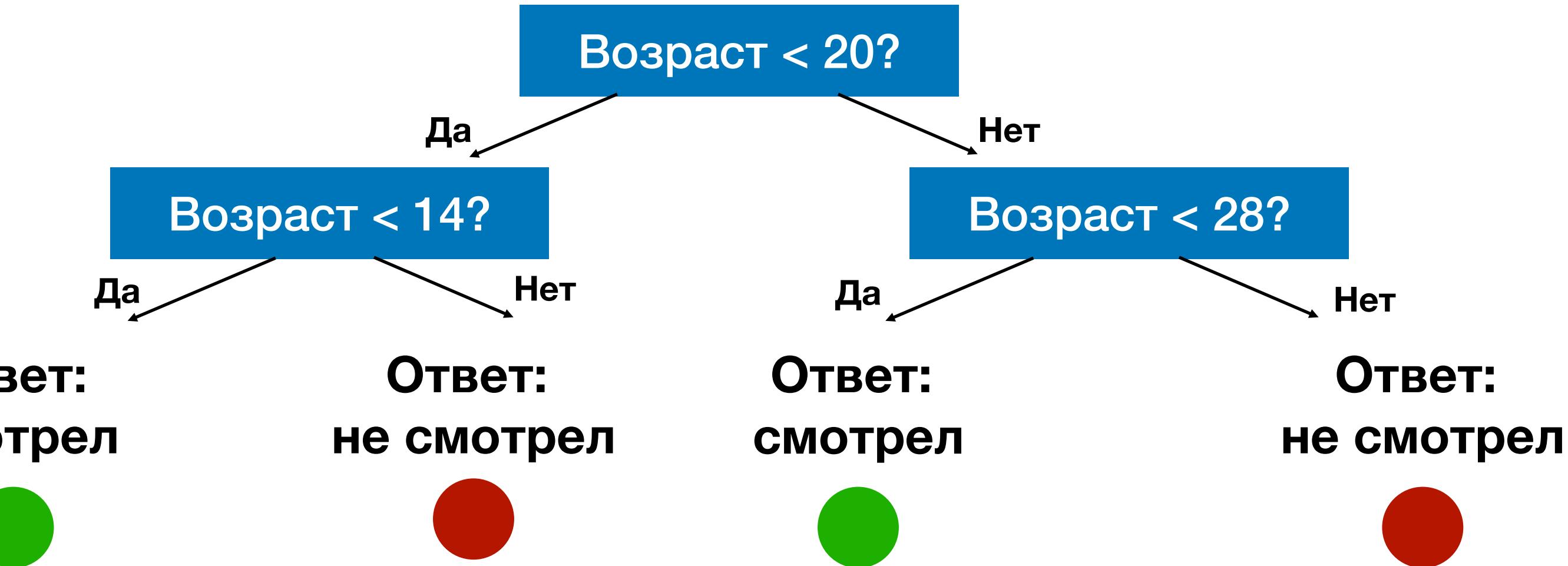
Дмитрий Меркушов

| План занятия

1. Recap: решающие деревья
2. Понятие ансамбля моделей
3. Бэггинг
4. Метод случайных подпространств
5. Случайный лес
6. Блендинг
7. Стекинг
8. Практика

Recap: Решающие деревья

- Последовательное построение узлов, разбивающих множество входящих объектов согласно принципу минимизации “нечистоты” (impurity) узла
- Предикаты для разбиения в узле выбираются из всего множества признаков
- Деревья могут легко переобучиться под выборку, если не ограничивать их глубину
- Деревья очень чувствительны к небольшим изменениям в выборке (шумам)



| Интуиция: жюри присяжных

- Пусть имеется жюри из N присяжных, принимающих решение независимо друг от друга
- Пусть вероятность “правильного” приговора у каждого присяжного = p
- Чему равна вероятность “правильного” приговора жюри, если голосуют “большинством”?



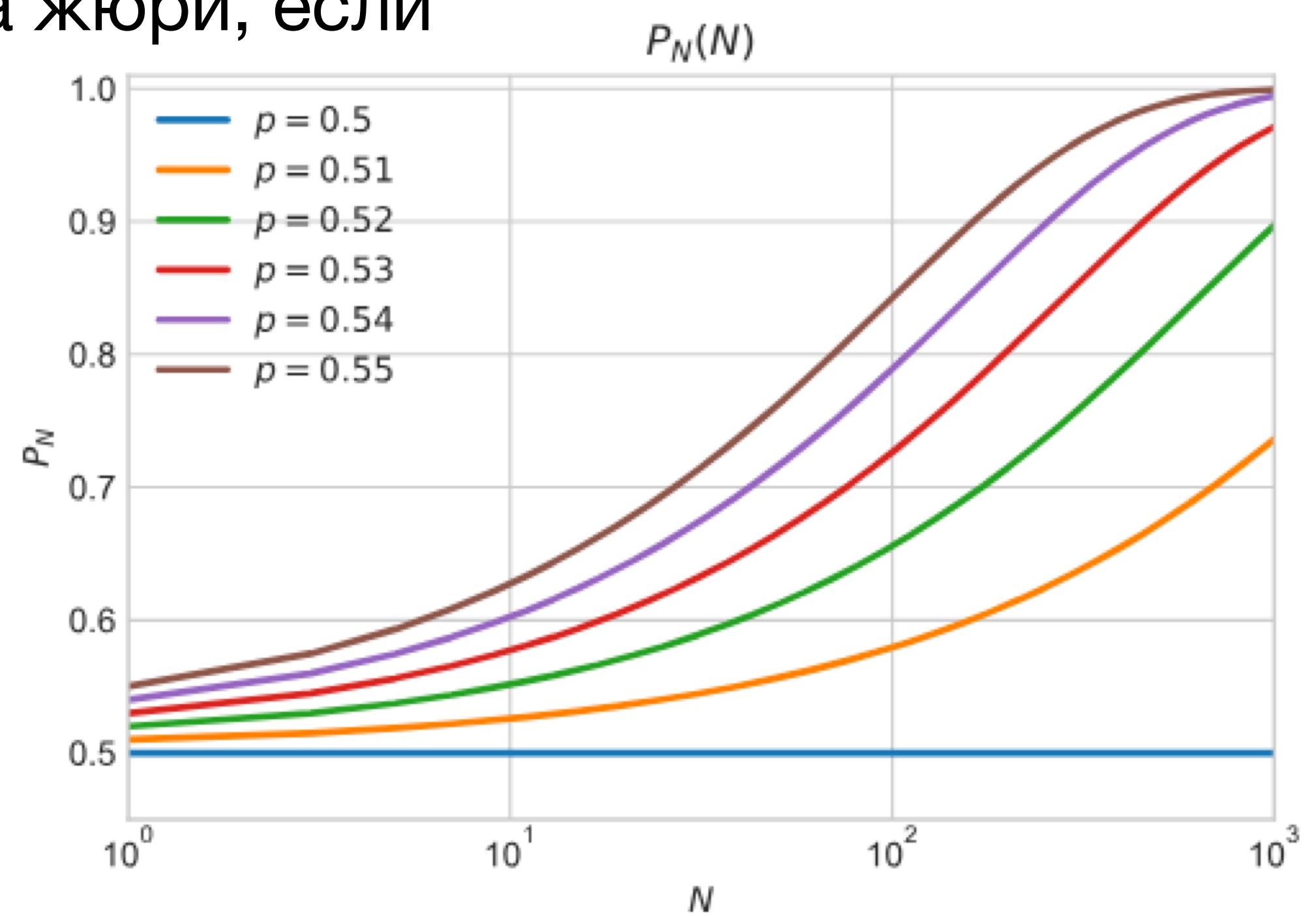
| Интуиция: жюри присяжных



- Пусть имеется жюри из N присяжных, принимающих решение независимо друг от друга
- Пусть вероятность “правильного” приговора у каждого присяжного = p
- Чему равна вероятность “правильного” приговора жюри, если голосуют “большинством”?

$$P_N = \sum_{i=[N/2]+1}^N C_N^i p^i (1-p)^{N-i}$$

- При $p > 0.5$ вероятность верного ответа стремится к 1 по мере увеличения N



Ансамбли

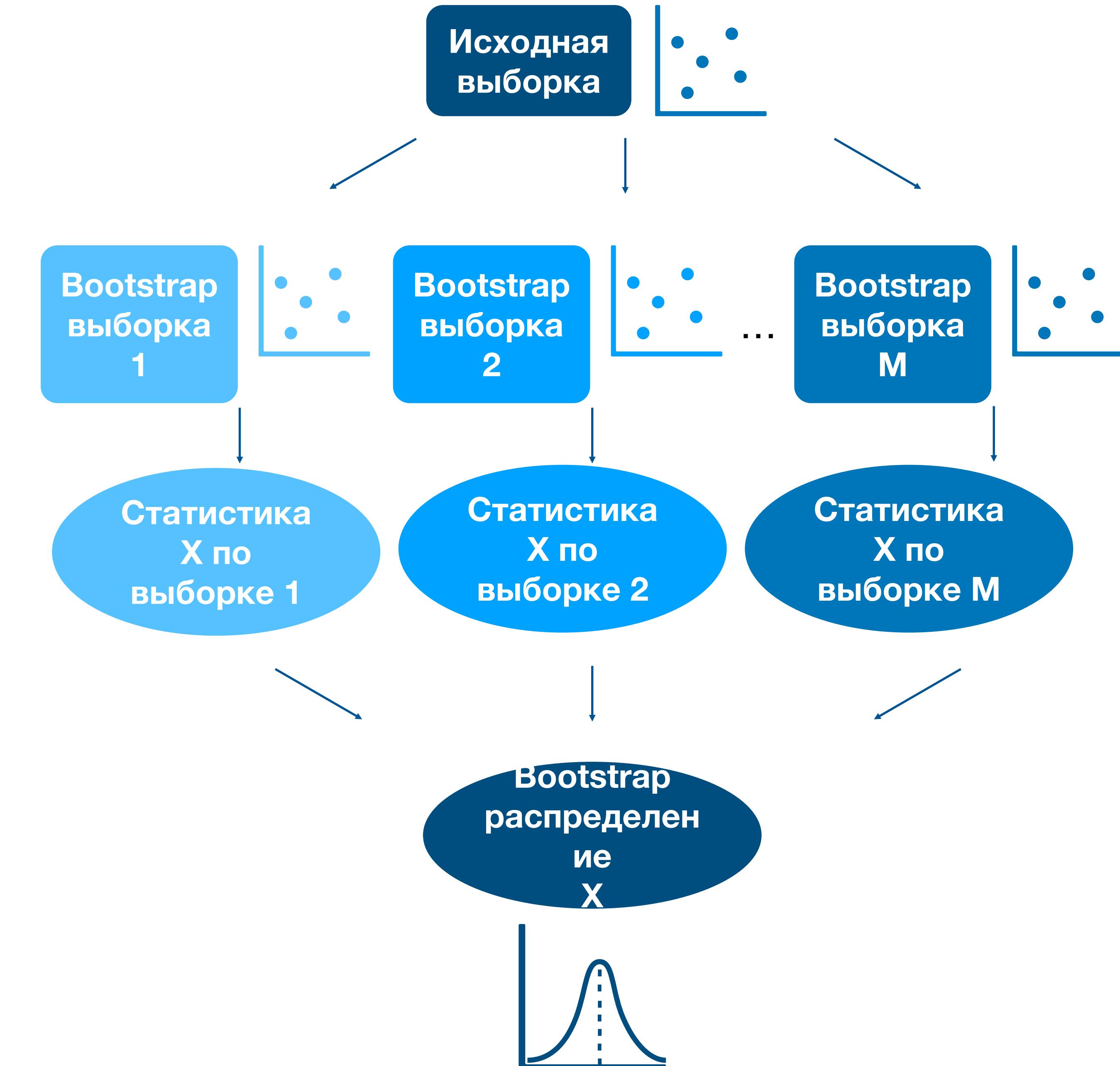
- В машинном обучении подход, при котором для предсказания используются сразу несколько моделей, называется **ансамблированием (ensembling)**
- На этой лекции рассмотрим
 - два варианта ансамблирования - **бэггинг (Bagging)** и его развитие с использованием деревьев решений - **случайный лес**
 - **Stacking** и **Blending**

Ансамбли

- Пусть на выборке (X, y) обучены базовые модели: $b_1(x), b_2(x), \dots, b_m(x)$
- Для “учета мнения” всех базовых алгоритмов ансамбля применяются следующие подходы:
 - Простое голосование (Simple Voting): $f(x) = \frac{1}{m} \sum_{i=1}^m b_i(x)$
 - Взвешенное голосование (Weighted Voting): $f(x) = \frac{1}{m} \sum_{i=1}^m w_i b_i(x)$
 - Смесь экспертов (Mixture of Experts): $f(x) = \frac{1}{m} \sum_{i=1}^m w_i(x) b_i(x)$
 - *Ансамбль с какими базовыми моделями будет лучше: с похожими или различными?*

Bootstrap

- Bagging = Bootstrap Aggregation
- **Bootstrap** (“бутстреп”) - метод из математической статистики, который позволяет оценивать параметры генеральной совокупности по выборке из нее
- Есть выборка из N элементов, нужно оценить статистику X
 - Делаем из выборки M подвыборок по N элементов с возвращением, получаем оценку статистики X на каждой подвыборке
 - Получаем выборочное распределение статистики X



Bootstrap: пример

- Задача: пусть θ - некоторый параметр генеральной совокупности, для которого нужно оценить стандартную ошибку (то есть ищем $Std(\theta)$)
- В распоряжении у нас есть только случайная выборка X размера N : (x_1, x_2, \dots, x_N)
- По выборке X можно сделать оценку для θ : $\langle \theta \rangle_N$
- Как оценить $Std(\theta)$?

- Сгенерируем n различных bootstrap-выборок (размера N) с возвращением:

$$(X_1^b, X_2^b, \dots, X_n^b)$$

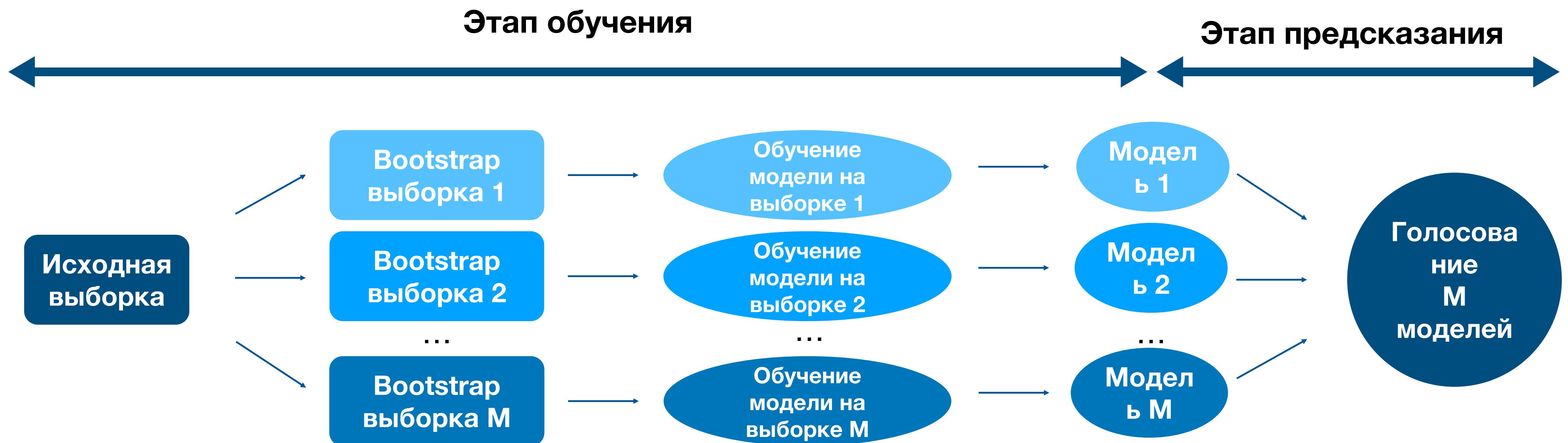
- Сделаем оценку параметра θ для каждой из n выборок, получим набор из n значений: $(\langle \theta \rangle_1^b, \langle \theta \rangle_2^b, \dots, \langle \theta \rangle_n^b)$
- Тогда bootstrap-оценкой стандартной ошибки θ будет

$$Std(\theta) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\langle \theta \rangle_i^b - \langle \theta \rangle)^2}$$

- Как будет вести себя распределение выборочных оценок θ при стремлении n к бесконечности?

Бэггинг (Bagging)

- Бэггинг - способ построения ансамблей, при котором базовые модели обучаются на **различных bootstrap-подвыборках исходной обучающей выборки**
- Использование различных подмножеств обучающей выборки повышает “различность” базовых алгоритмов



Бэггинг (Bagging)

- “Усреднение” ответов отдельных моделей приводит к улучшению качества за счет “компенсации” их ошибок
- Стратегия выбора объектов приводит к повышению устойчивости к выбросам (всегда найдутся базовые модели, которые “не видели” данный конкретный выброс)



| Out-of-bag-score (OOB)

- Использование бэггинга позволяет оценивать модели без дополнительных (валидационных) данных
- Т.к. для обучения каждой базовой модели используется подвыборка, сэмплированная из исходной выборки с возвращением, **то есть объекты, которые в нее не попали (они называются Out-of-bag, OOB)**
- Оценка качества модели на OOB-объектах называется **OOB-score**
- Чему примерно равен размер тестовой выборки в этом случае?
 - Всего объектов N, будем брать N объектов с возвращением
 - Вероятность взять каждый объект на каждом шаге:

| Out-of-bag-score (OOB)

- Использование бэггинга позволяет оценивать модели без дополнительных (валидационных) данных
- Т.к. для обучения каждой базовой модели используется подвыборка, сэмплированная из исходной выборки с возвращением, **то есть объекты, которые в нее не попали (они называются Out-of-bag, OOB)**
- Оценка качества модели на OOB-объектах называется **OOB-score**
- Чему примерно равен размер тестовой выборки в этом случае?
 - Всего объектов N, будем брать N объектов с возвращением
 - Вероятность взять каждый объект на каждом шаге: $P = \frac{1}{N}$
 - Вероятность взять объект в подвыборку:

| Out-of-bag-score (OOB)

- Использование бэггинга позволяет оценивать модели без дополнительных (валидационных) данных
- Т.к. для обучения каждой базовой модели используется подвыборка, сэмплированная из исходной выборки с возвращением, **то есть объекты, которые в нее не попали (они называются Out-of-bag, OOB)**
- Оценка качества модели на OOB-объектах называется **OOB-score**
- Чему примерно равен размер тестовой выборки в этом случае?
 - Всего объектов N, будем брать N объектов с возвращением
 - Вероятность взять каждый объект на каждом шаге: $P = \frac{1}{N}$
 - Вероятность взять объект в подвыборку: $P_1 = 1 - (1 - \frac{1}{N})^N \rightarrow 1 - \frac{1}{e} \approx 0.63$
 - Итого, примерно 37% объектов не попадают в обучающую подборку каждого дерева

Recap: Bias & variance

- Пусть дана целевая переменная:

$$y(x) = f(x) + \epsilon, \epsilon \sim N(0, \sigma^2), \mathbb{E}[y] = f$$

- На некоторой выборке обучается модель:

$$a = a(x)$$

- ожидание квадрата ошибки модели:

$$\mathbb{E}[(y - a)^2] = \mathbb{E}[y^2 - 2ay + a^2] = \mathbb{E}[y^2] - 2\mathbb{E}[ay] + \mathbb{E}[a^2] =$$

$$\mathbb{E}[y^2] - 2f\mathbb{E}[a] + \mathbb{E}[a^2] = \mathbb{E}[y^2] - 2f\mathbb{E}[a] + \mathbb{E}[a^2] +$$

$$((\mathbb{E}[y])^2 - (\mathbb{E}[y])^2) + ((\mathbb{E}[a])^2 - (\mathbb{E}[a])^2) =$$

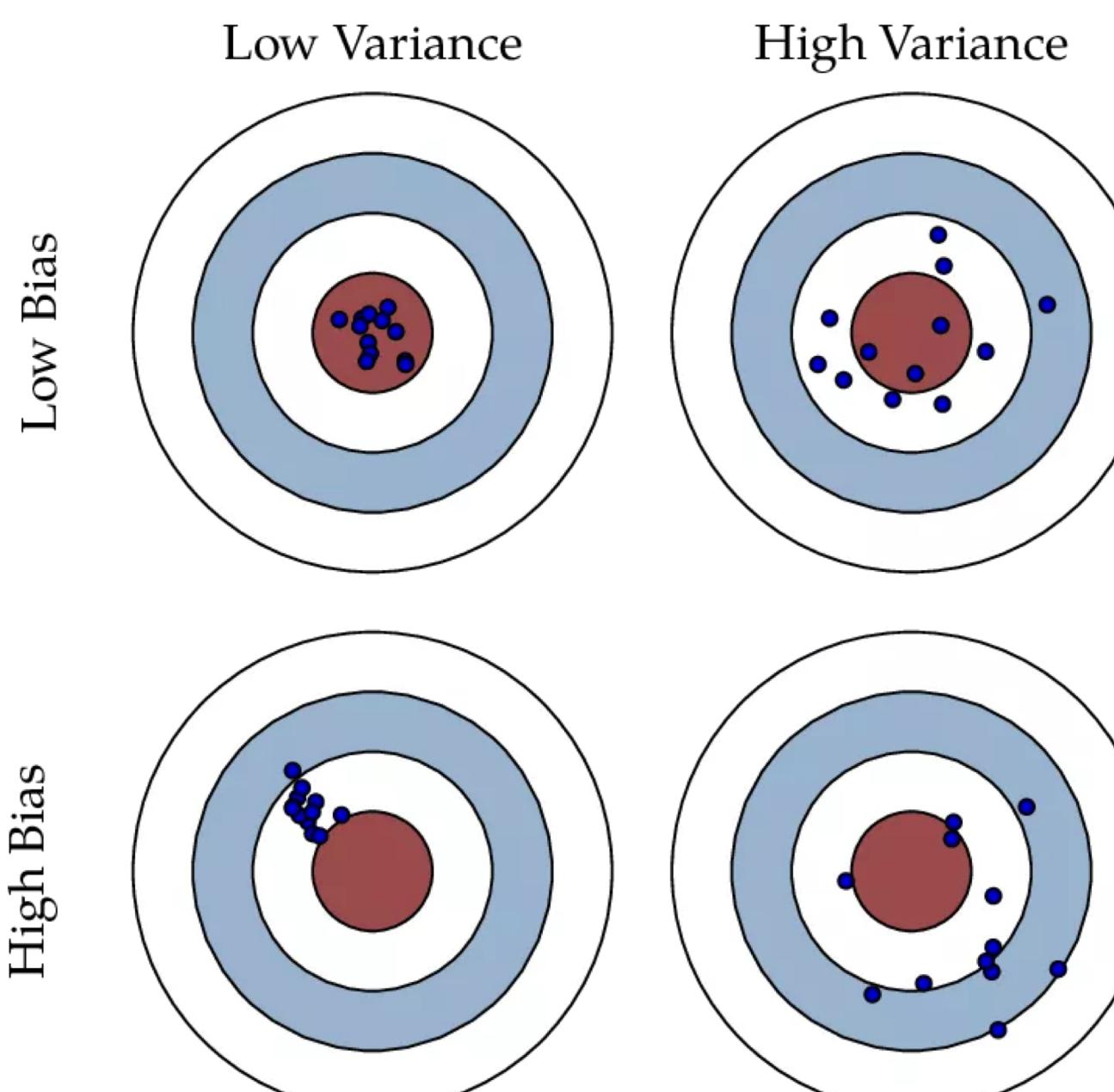
$$(\mathbb{E}[y^2] - (\mathbb{E}[y])^2) + (\mathbb{E}[a^2] - (\mathbb{E}[a])^2) + (\mathbb{E}[y])^2 - 2f\mathbb{E}[a] + (\mathbb{E}[a])^2 =$$

$$\boxed{\mathbb{D}[y]} + \boxed{\mathbb{D}[a]} + \boxed{(\mathbb{E}[f - a])^2}$$

Разброс целевой переменной
("неустранимая ошибка")

"Разброс" модели
(Variance)

Квадрат "смещения" модели
(Bias)



Bias & variance & bagging

- Ансамблирование моделей позволяет получить модель с лучшими Bias/Variance по сравнению с базовыми алгоритмами
- При использовании бэггинга из M моделей в задаче регрессии **разброс (Variance) уменьшается в M раз при условии некоррелированности ошибок базовых моделей**
- При этом смещение у ансамбля моделей такое же, как у одной модели
- Решающие деревья выглядят неплохими кандидатами на роль базовых моделей для бэггинга
 - *Почему?*

Bias & variance & bagging

- Ансамблирование моделей позволяет получить модель с лучшими Bias/Variance по сравнению с базовыми алгоритмами
- При использовании бэггинга из M моделей в задаче регрессии **разброс (Variance) уменьшается в M раз при условии некоррелированности ошибок базовых моделей**
- При этом смещение у ансамбля моделей такое же, как у одной модели
- Решающие деревья выглядят неплохими кандидатами на роль базовых моделей для бэггинга
 - *Почему?*
 - Каждое дерево в отдельности может полностью “выучить” выборку ($\text{Bias}=0$), но при этом иметь высокий разброс - при объединении в ансамбль мы сохраним низкий Bias, но многократно уменьшим Variance

Метод случайных подпространств

- Метод случайных подпространств (**Random Subspace Method**) - способ построения ансамблей, при котором базовые модели обучаются на **различных наборах признаков** исходной обучающей выборки
- Использование различных подмножеств признаков обучающей выборки также повышает “различность” базовых алгоритмов

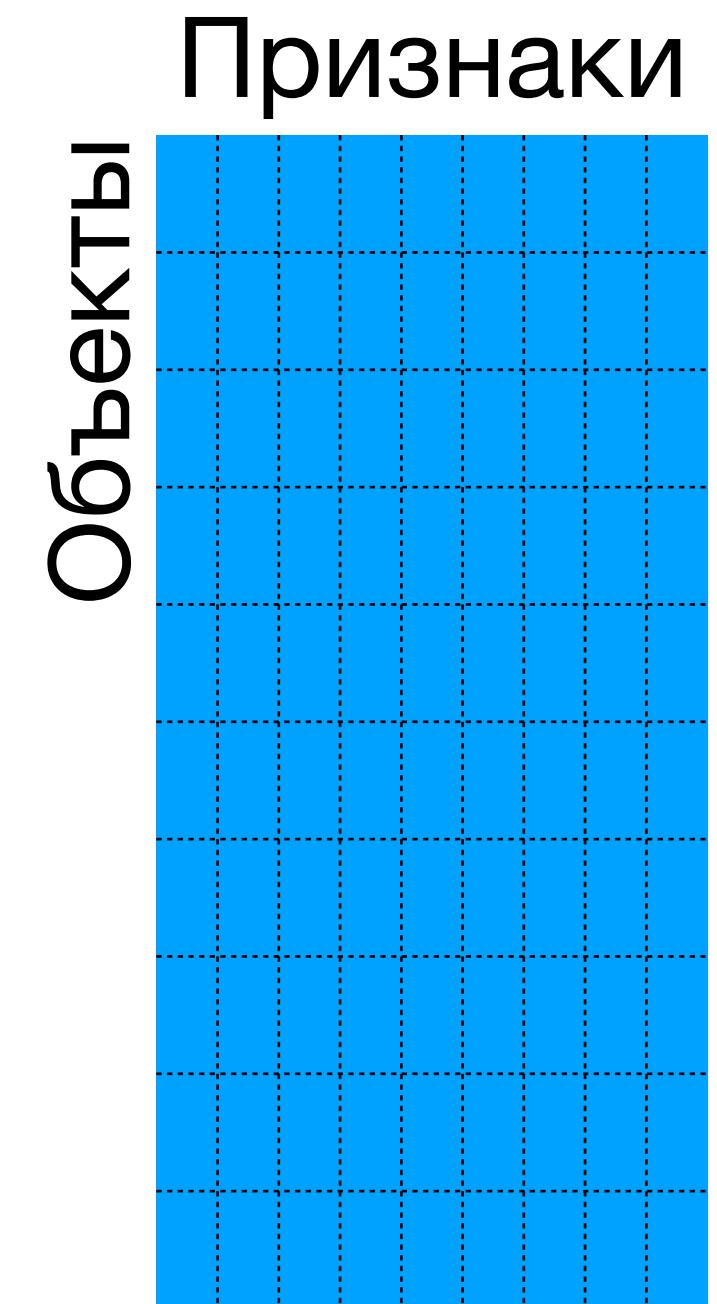


Метод случайных подпространств

- Каждый базовый алгоритм вынужден строить предсказания по “неполным данным” об объектах
- Может улучшить качество по сравнению с одной моделью, если признаки сильно разнородны

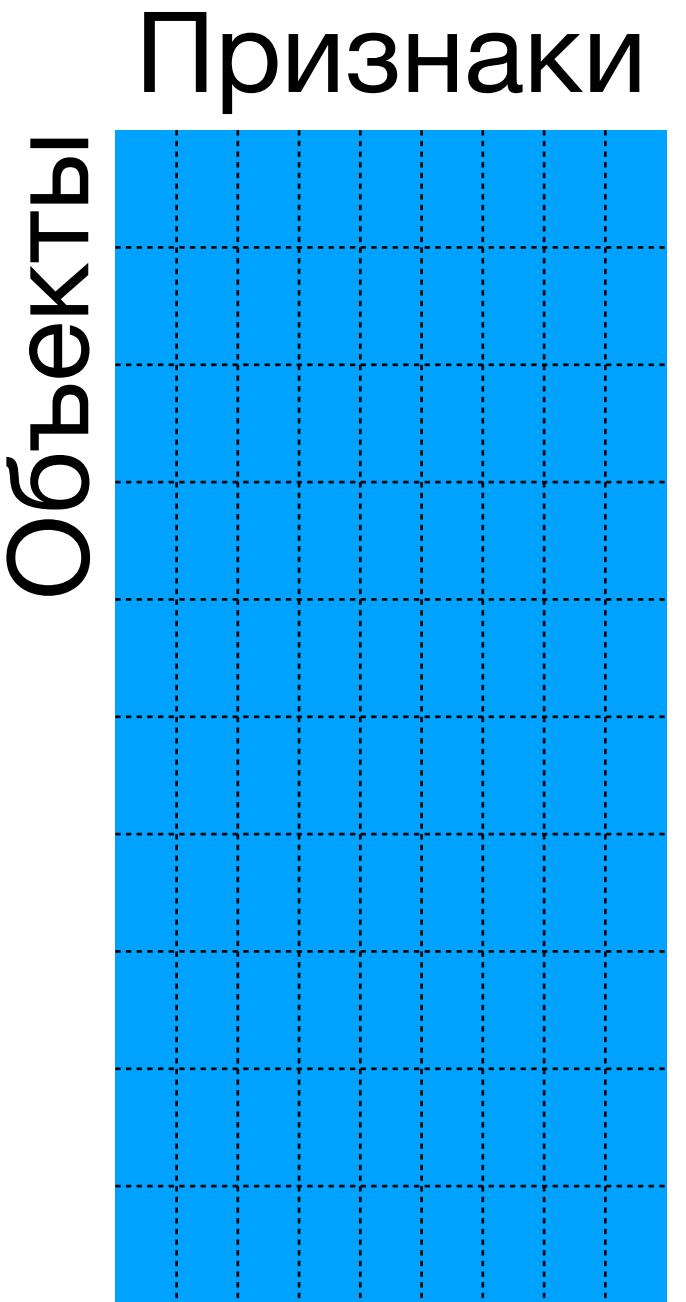
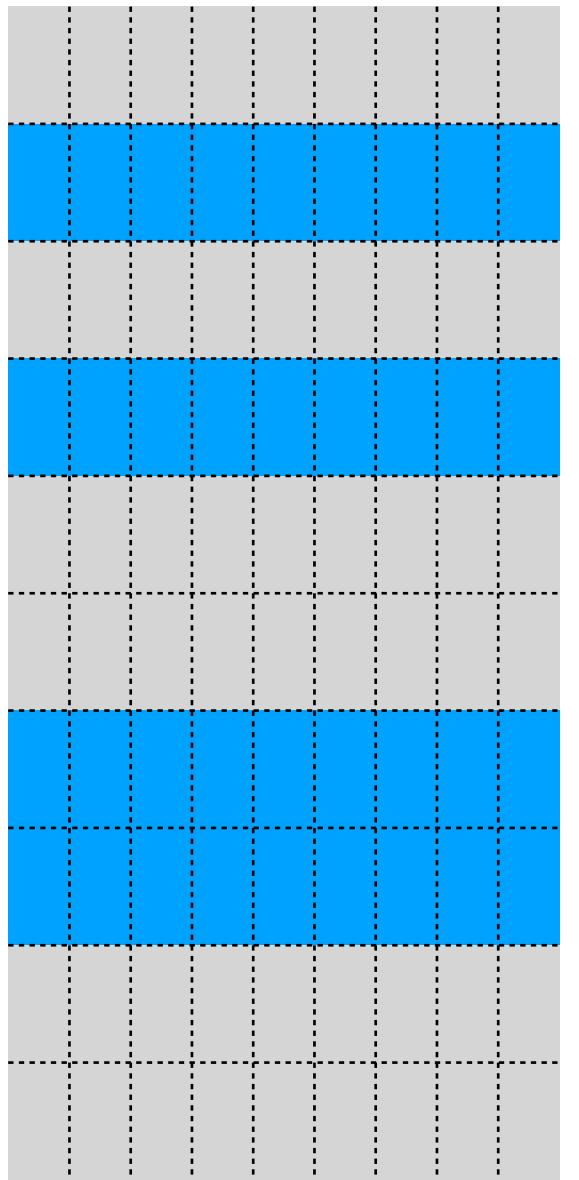


| Bagging & RSM



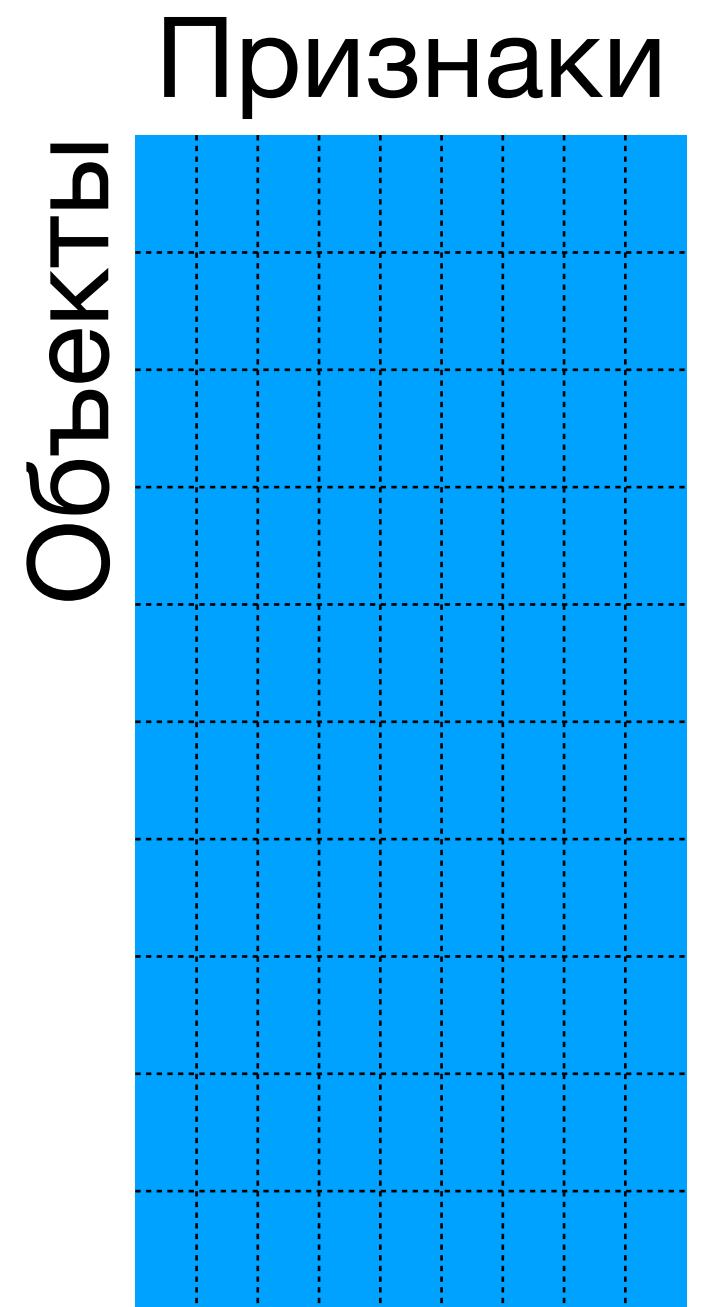
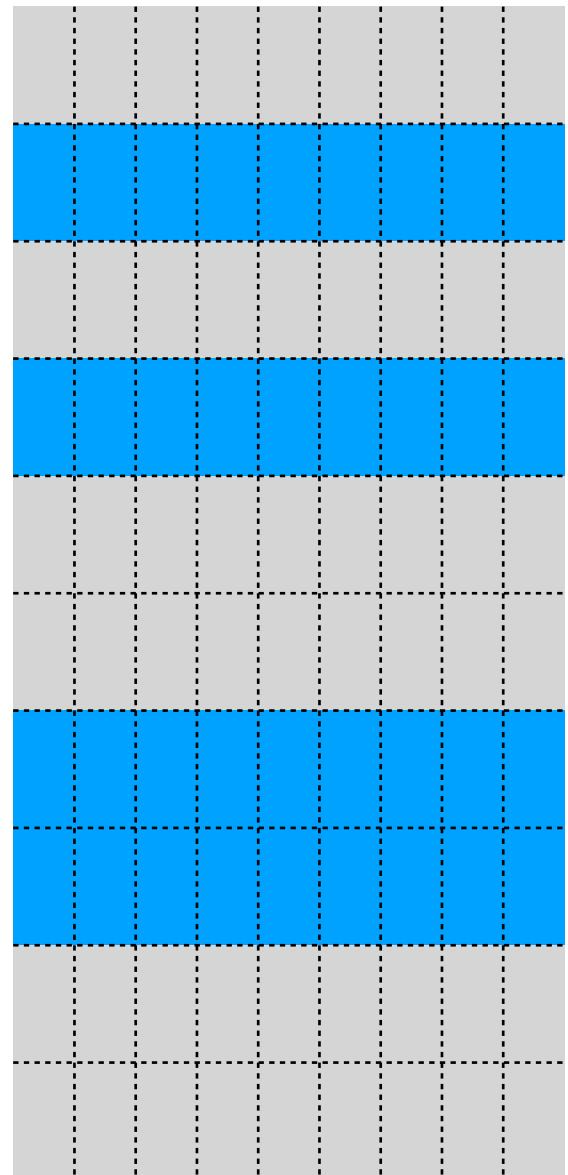
| Bagging & RSM

Бэггинг
Подмножества объектов
Все признаки

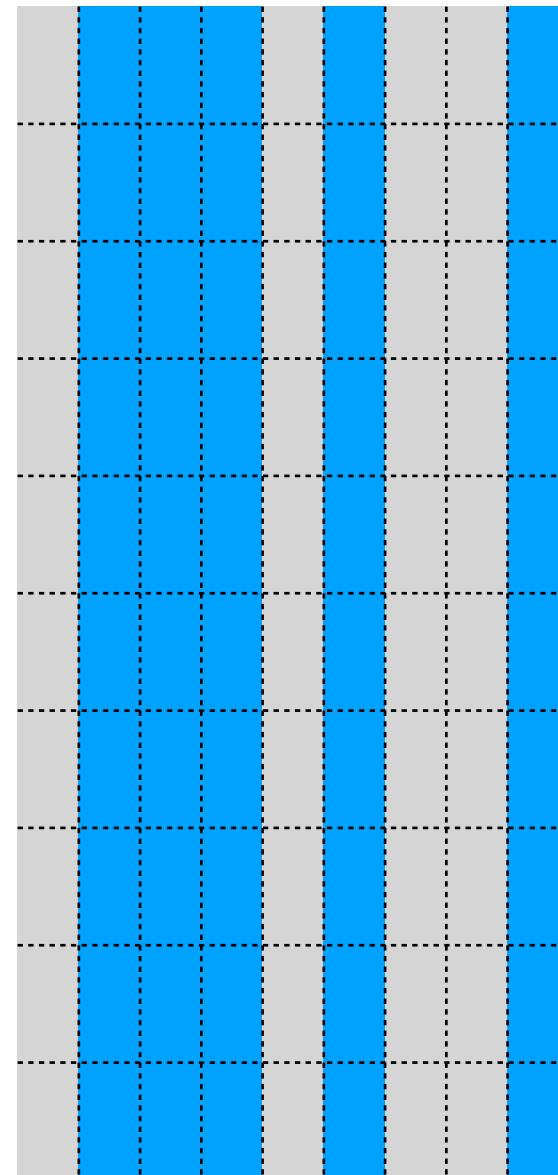


| Bagging & RSM

Бэггинг
Подмножества объектов
Все признаки

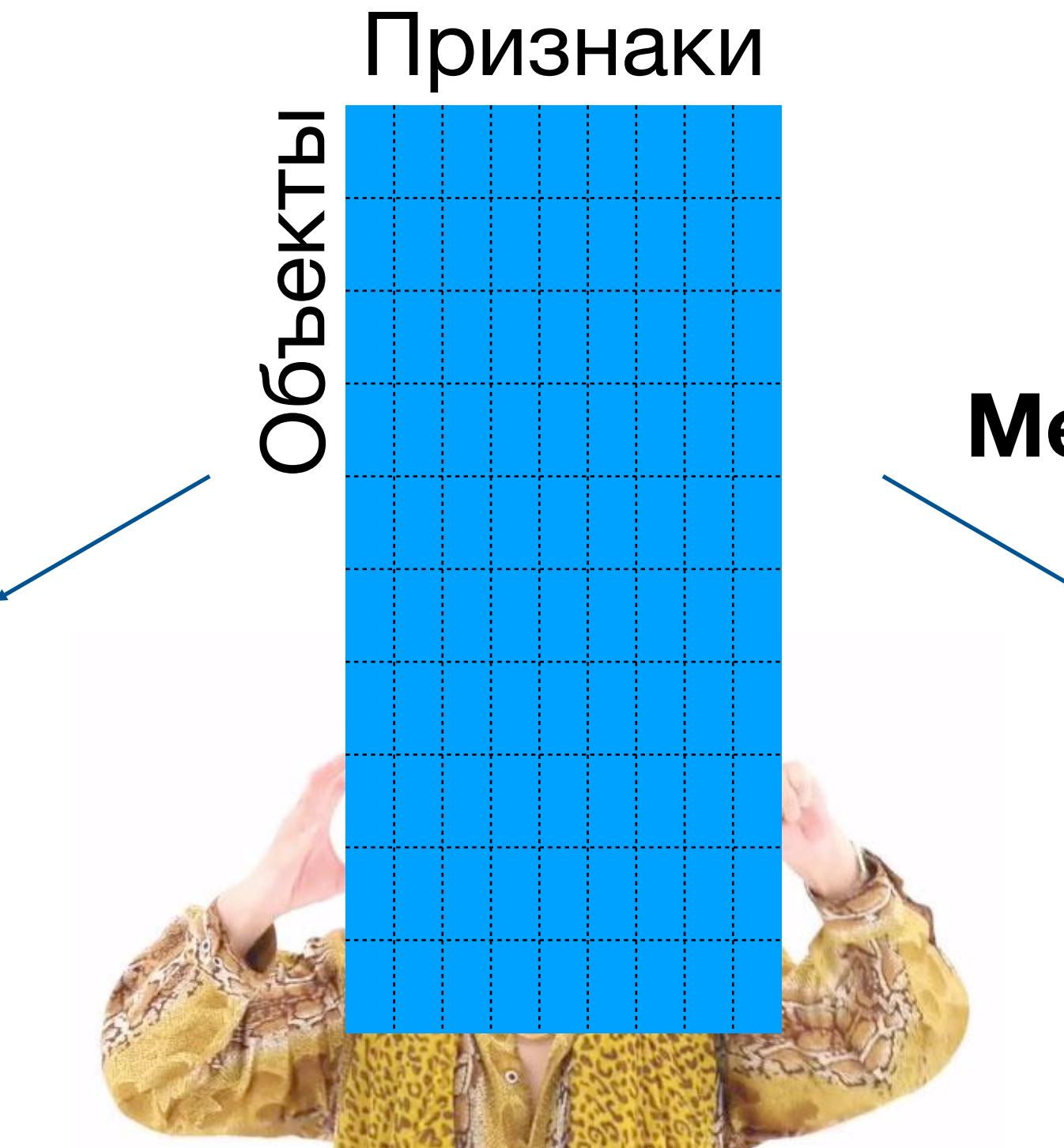
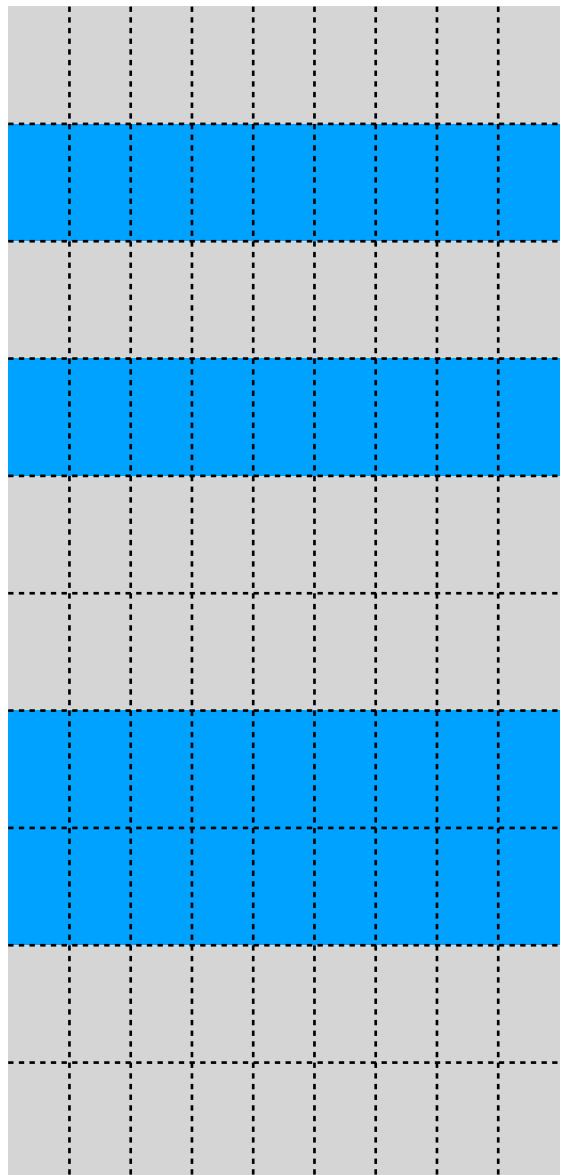


Метод случайных подпространств
Все объекты
Подмножества признаков

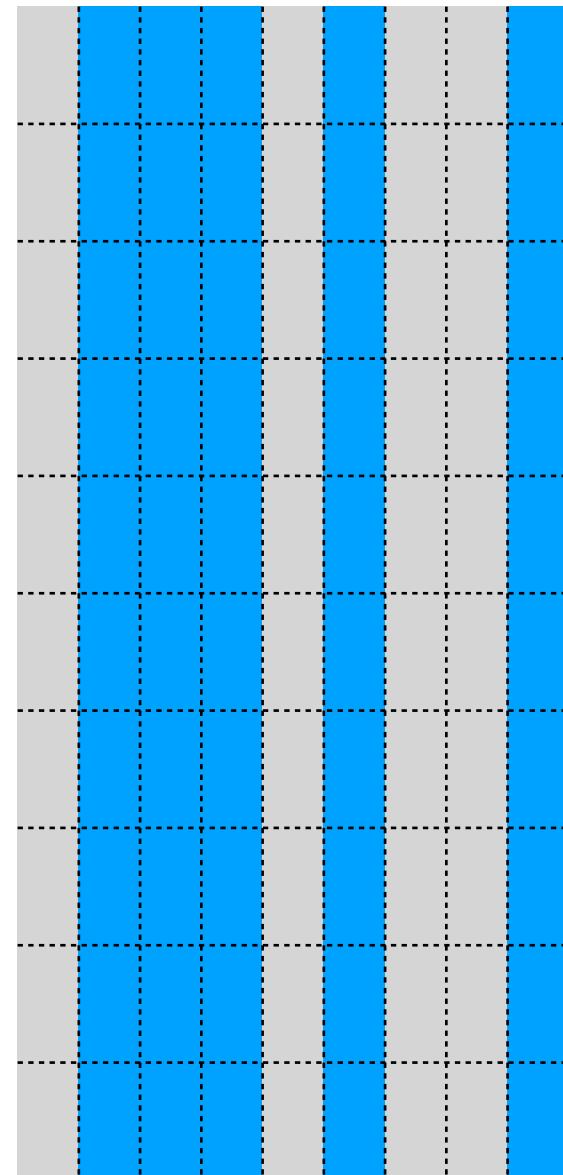


| Bagging & RSM

Бэггинг
Подмножества объектов
Все признаки

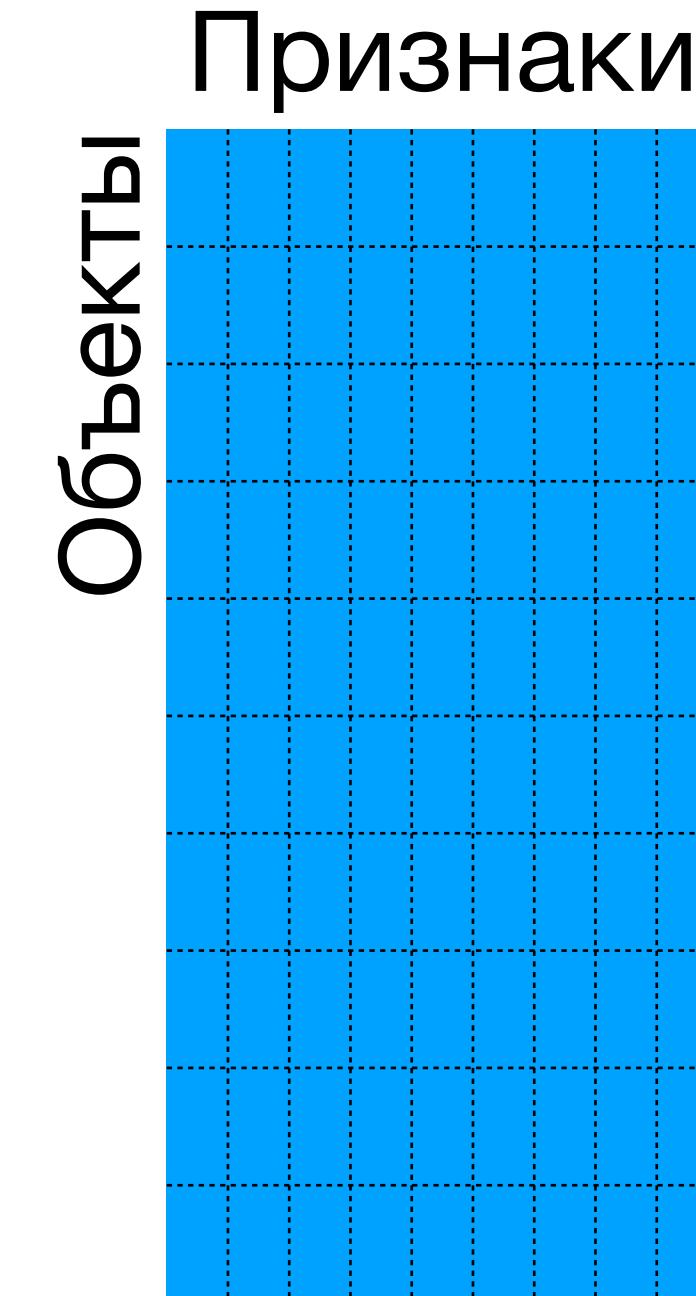
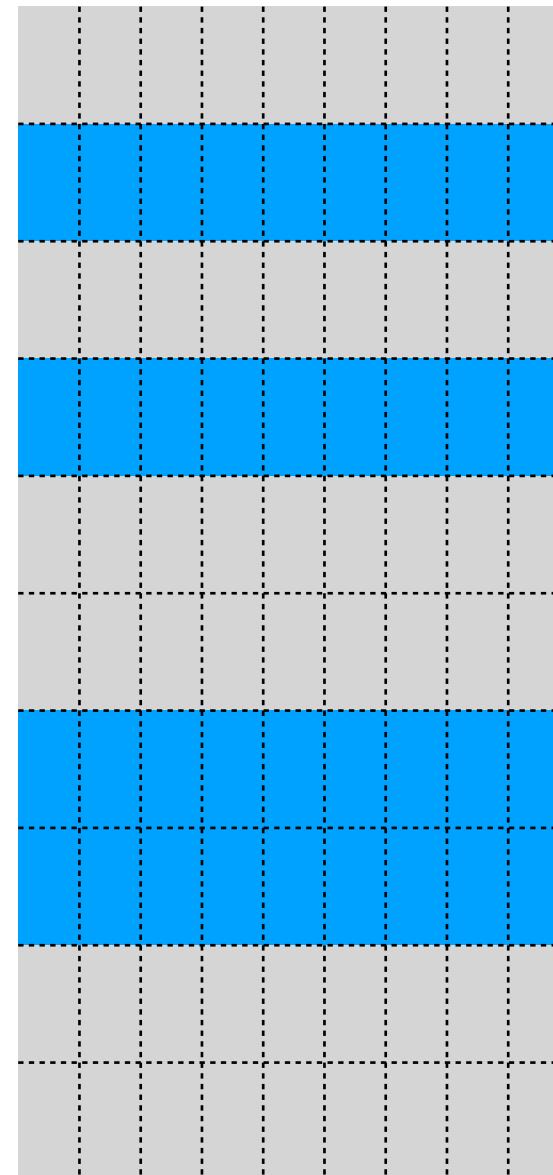


Метод случайных подпространств
Все объекты
Подмножества признаков

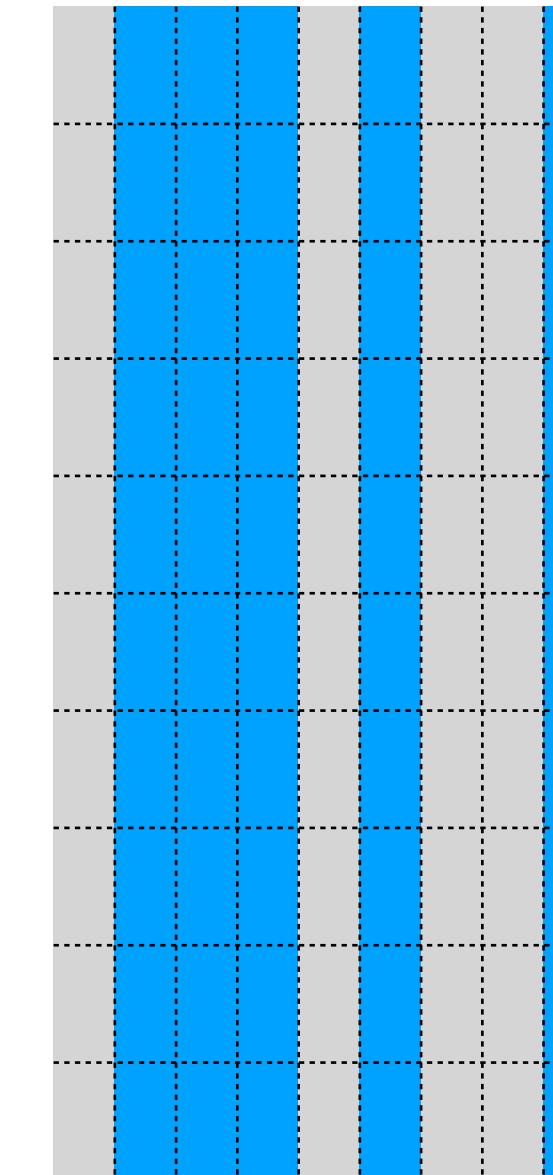
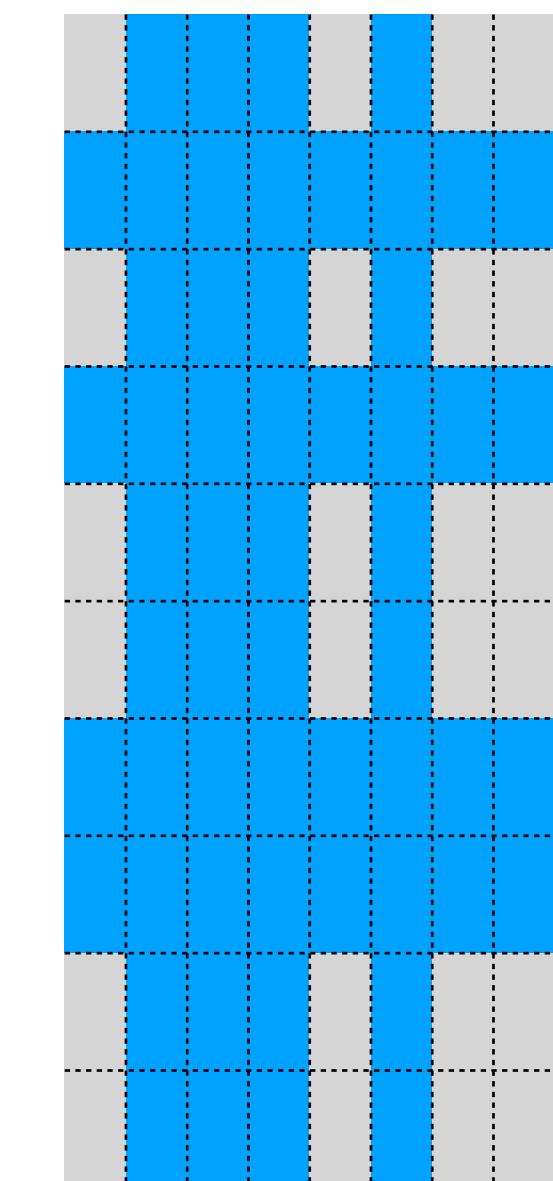


Random Forest

Бэггинг
Подмножества объектов
Все признаки



Метод случайных подпространств
Все объекты
Подмножества признаков



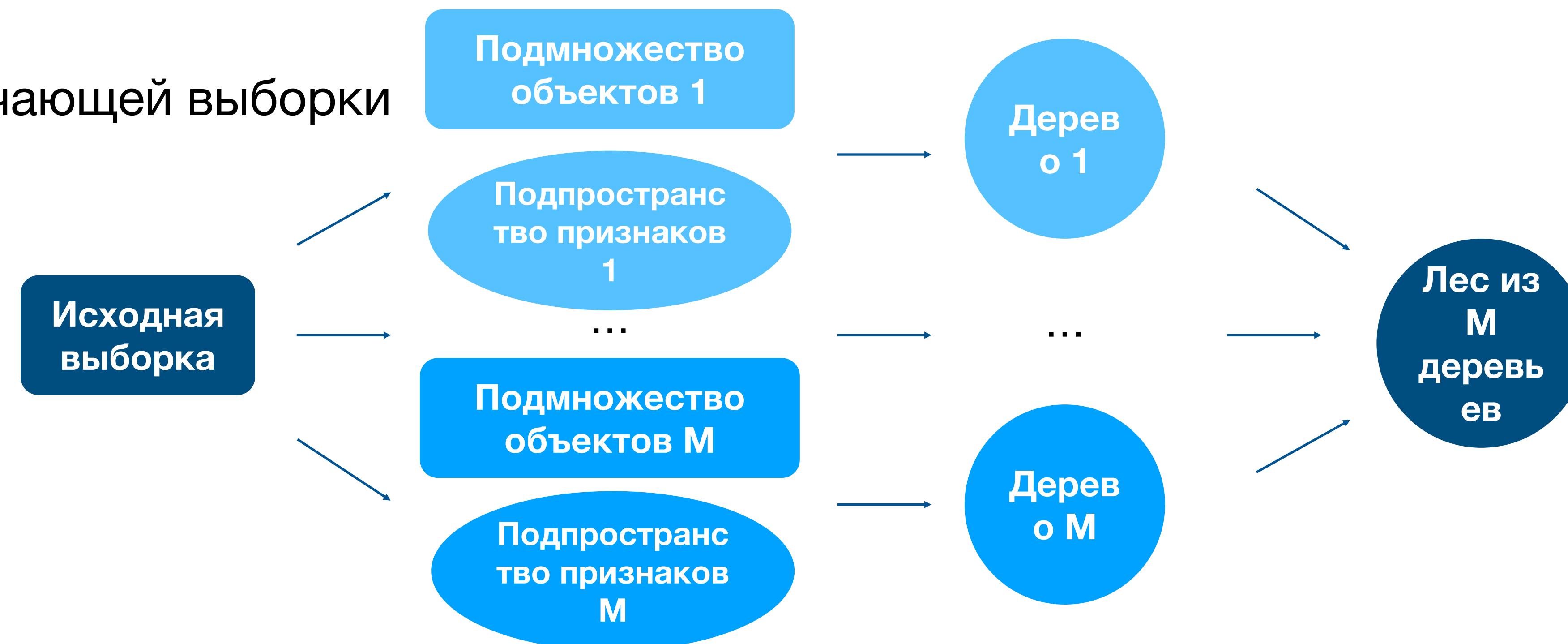
| Случайный лес

- Случайный лес (**Random Forest**) - способ построения ансамблей,

при котором базовые модели (= деревья решений) обучаются

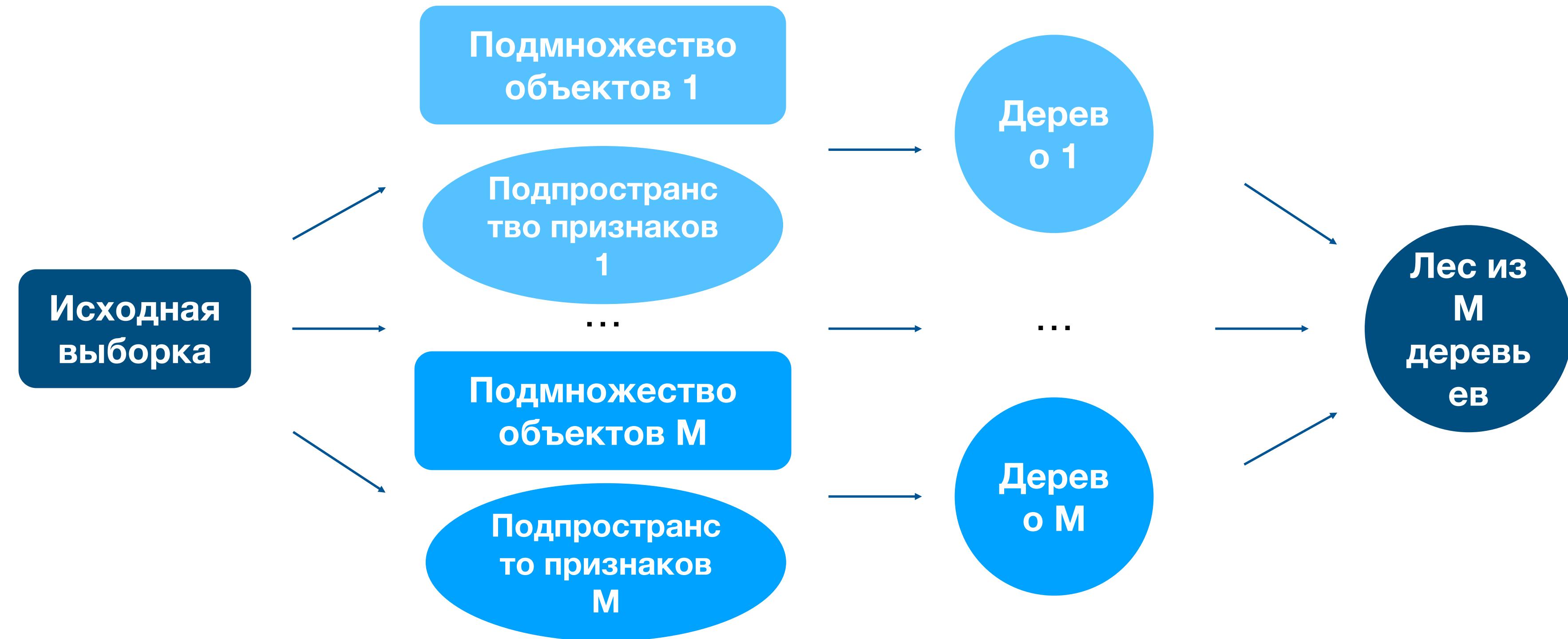
на различных наборах признаков различных подмножеств объектов

исходной обучающей выборки



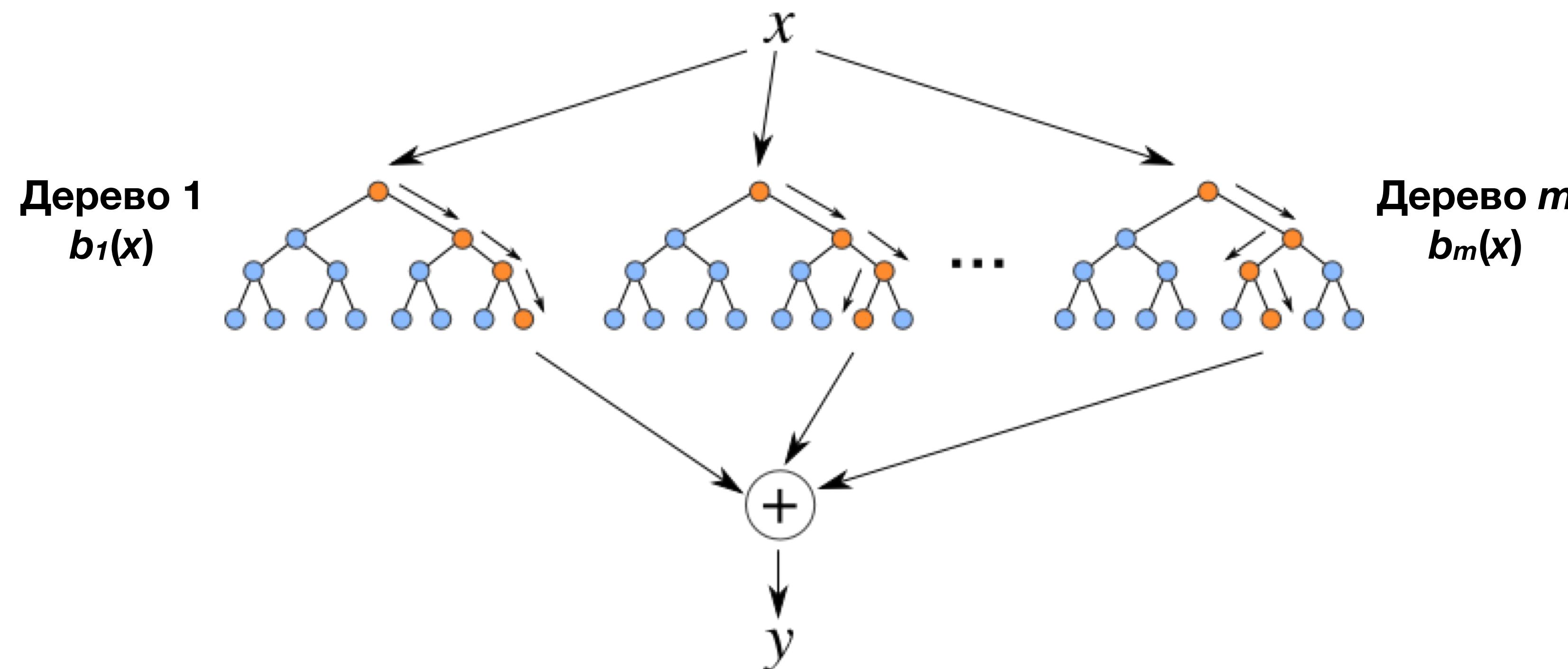
| Случайный лес

- Задаем количество “деревьев в лесу” (базовых моделей) и параметры одного дерева
- Получаем ансамбль (лес) базовых моделей (деревьев)
- Каждое дерево видело только часть объектов (сколько?) и знает только о части их признаков



| Случайный лес

- Предсказание ответа производится путем простого голосования базовых деревьев



$$y = f(x) = \frac{1}{m} \sum_{i=1}^m b_i(x)$$

| Случайный лес

- Настройка случайного леса = настройка параметров 1 дерева + настройка ансамблирования
- Основные параметры одного дерева решений (см. прошлую лекцию):
 - Максимальная глубина
 - Минимальный размер листа
 - ...
- Основные параметры случайного леса:
 - Число деревьев
 - Размер подмножества признаков для обучения одного дерева ($\sim [m^{1/2}]$ для классификации, $\sim [m / 3]$ для регрессии, где m - число признаков)

| Важность признаков

- Вспомним, как происходит принятие решения о разбиении узла дерева:
 - Для исходного узла вычисляется мера его “неоднородности” (impurity)
 - Путем перебора всех доступных признаков выбирается такой предикат для разбиения множества объектов, что уменьшение неоднородности максимально
- Чем чаще и чем “выше” использовался признак, тем он “важнее”
- Рассмотрим 2 метода, позволяющие оценивать важность признаков на основе построенного случайного леса

Оценка важности признаков: метод MDI

- MDI = Mean Decrease in Impurity
- Основан на идее о том, что важные признаки в среднем больше уменьшают неоднородность
- Алгоритм:
 - Обучим на выборке (X, y) случайный лес из деревьев b_1, b_2, \dots, b_M ; F - общее число признаков в X
 - Для каждого признака f_j :
 - Зададим начальную важность: $S_j = 0$
 - Для каждого дерева, построенного с использованием f_j
 - Для каждого узла $node$, в котором f_j использовался для разбиения:
 - $S_j = S_j + N_{node} * G_{node}$ где N_{node} = число объектов в этом узле на этапе обучения, G_{node} = прирост критерия (**gain**)
 - Отнормируем: $S_j = \frac{S_j}{\sum_{k=1}^F S_k}$

Оценка важности признаков: метод MDA

- MDA = Mean Decrease in Accuracy
- Основан на идее о том, что зашумление важного признака ухудшает качество
- Алгоритм:
 - Обучим на выборке (X, y) случайный лес из деревьев b_1, b_2, \dots, b_M ; F - общее число признаков в X
 - Для каждого дерева b_i :
 - Выберем случайный признак f_j , использованный при построении b_i
 - Возьмем ОOB-объекты данного дерева X_i^{oob} , посчитаем метрику качества дерева b_i на X_i^{oob} q_{ji}
 - Случайно перемешаем значения признака f_j , получим $X_i'^{oob}$. Посчитаем метрику качества b_i на $X_i'^{oob}$ q'_{ji}
 - Найдем изменение качества дерева: $S_{ji} = q_{ji} - q'_{ji}$
 - Усредним по всем деревьям уменьшение качества для каждого признака: $S_j = \frac{1}{|B_j|} \sum_{i: b_i \in B_j} S_{ji}$
 - Отнормируем: $S_j = \frac{S_j}{\sum_{k=1}^F S_k}$

Random Forest and Kinect

- В статье на CVPR'11 Microsoft рассказали о принципах работы датчика движений Kinect
- Kinect оборудован камерой, предсказывающей глубину (расстояние до объекта)
- На основе признаков, вычисленных по картам глубины, работает алгоритм (случайного леса!) для классификации частей тела

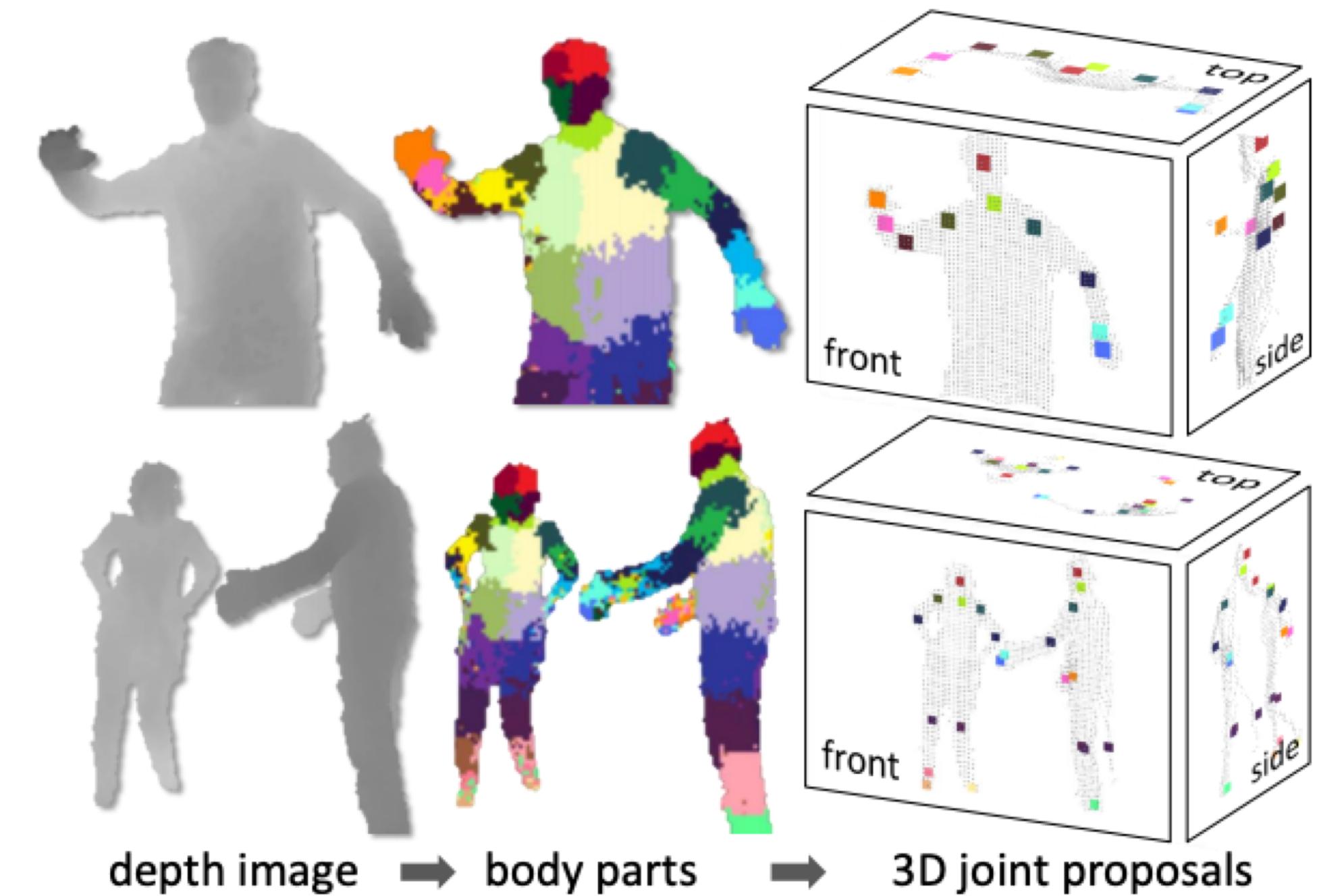


Figure 1. **Overview.** From a single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

Итоги

- **Преимущества случайного леса**
 - Один из сильнейших "классических" алгоритмов машинного обучения
 - Не требователен к обучающей выборке (не требуется нормализация, очистка от шума, ...)
 - Легко параллелируется (т.к. базовые модели обучаются независимо друг от друга)
 - Не подвержен переобучению
 - Дает оценку важности признаков
 - Не требует дополнительной валидационной выборки (за счет OOB-score)
- **Недостатки случайного леса**
 - Как и у решающих деревьев, нет экстраполяции данных
 - Разреженные признаки - ?

| В следующей серии...

- Следует ли смешивать модели? А взбалтывать?

@ mail

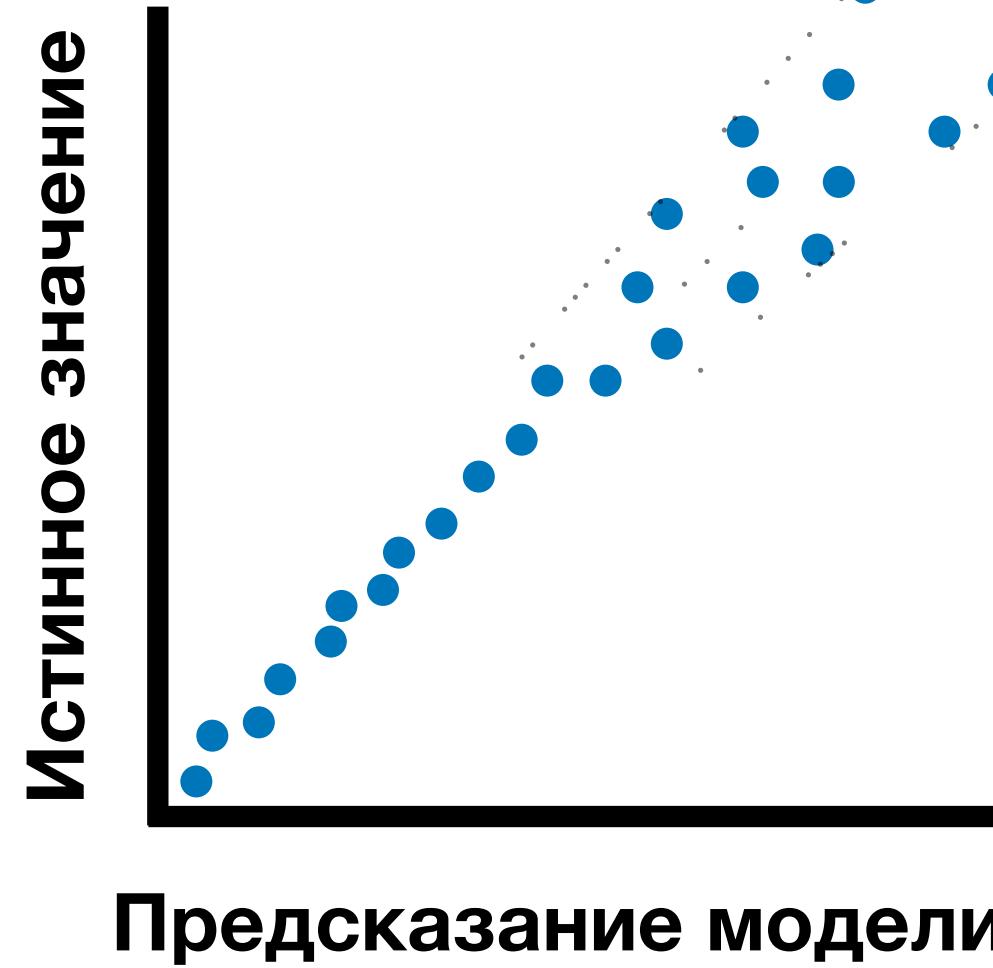
Stacking & Blending

| Интуиция: ансамбли

- В рассмотренных методах построения ансамблей (бэггинг, RSM, случайный лес) решение принимается с помощью усреднения предсказаний
- Кажется, ошибки отдельных моделей должны компенсироваться
- Что будет, если алгоритмы будут работать “хорошо” только в какой-то одной области признакового пространства, а в других - плохо?

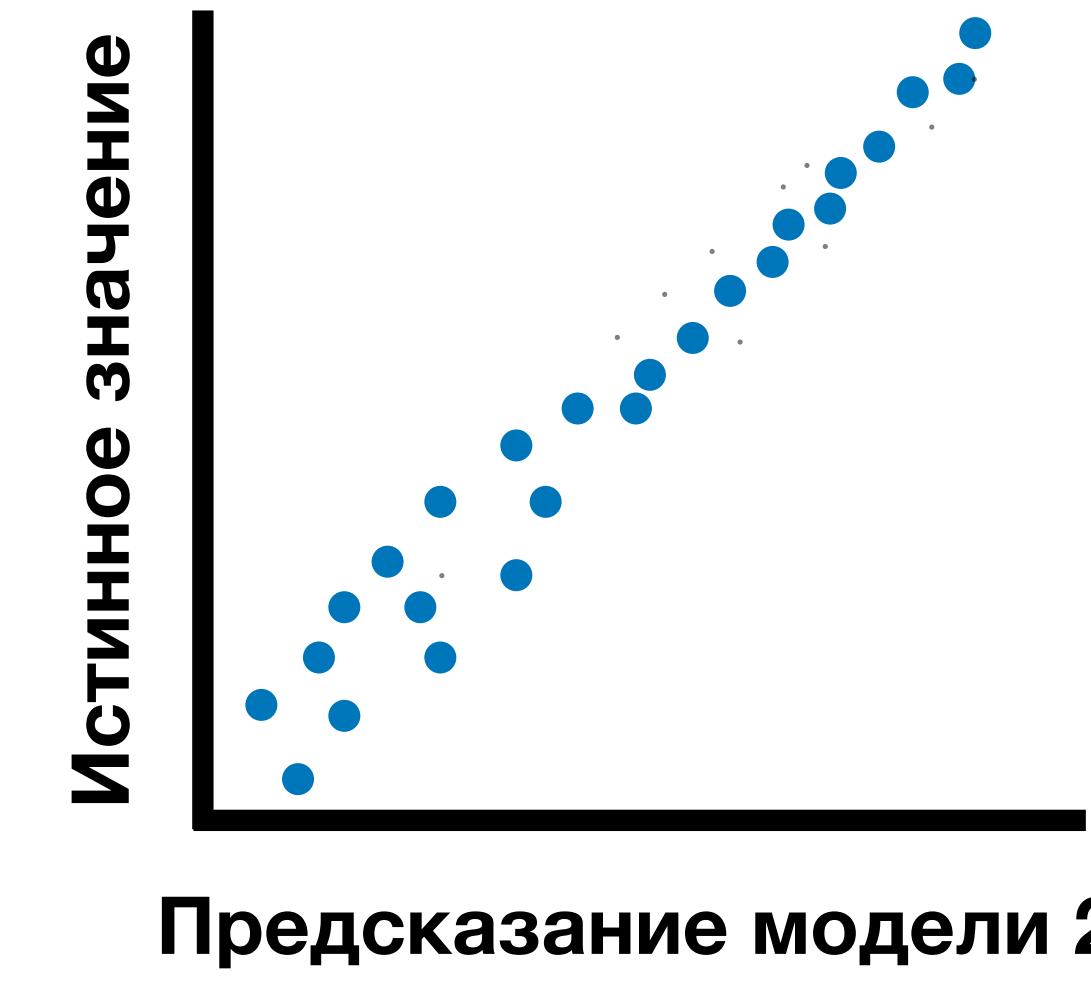
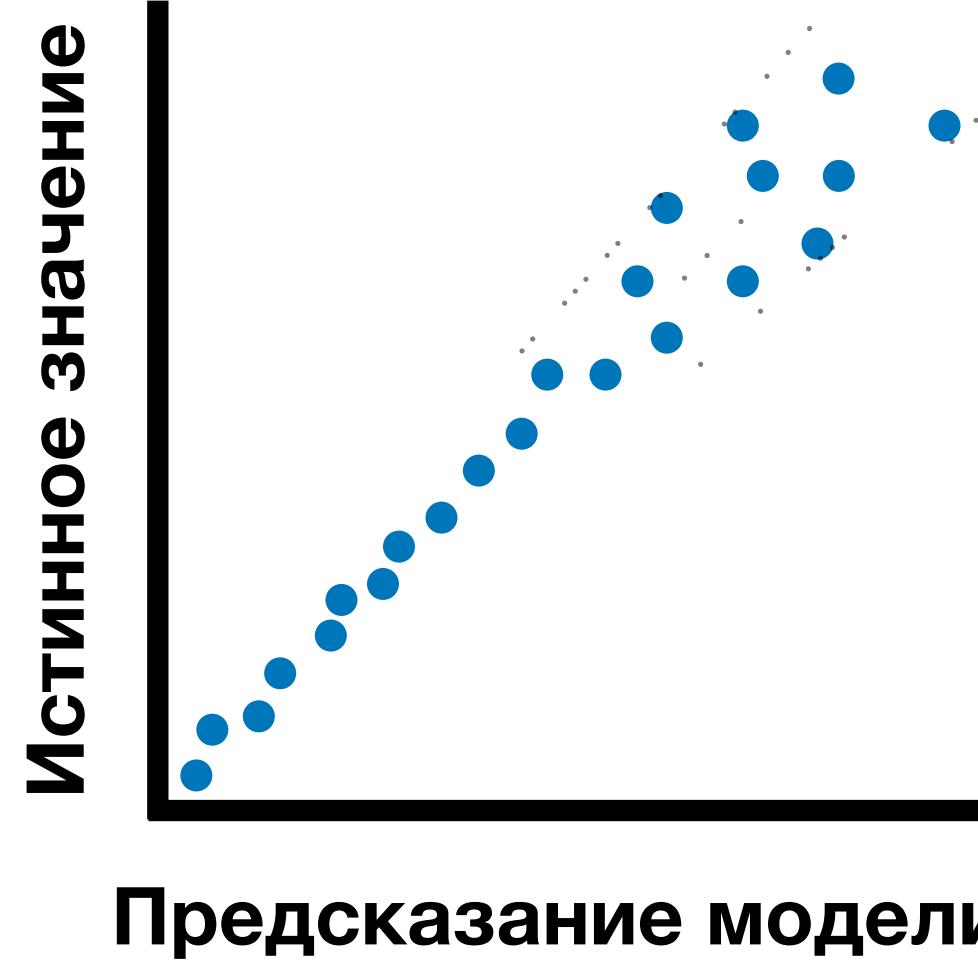
| Интуиция: ансамбли

- В рассмотренных методах построения ансамблей (бэггинг, RSM, случайный лес) решение принимается с помощью усреднения предсказаний
- Кажется, ошибки отдельных моделей должны компенсироваться
- Что будет, если алгоритмы будут работать “хорошо” только в какой-то одной области признакового пространства, а в других - плохо?



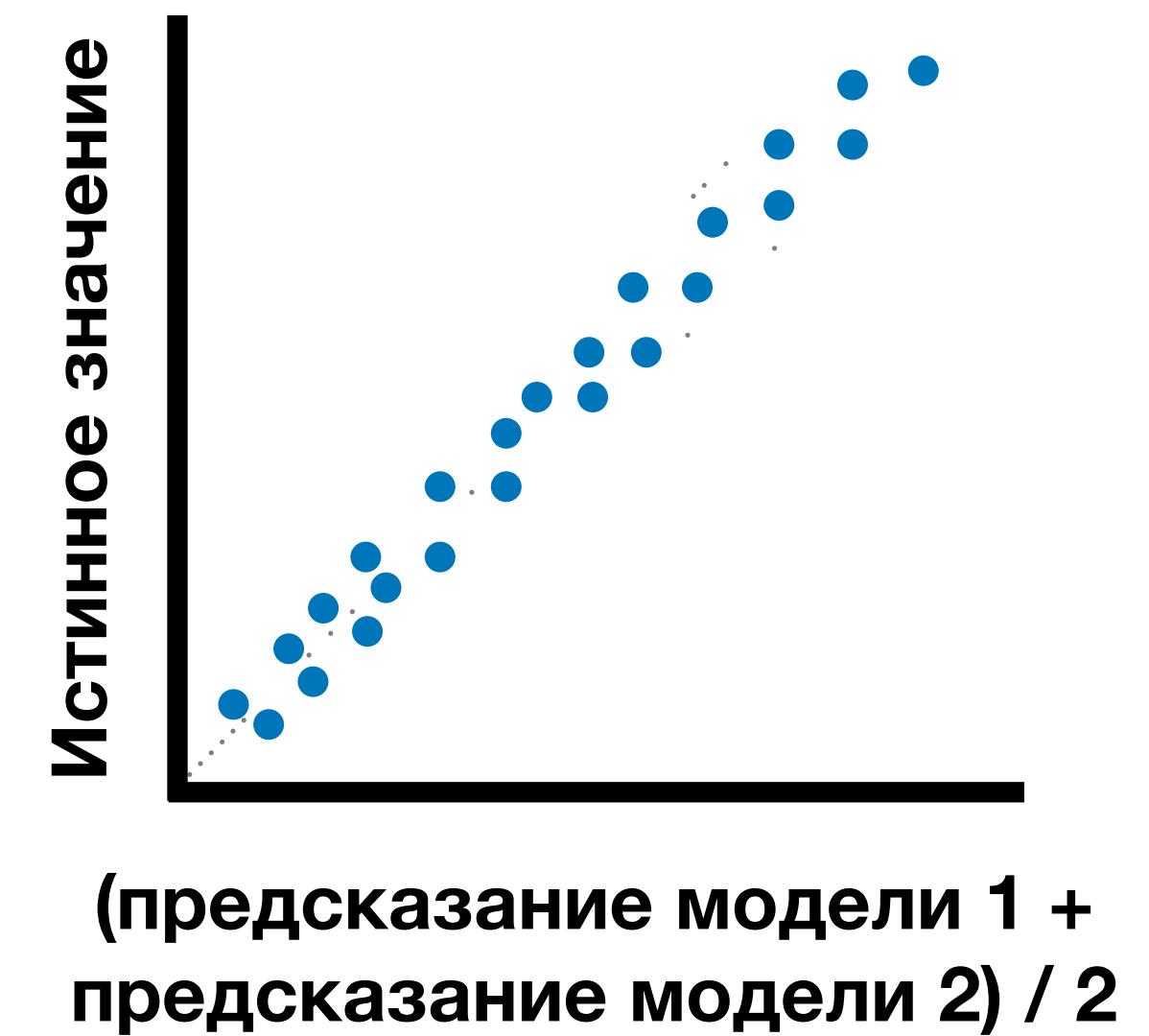
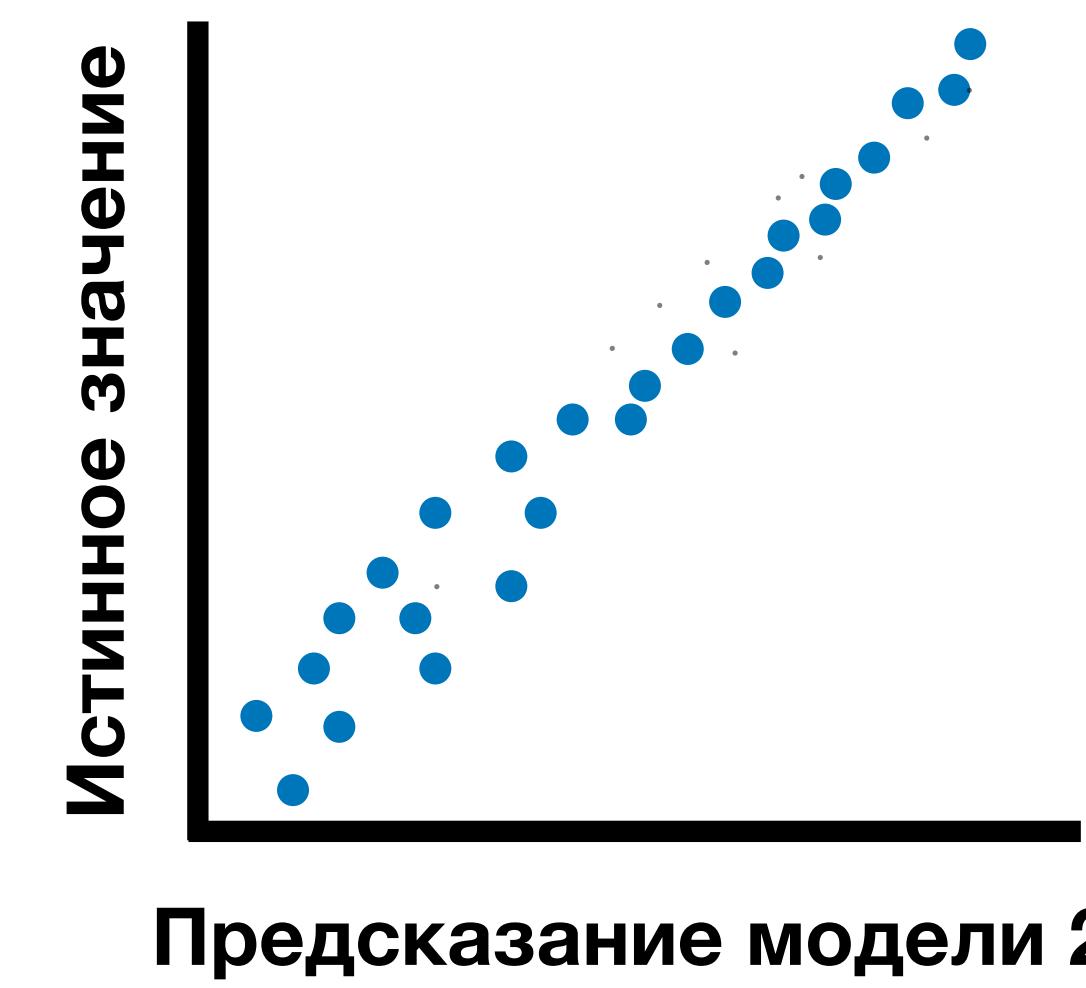
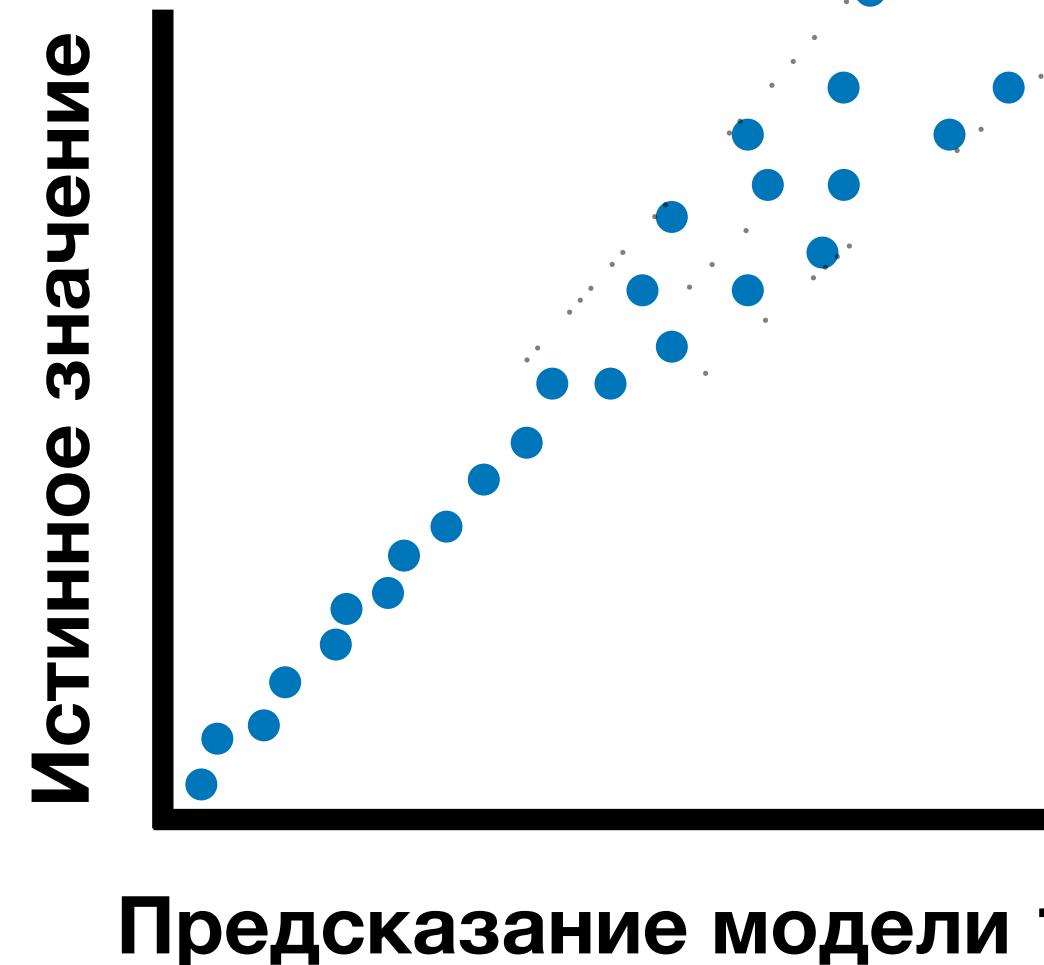
| Интуиция: ансамбли

- В рассмотренных методах построения ансамблей (бэггинг, RSM, случайный лес) решение принимается с помощью усреднения предсказаний
- Кажется, ошибки отдельных моделей должны компенсироваться
- Что будет, если алгоритмы будут работать “хорошо” только в какой-то одной области признакового пространства, а в других - плохо?



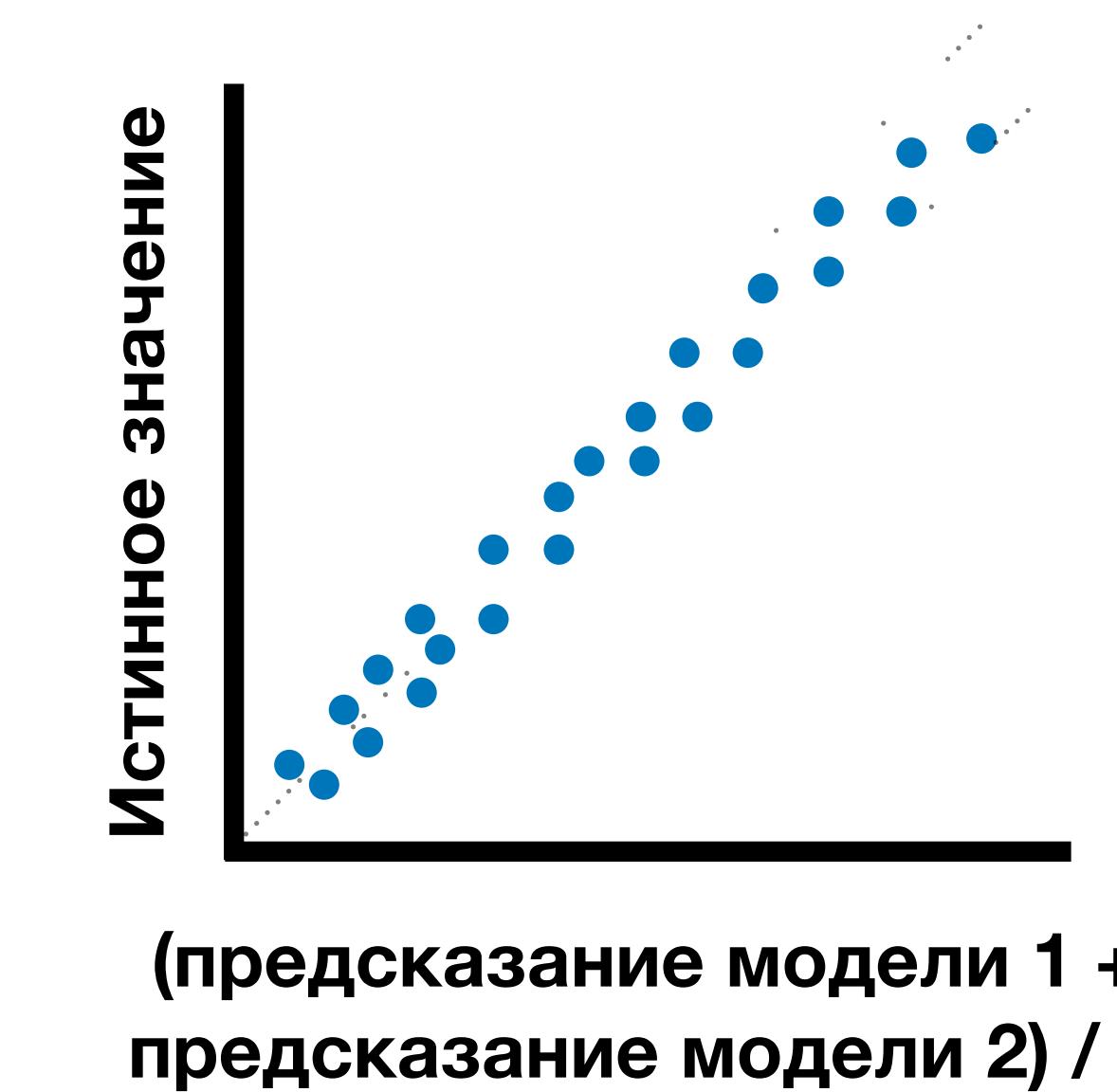
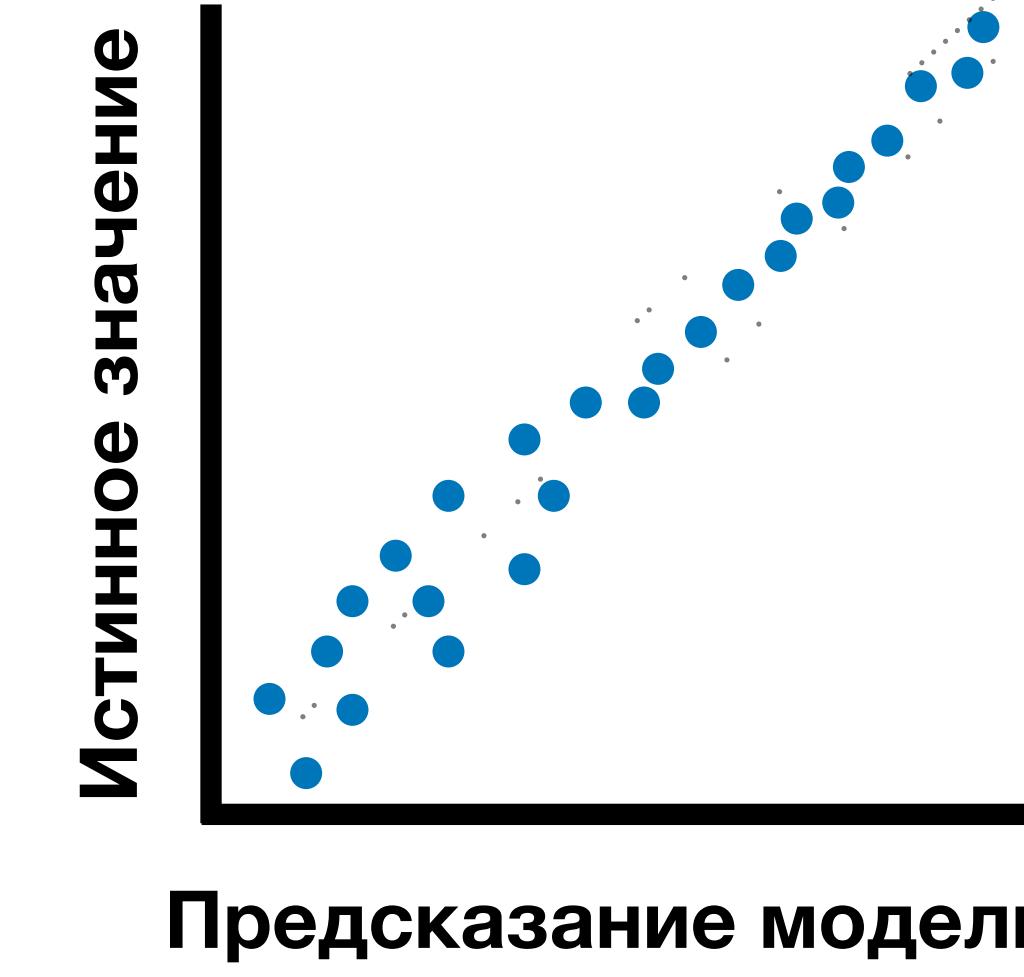
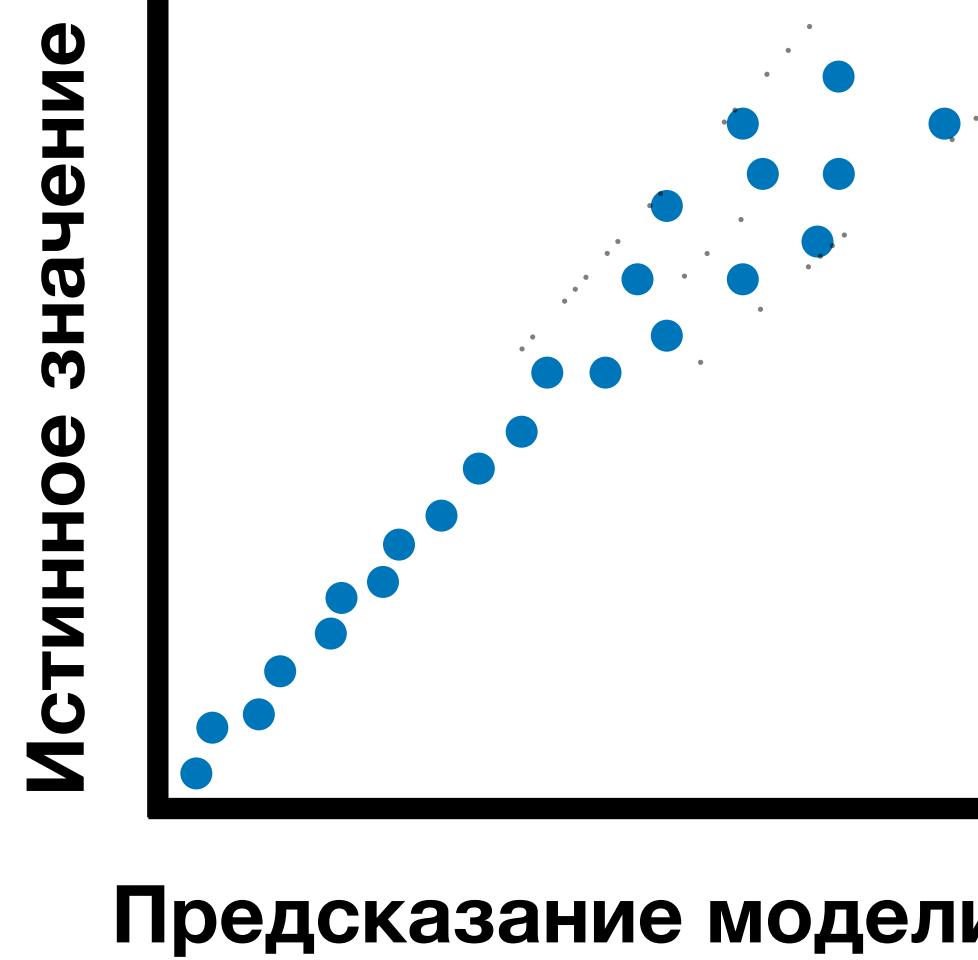
| Интуиция: ансамбли

- В рассмотренных методах построения ансамблей (бэггинг, RSM, случайный лес) решение принимается с помощью усреднения предсказаний
- Кажется, ошибки отдельных моделей должны компенсироваться
- Что будет, если алгоритмы будут работать “хорошо” только в какой-то одной области признакового пространства, а в других - плохо?



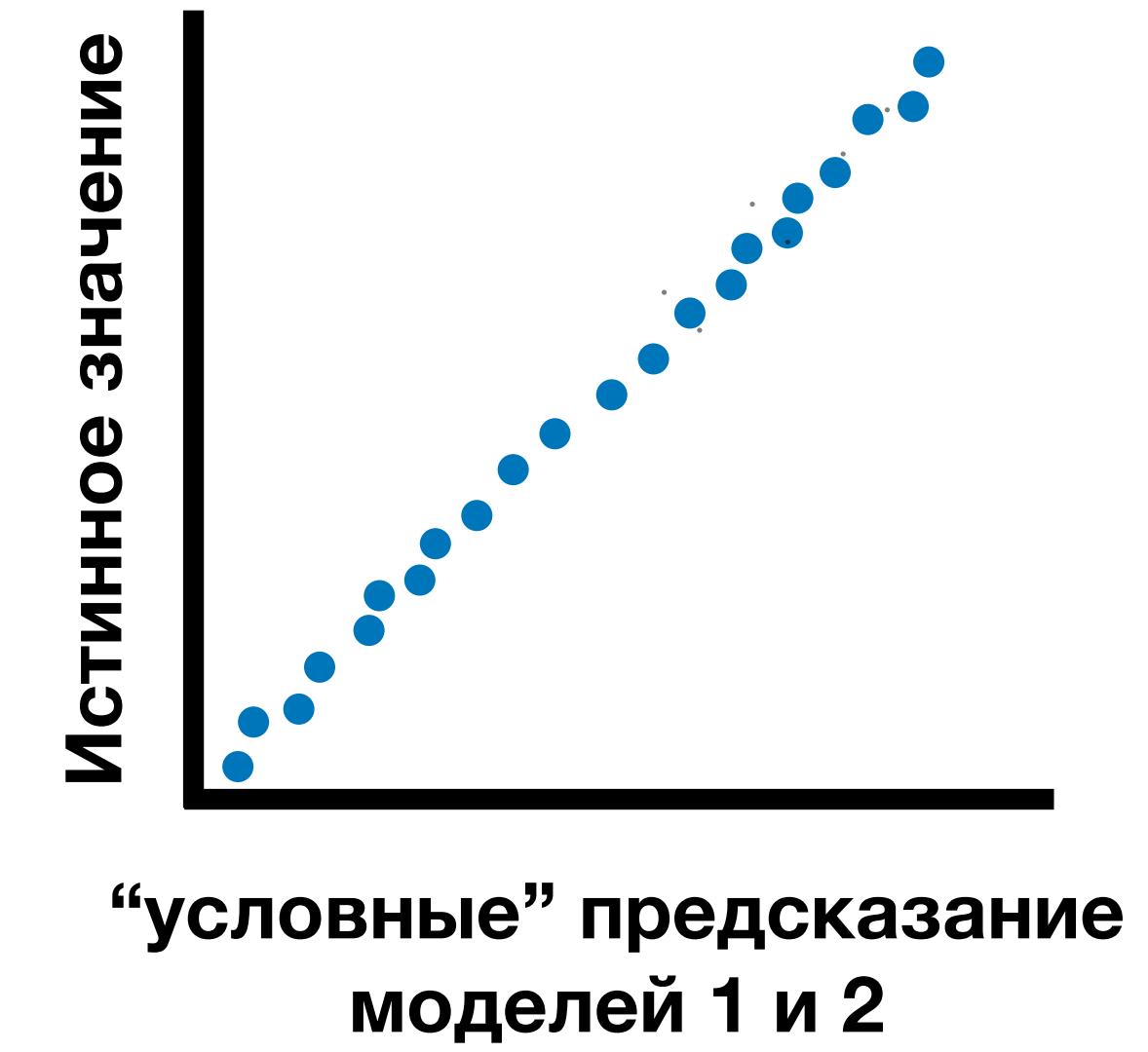
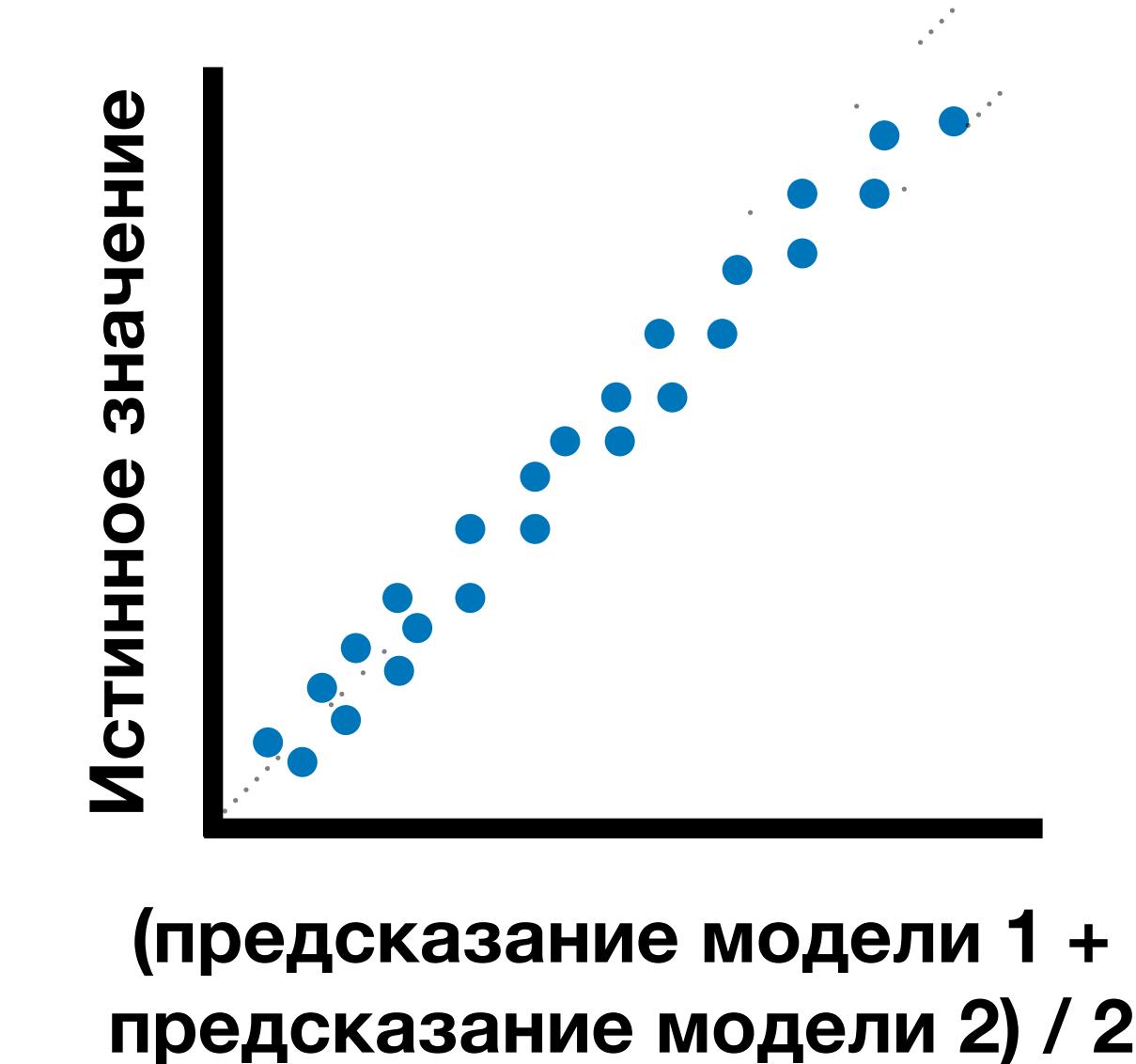
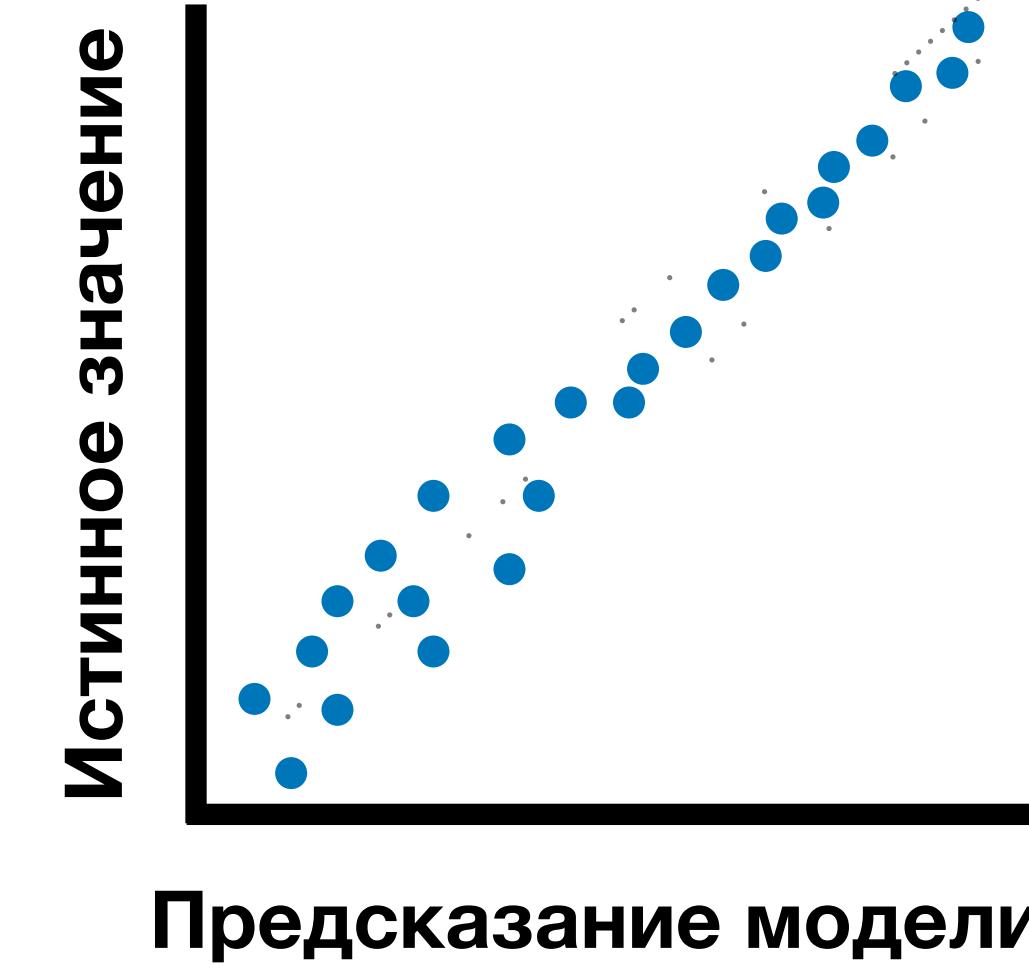
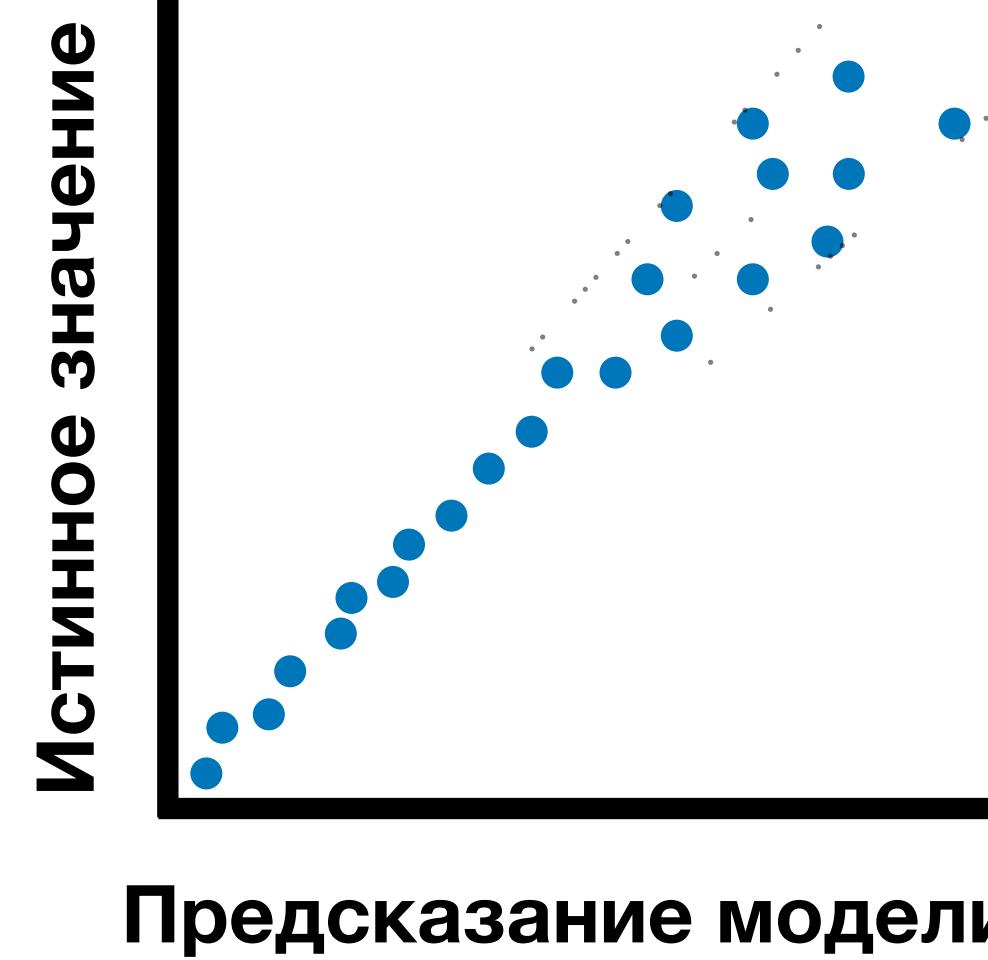
| Интуиция: ансамбли

- Попробуем делать предсказания так - в каждой области используем предсказание той модели, которая работает в ней “лучше”



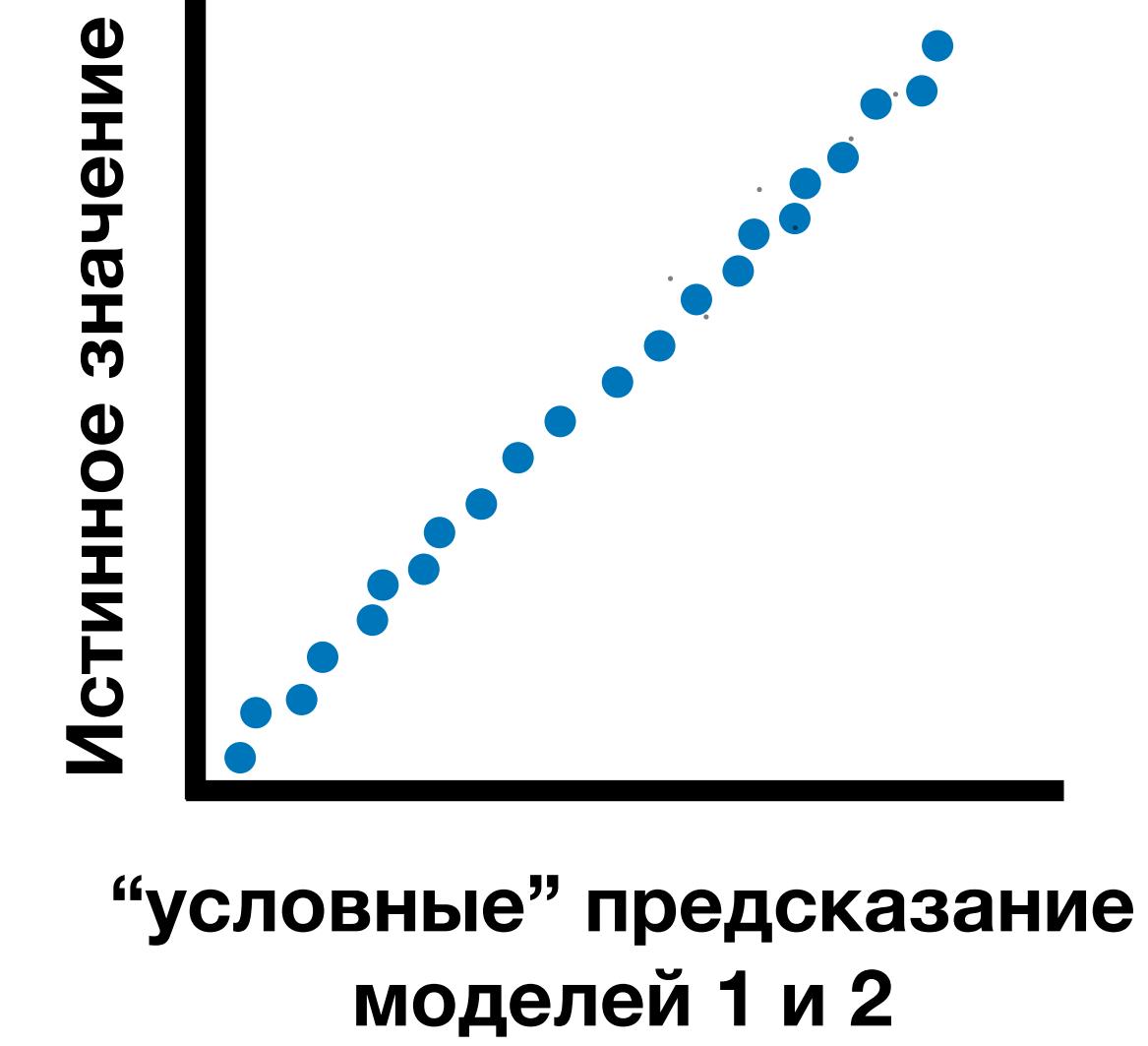
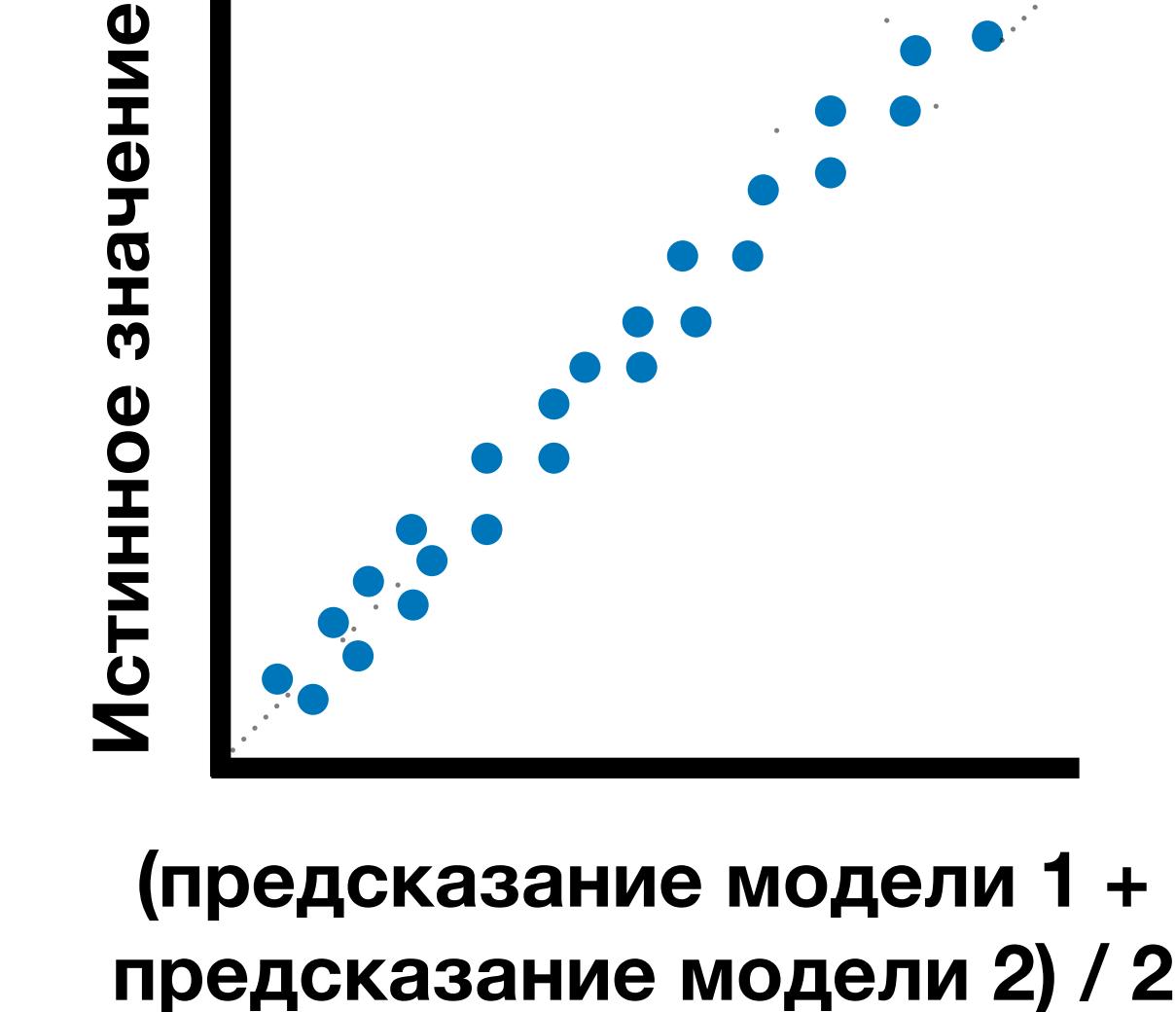
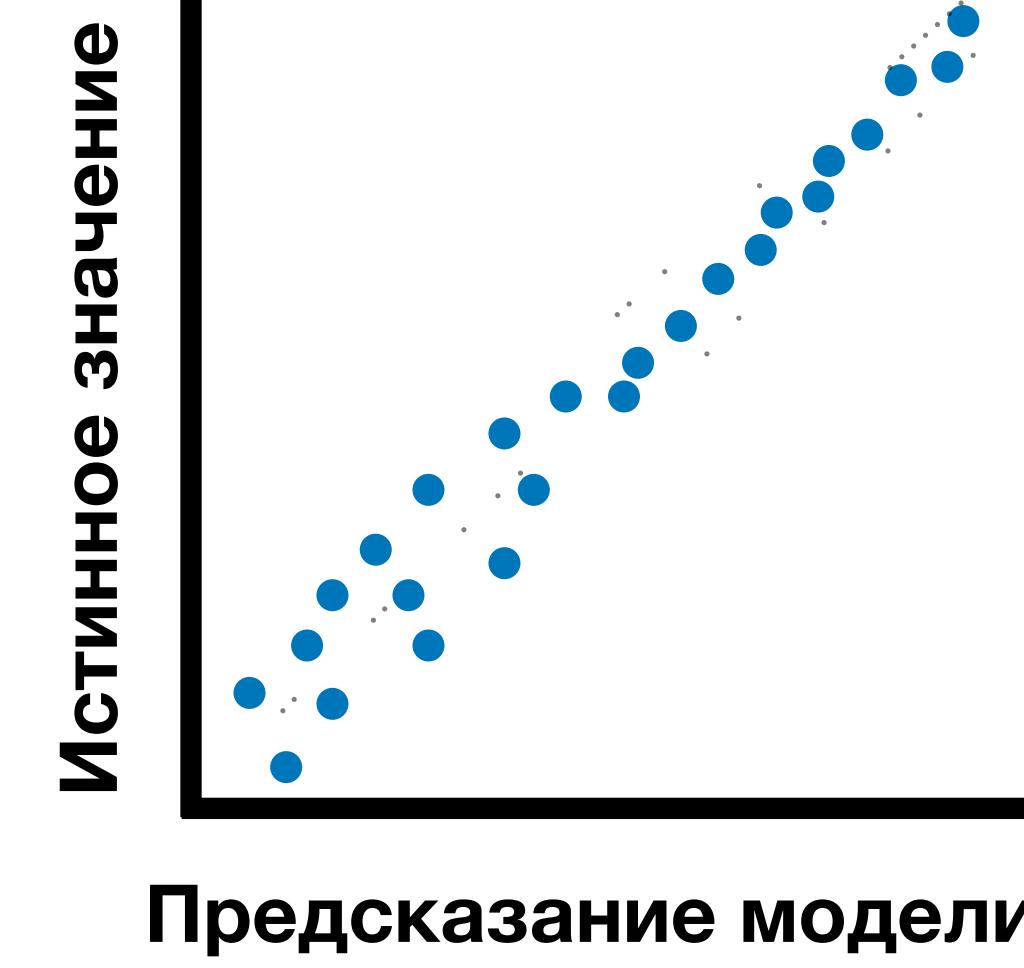
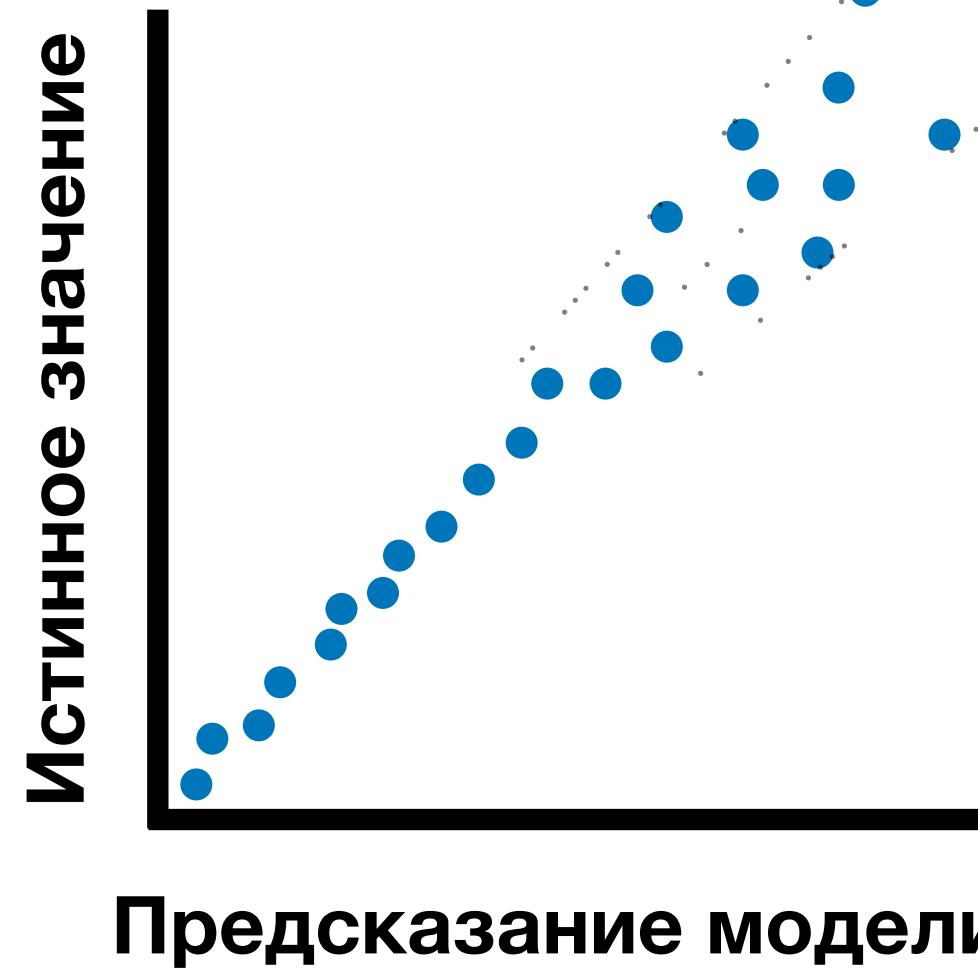
| Интуиция: ансамбли

- Попробуем делать предсказания так - в каждой области используем предсказание той модели, которая работает в ней “лучше”



| Интуиция: ансамбли

- Попробуем делать предсказания так - в каждой области используем предсказание той модели, которая работает в ней “лучше”
- Здесь мы сами решили, как формировать предсказания
- *Можно ли делать это автоматически?*



“Умное” ансамблирование

- Решение о том, насколько “доверять” каждой отдельной модели, можно принимать еще одной моделью
- Такая модель называется **мета-алгоритмом**
- Рассмотрим два наиболее популярных подхода к построению мета-алгоритмов - **блending (blending)** и **стекинг (stacking)**

| Блендинг (Blending)



Блендинг (Blending)

- Даны обучающая (X, y) и тестовая выборки (X_{test}, y_{test})
- Хотим использовать M моделей: $b_1(x), b_2(x), \dots, b_m(x)$
- Поделим (X, y) на 2 части: (X_{train}, y_{train}), (X_{meta}, y_{meta})
- Назовем (X_{train}, y_{train}) = A , (X_{meta}, y_{meta}) = B , (X_{test}, y_{test}) = C
- Для каждой модели b_i :
 - Обучим модель b_i на подвыборке A
 - Для каждого объекта из B сделаем предсказание с помощью b_i , получим i -й столбец матрицы “мета-признаков B ”
 - Для каждого объекта из C сделаем предсказание с помощью b_i , получим i -й столбец матрицы “мета-признаков C ”
 - Получим новую матрицу “мета-признаков” B_{meta} (размера $N_{obj_B} \times M$), составленную из предсказаний моделей b_i для объектов из B , и матрицу “мета-признаков” C_{meta} (размера $N_{obj_C} \times M$), составленную из предсказаний моделей b_i для объектов из C
 - Обучим мета-алгоритм b_{meta} на подборке B_{meta}
 - Для каждого из объектов из C_{meta} сделаем предсказание с помощью b_{meta} - это и будут ответы блендинга

| Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					0
					0
					1
					1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					?
					?
					?
					?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Учим на А

Модель b_1

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					0
					0
					1
					1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					?
					?
					?
					?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **B**

Модель b_1

Получаем 1-й
столбец
B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1					0
0.25					0
0.56					1
0.89					1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					?
					?
					?
					?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **C**

Модел
ь b_1

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1					0
0.25					0
0.56					1
0.89					1

B_{meta}

Получаем 1-й
столбец
C_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33					?
0.28					?
0.57					?
0.99					?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Учим на A

Модел
ъ b₂

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1					0
0.25					0
0.56					1
0.89					1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33					?
0.28					?
0.57					?
0.99					?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **B**

Модель b_2

Получаем 2-й
столбец
B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02				0
0.25	0.34				0
0.56	0.45				1
0.89	0.68				1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33					?
0.28					?
0.57					?
0.99					?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **C**

Модел
ь b_2

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02				0
0.25	0.34				0
0.56	0.45				1
0.89	0.68				1

B_{meta}

Получаем 2-й
столбец
C_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29				?
0.28	0.78				?
0.57	0.4				?
0.99	0.66				?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Учим на А

Модел
ъ въз

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02				0
0.25	0.34				0
0.56	0.45				1
0.89	0.68				1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29				?
0.28	0.78				?
0.57	0.4				?
0.99	0.66				?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **B**

Модел
ь b_3

Получаем 3-й
столбец
B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34			0
0.25	0.34	0.5			0
0.56	0.45	0.49			1
0.89	0.68	0.30			1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29				?
0.28	0.78				?
0.57	0.4				?
0.99	0.66				?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **C**

Модел
ь в3

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34			0
0.25	0.34	0.5			0
0.56	0.45	0.49			1
0.89	0.68	0.30			1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67			?
0.28	0.78	0.56			?
0.57	0.4	0.33			?
0.99	0.66	0.56			?

C_{meta}

Получаем 3-й
столбец
C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Учим на А



X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34	...		0
0.25	0.34	0.5	...		0
0.56	0.45	0.49	...		1
0.89	0.68	0.30	...		1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...		?
0.28	0.78	0.56	...		?
0.57	0.4	0.33	...		?
0.99	0.66	0.56	...		?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **B**

Модель **b_m**

Получаем М-й
столбец
B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34	...	0.34	0
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.25	1
0.89	0.68	0.30	...	0.45	1

B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...		?
0.28	0.78	0.56	...		?
0.57	0.4	0.33	...		?
0.99	0.66	0.56	...		?

C_{meta}

Блендинг (Blending)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1

B

x_1	x_2	x_3	y
2.24	1.24	0.53	0
8.52	7.01	0.9	0
0.53	0.24	1.3	1
0.54	3.33	3.41	1

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Предсказываем
на **C**

Модел
ь b_m

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34	...	0.34	0
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.25	1
0.89	0.68	0.30	...	0.45	1

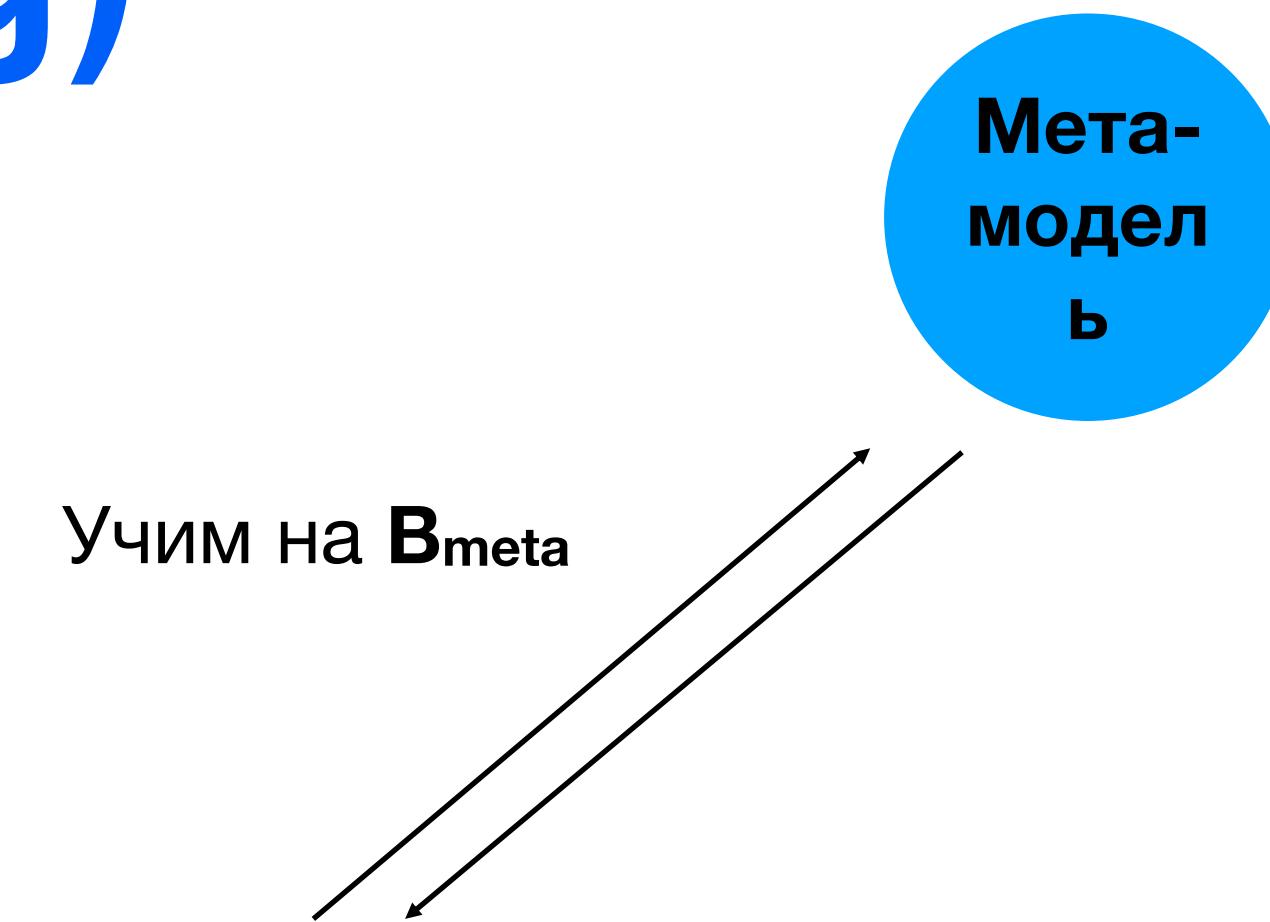
B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

C_{meta}

Получаем M -й
столбец
C_{meta}

| Блендинг (Blending)



X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34	...	0.34	0
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.25	1
0.89	0.68	0.30	...	0.45	1

\mathbf{B}_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

\mathbf{C}_{meta}

| Блендинг (Blending)

Мета-
модел
ь

Предсказываем
итоговые ответы
на \mathbf{C}_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34	...	0.34	0
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.25	1
0.89	0.68	0.30	...	0.45	1

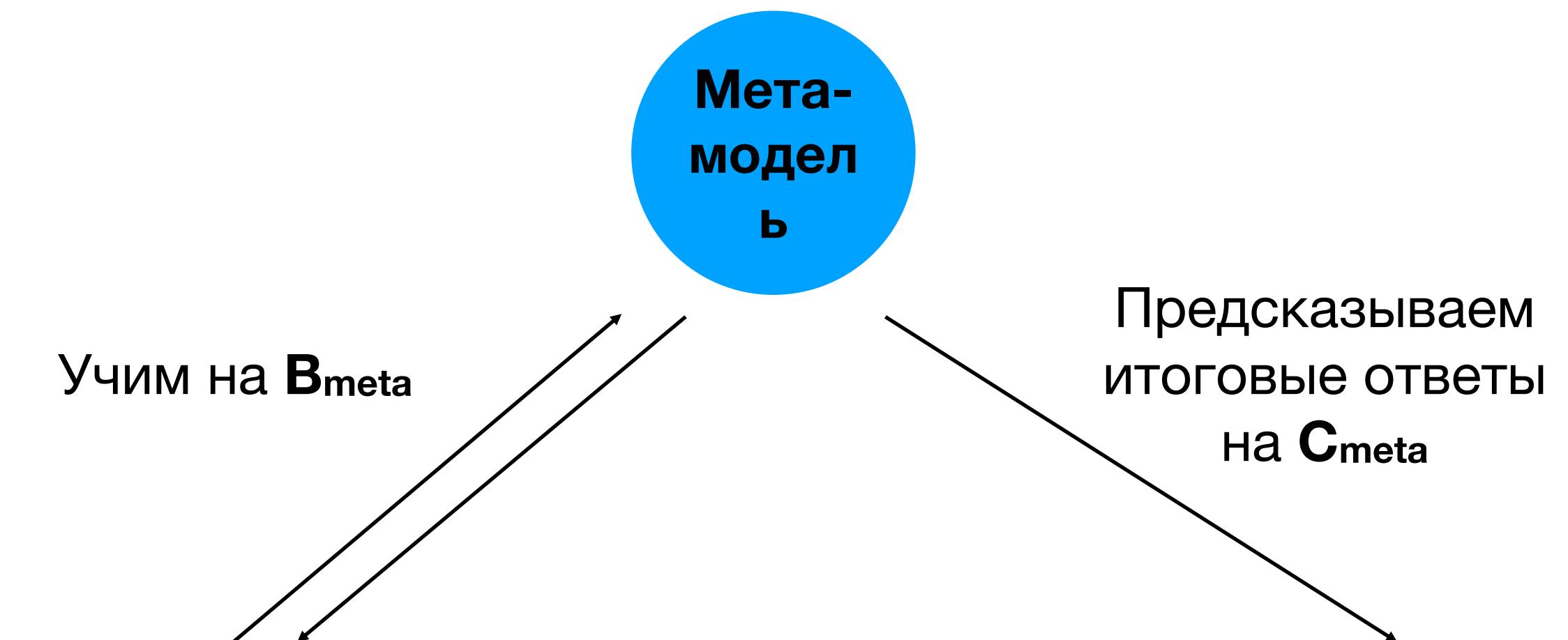
\mathbf{B}_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...	0.28	0
0.28	0.78	0.56	...	0.4	0
0.57	0.4	0.33	...	0.66	1
0.99	0.66	0.56	...	0.67	1

\mathbf{C}_{meta}

| Блендинг (Blending)

- Мета-модель обучается на предсказаниях базовых моделей
- Если вместо мета-модели сделать усреднение предсказаний на тестовой выборке, получится обычное голосование



X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.02	0.34	...	0.34	0
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.25	1
0.89	0.68	0.30	...	0.45	1

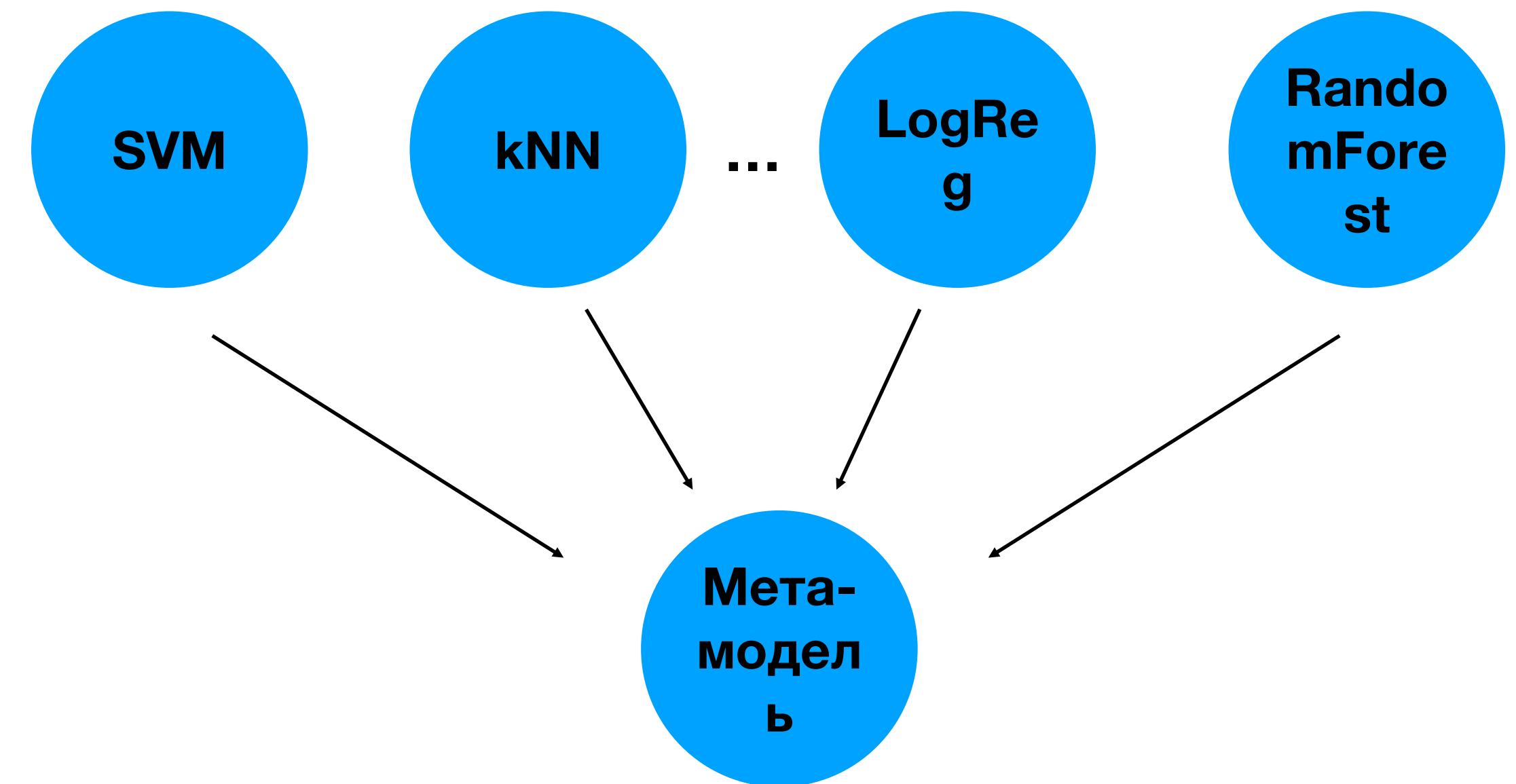
B_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...	0.28	0
0.28	0.78	0.56	...	0.4	0
0.57	0.4	0.33	...	0.66	1
0.99	0.66	0.56	...	0.67	1

C_{meta}

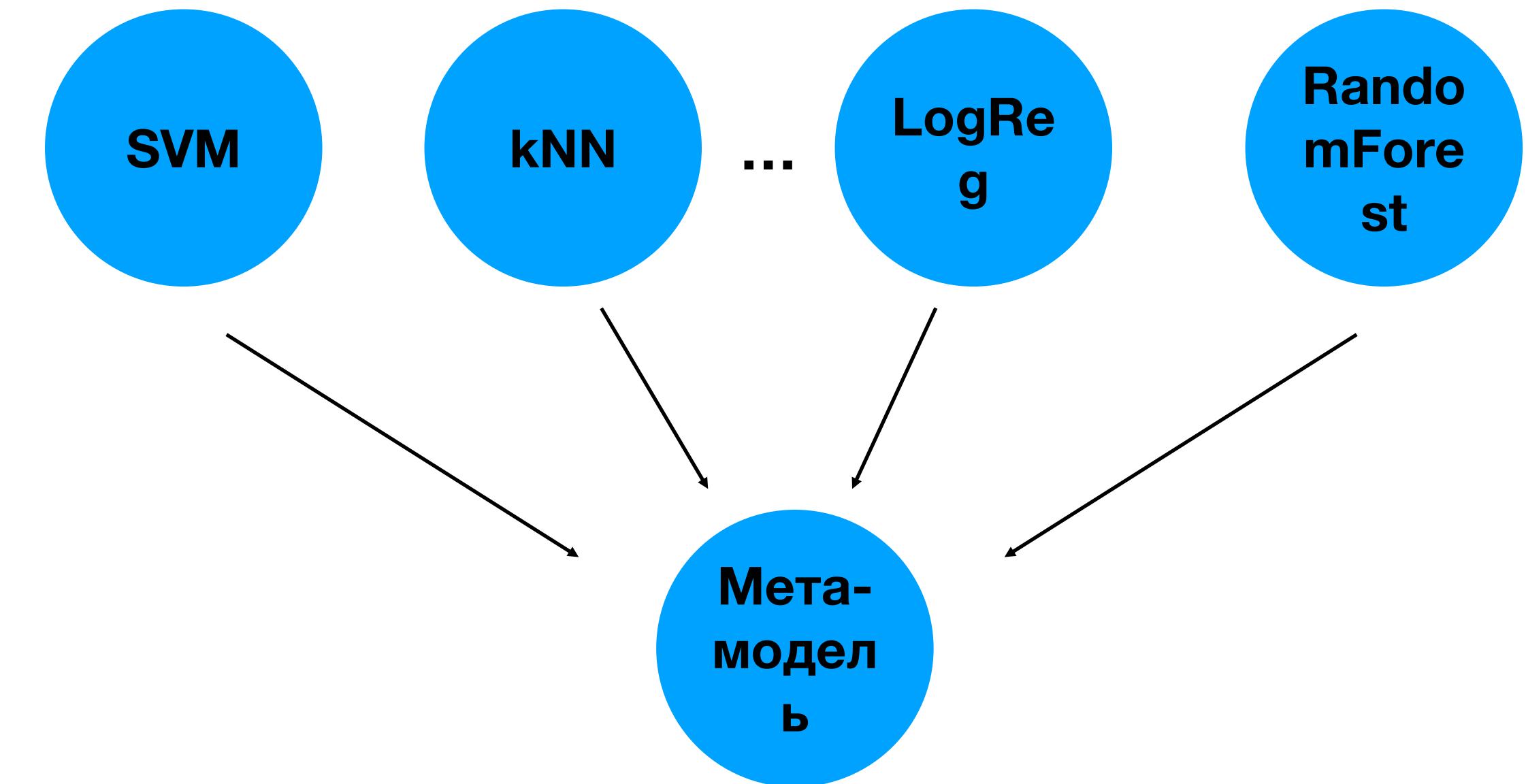
| Блендинг (Blending)

- Можно (и нужно!) использовать разные базовые алгоритмы:
 - Линейные модели (логистическая регрессия, метод опорных векторов, ...)
 - Метод k ближайших соседей
 - Дерево решений
 - Случайный лес
 - ...
- Что следует брать в качестве мета-алгоритма?



| Блендинг (Blending)

- Мета-модель необязательно сложна - иногда достаточно линейной модели (которая буквально “взвесит” предсказания всех базовых моделей)



| Блендинг (Blending)

- *Каков главный и очевидный недостаток блендинга?*

| Блендинг (Blending)

- *Каков главный и очевидный недостаток блендинга?*
- Для обучения каждого из алгоритмов используется не вся выборка:
 - Базовые алгоритмы учатся только на части **A**
 - Мета-алгоритм учится только на части **B**

| Стекинг (Stacking)



Стекинг (Stacking)

- Идея стекинга является обобщением идеи блендинга на случай, когда число разбиений обучающей выборки для построения мета-алгоритма больше 2
- Даны обучающая (\mathbf{X}, \mathbf{y}) и тестовая выборки ($\mathbf{X}_{test}, \mathbf{y}_{test}$)
- Хотим использовать M моделей: $b_1(x), b_2(x), \dots, b_m(x)$
- Поделим (\mathbf{X}, \mathbf{y}) на N равных частей (фолдов, как в кросс-валидации)
- Назовем $(\mathbf{X}_{train,i}, \mathbf{y}_{train,i}) = \mathbf{A}_i (i = 1, 2, 3, \dots, N)$, $(\mathbf{X}_{test}, \mathbf{y}_{test}) = \mathbf{C}$
- Для каждого фолда \mathbf{A}_i :
 - Обучим M моделей на остальных $N-1$ фолдах (то есть на $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{i-1}, \mathbf{A}_{i+1}, \dots, \mathbf{A}_N$)
 - Для каждого объекта из \mathbf{A}_i сделаем предсказание
 - Получим новую матрицу “мета-признаков” для данного фолда (размера $N_{obj_Ai} \times M$)
- Каждую из базовых моделей обучаем **на всей выборке A** и делаем предсказания для \mathbf{C} , получаем матрицу “мета-признаков” \mathbf{C}_{meta} (размера $N_{obj_c} \times M$), составленную из предсказаний моделей b_i для объектов из \mathbf{C}
- Обучим мета-алгоритм b_{meta} на подвыборке \mathbf{A}_{meta}
- Для каждого из объектов из \mathbf{C}_{meta} сделаем предсказание с помощью b_{meta} - это и будут ответы стекинга

Стекинг (Stacking)

A			
x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

A_1	x_1	x_2	x_3	y
	0.15	0.11	3.41	1
	2.42	2.5	0.12	0

A_2	x_1	x_2	x_3	y
	5.14	3.41	7.62	0
	1.22	0.86	1.38	1

A_3	x_1	x_2	x_3	y
	8.52	0.46	0.62	1
	1.1	0.78	0.24	0

- Пример:
- 3 фолда
- M моделей

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

Стекинг (Stacking)

	A			
	x_1	x_2	x_3	y
	0.15	0.11	3.41	1
	2.42	2.5	0.12	0
	5.14	3.41	7.62	0
	1.22	0.86	1.38	1
	8.52	0.46	0.62	1
	1.1	0.78	0.24	0

	x_1	x_2	x_3	y
A_1	0.15	0.11	3.41	1
A_1	2.42	2.5	0.12	0

	x_1	x_2	x_3	y
A_2	5.14	3.41	7.62	0
A_2	1.22	0.86	1.38	1

	x_1	x_2	x_3	y
A_3	8.52	0.46	0.62	1
A_3	1.1	0.78	0.24	0

На этом фолде обучаем M моделей

Для этого фолда делаем M предсказаний

Ameta

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					1
					0
					0
					1
					1
					0

Стекинг (Stacking)

A

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

A₁

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0

A₂

x_1	x_2	x_3	y
5.14	3.41	7.62	0
1.22	0.86	1.38	1

A₃

x_1	x_2	x_3	y
8.52	0.46	0.62	1
1.1	0.78	0.24	0

· Шаг ·

· 1 ·

⋮ ⋮

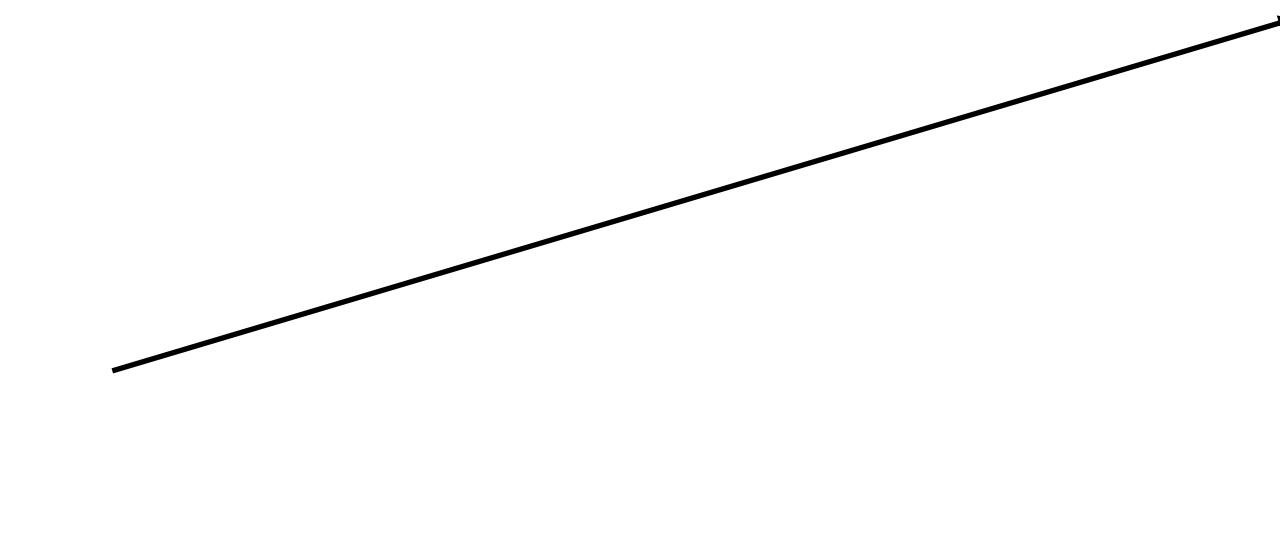
⋮ ⋮

На этом фолде обучаем M
моделей

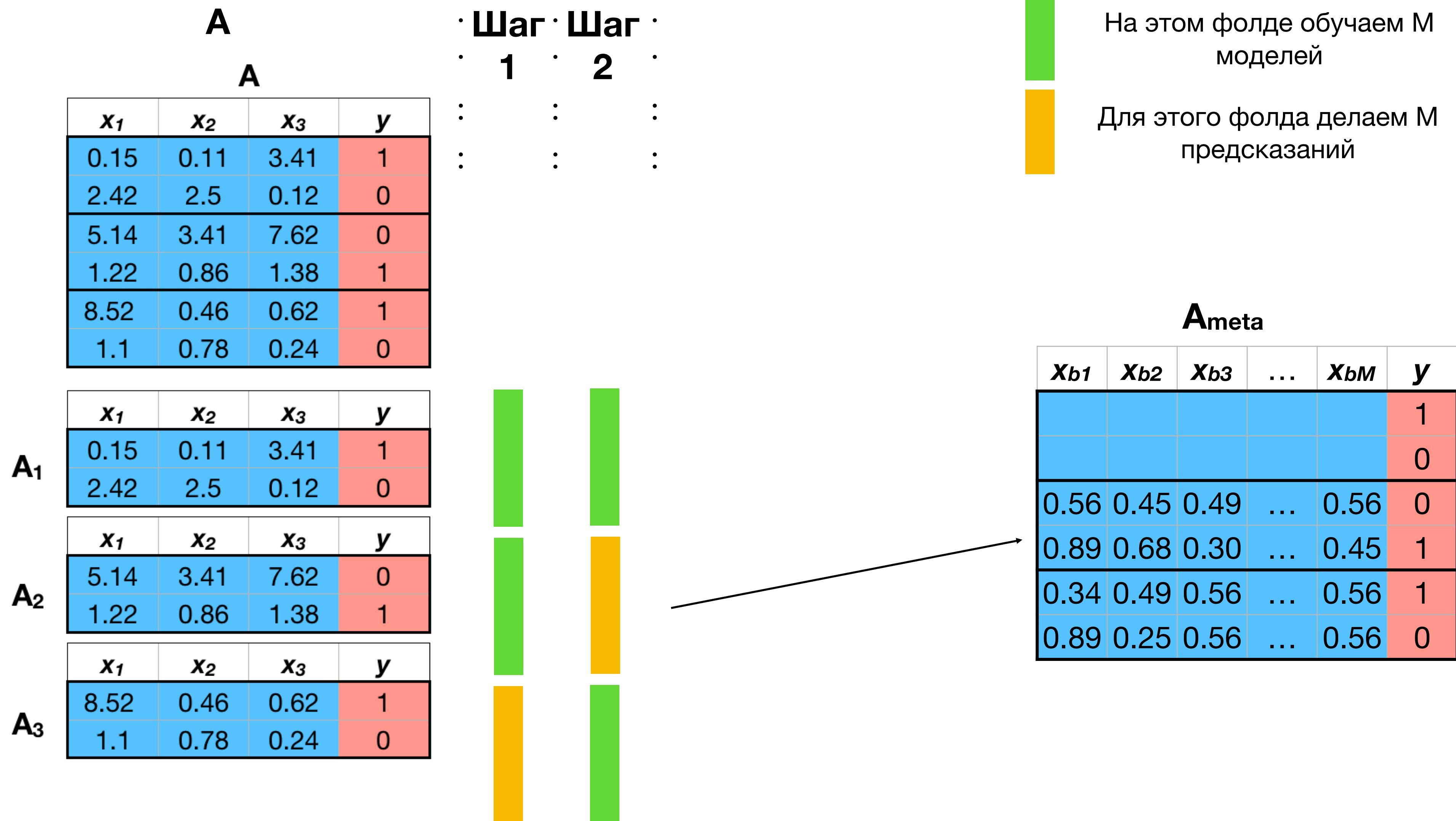
Для этого фолда делаем M
предсказаний

A_{meta}

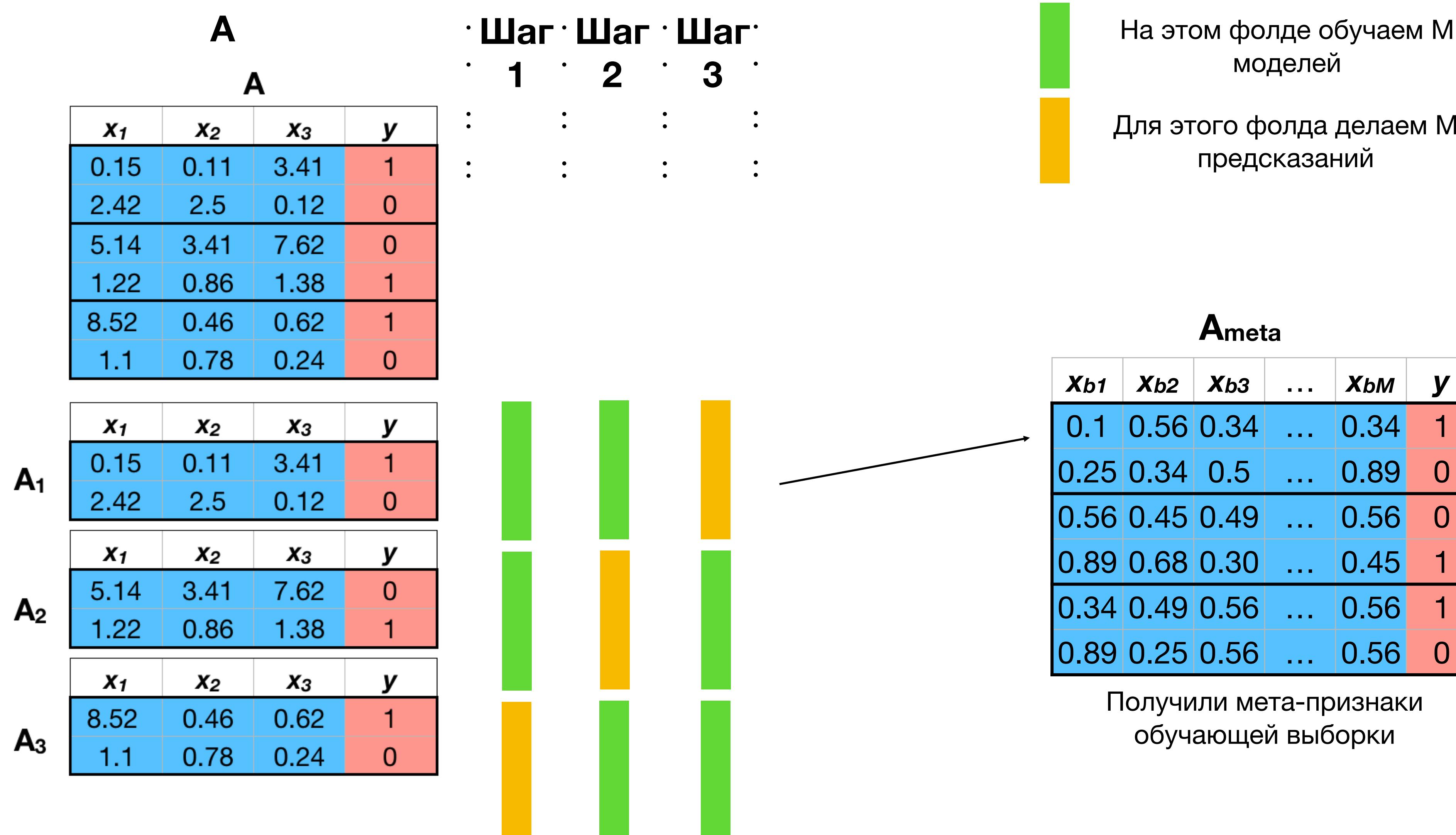
X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
					1
					0
					0
					1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0



Стекинг (Stacking)



Стекинг (Stacking)



Стекинг (Stacking)

C

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

A_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

Мета-признаки обучающей
выборки

C_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
					?
					?
					?
					?

Мета-признаки тестовой
выборки

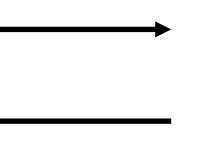
Стекинг (Stacking)

A

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

Учим на A



C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

A_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

Мета-признаки обучающей
выборки

C_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
					?
					?
					?
					?

Мета-признаки тестовой
выборки

Стекинг (Stacking)

A

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0



Предсказываем
на C

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

A_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

Мета-признаки обучающей
выборки

C_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

Мета-признаки тестовой
выборки

Стекинг (Stacking)

A

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

C

x_1	x_2	x_3	y
1.12	5.5	3.33	?
0.53	7.53	3.33	?
3.32	8.52	1.38	?
9.1	3.33	0.53	?

A_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

C_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

Мета-признаки обучающей
выборки

Мета-признаки тестовой
выборки

Стекинг (Stacking)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

A_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

Мета-признаки обучающей
выборки

C_{meta}

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

Мета-признаки тестовой
выборки

Стекинг (Stacking)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

Мета-
модел
ь

A_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

Мета-признаки обучающей
выборки

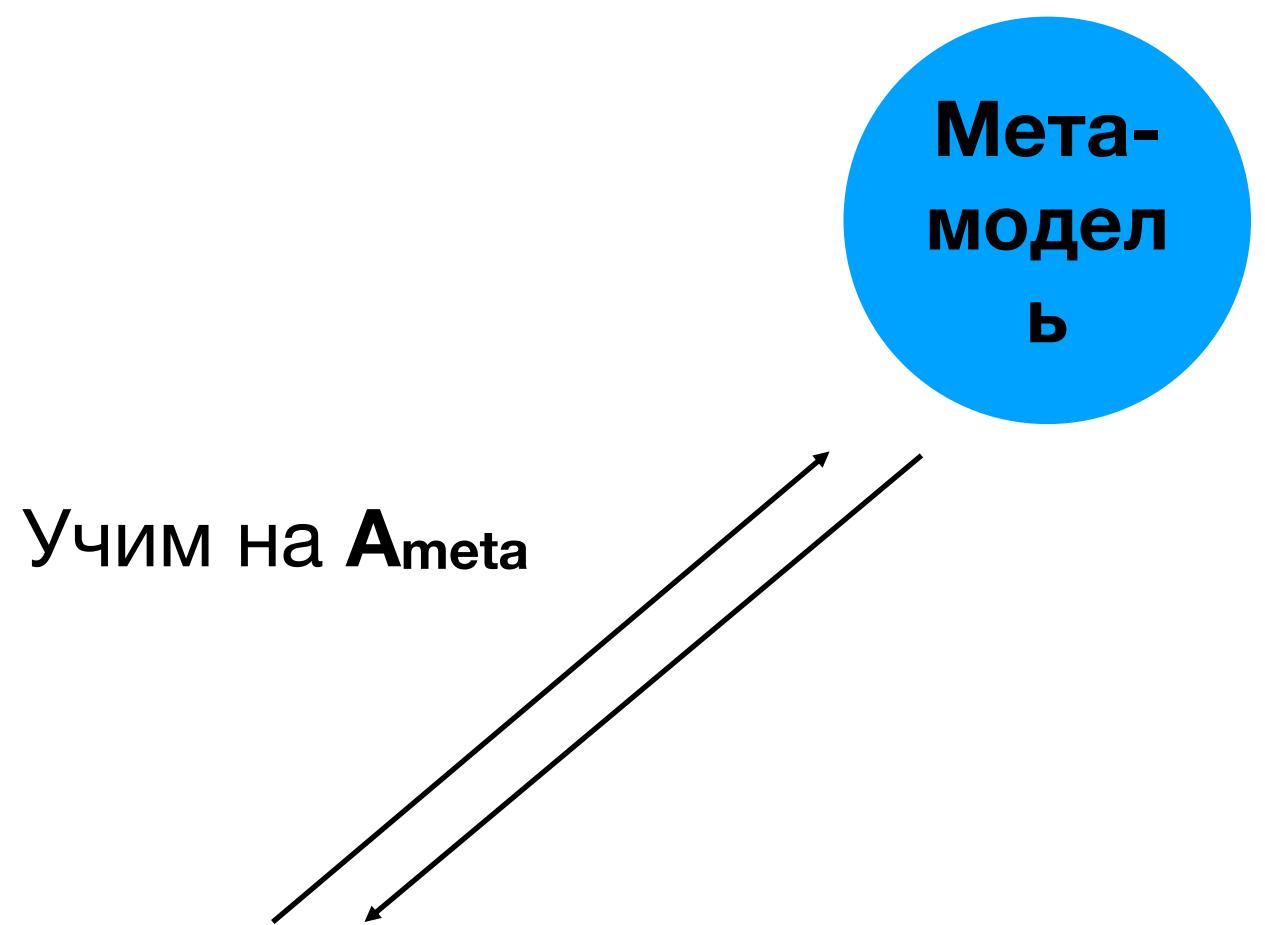
C_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

Мета-признаки тестовой
выборки

Стекинг (Stacking)

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0



x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

Мета-признаки обучающей
выборки

x_{b1}	x_{b2}	x_{b3}	...	x_{bM}	y
0.33	0.29	0.67	...	0.28	?
0.28	0.78	0.56	...	0.4	?
0.57	0.4	0.33	...	0.66	?
0.99	0.66	0.56	...	0.67	?

Мета-признаки тестовой
выборки

Стекинг (Stacking)

A

x_1	x_2	x_3	y
0.15	0.11	3.41	1
2.42	2.5	0.12	0
5.14	3.41	7.62	0
1.22	0.86	1.38	1
8.52	0.46	0.62	1
1.1	0.78	0.24	0

Мета-
модел
ь

Предсказываем
итоговые ответы
на C_{meta}

A_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.1	0.56	0.34	...	0.34	1
0.25	0.34	0.5	...	0.89	0
0.56	0.45	0.49	...	0.56	0
0.89	0.68	0.30	...	0.45	1
0.34	0.49	0.56	...	0.56	1
0.89	0.25	0.56	...	0.56	0

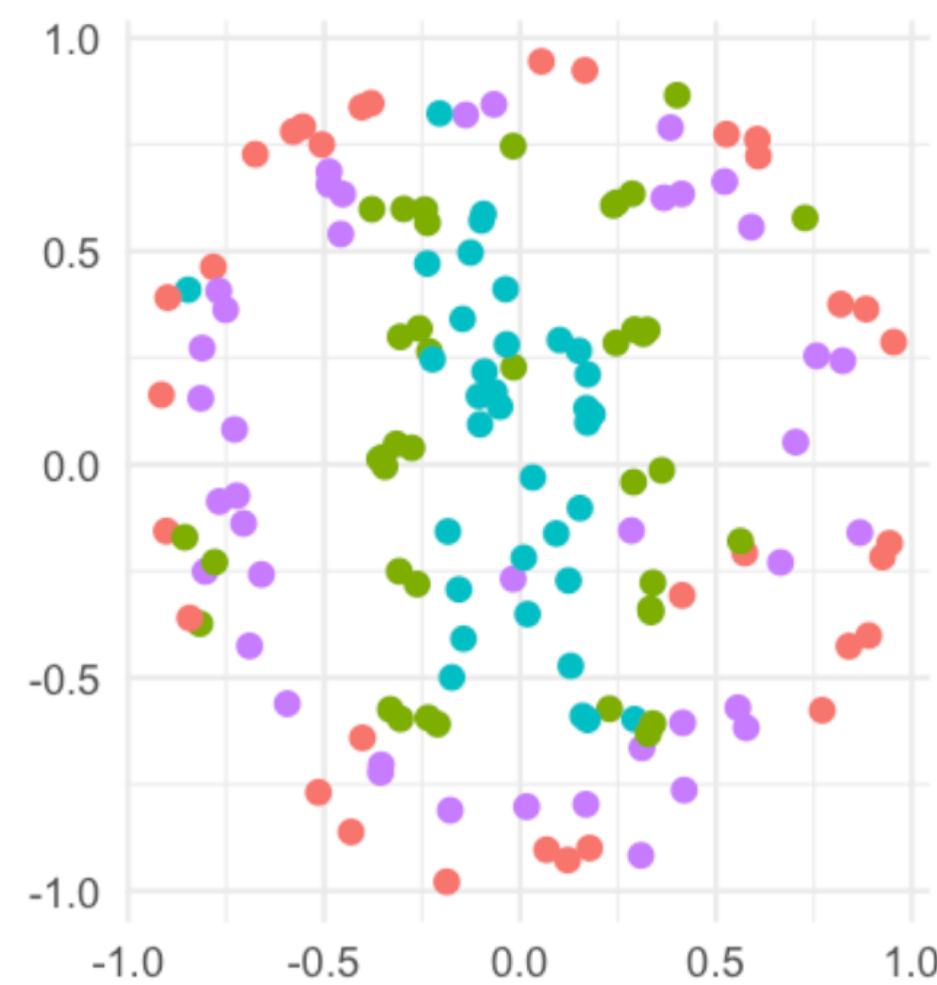
Мета-признаки обучающей
выборки

C_{meta}

X_{b1}	X_{b2}	X_{b3}	...	X_{bM}	y
0.33	0.29	0.67	...	0.28	0
0.28	0.78	0.56	...	0.4	0
0.57	0.4	0.33	...	0.66	1
0.99	0.66	0.56	...	0.67	1

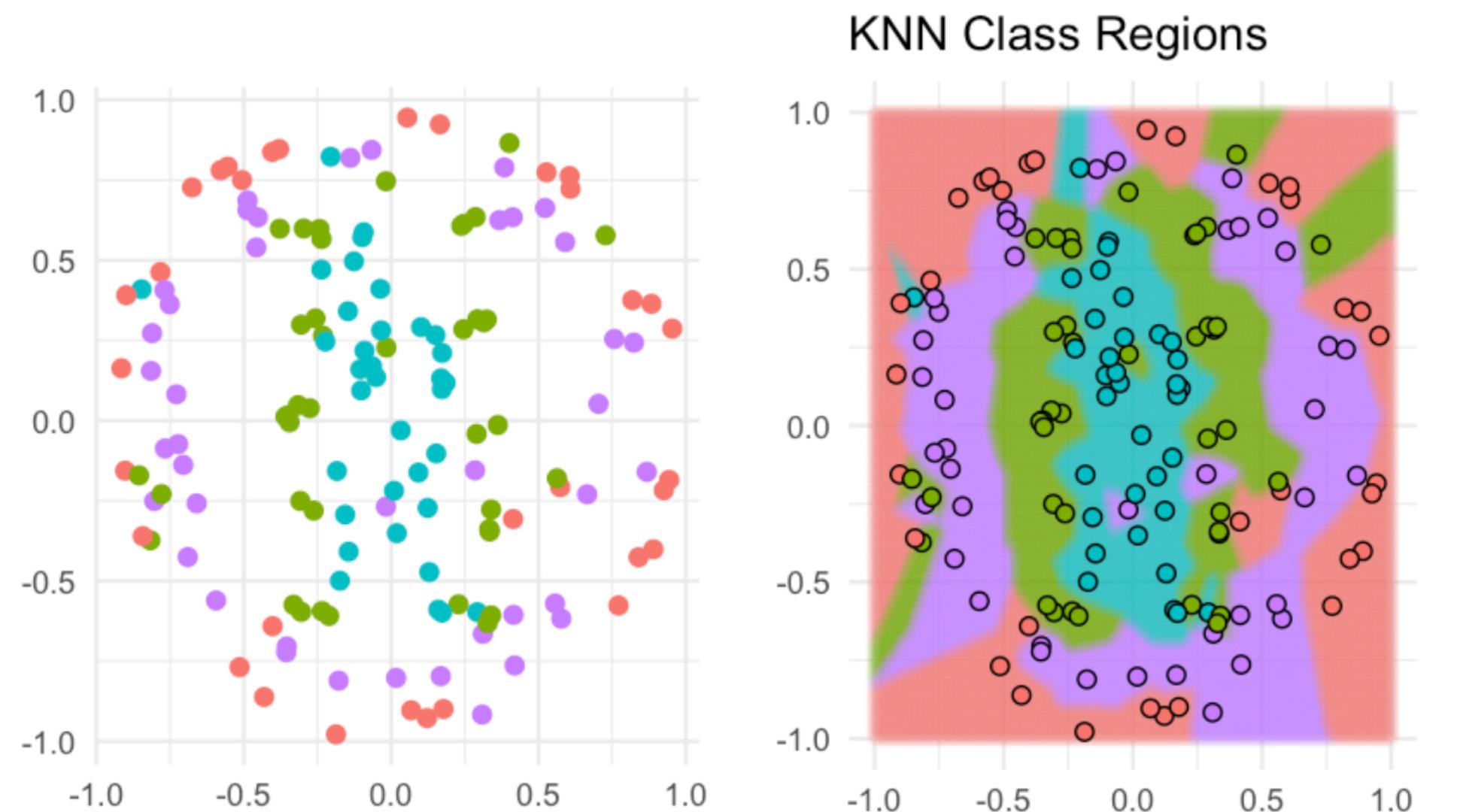
Мета-признаки тестовой
выборки

| Стекинг (Stacking)

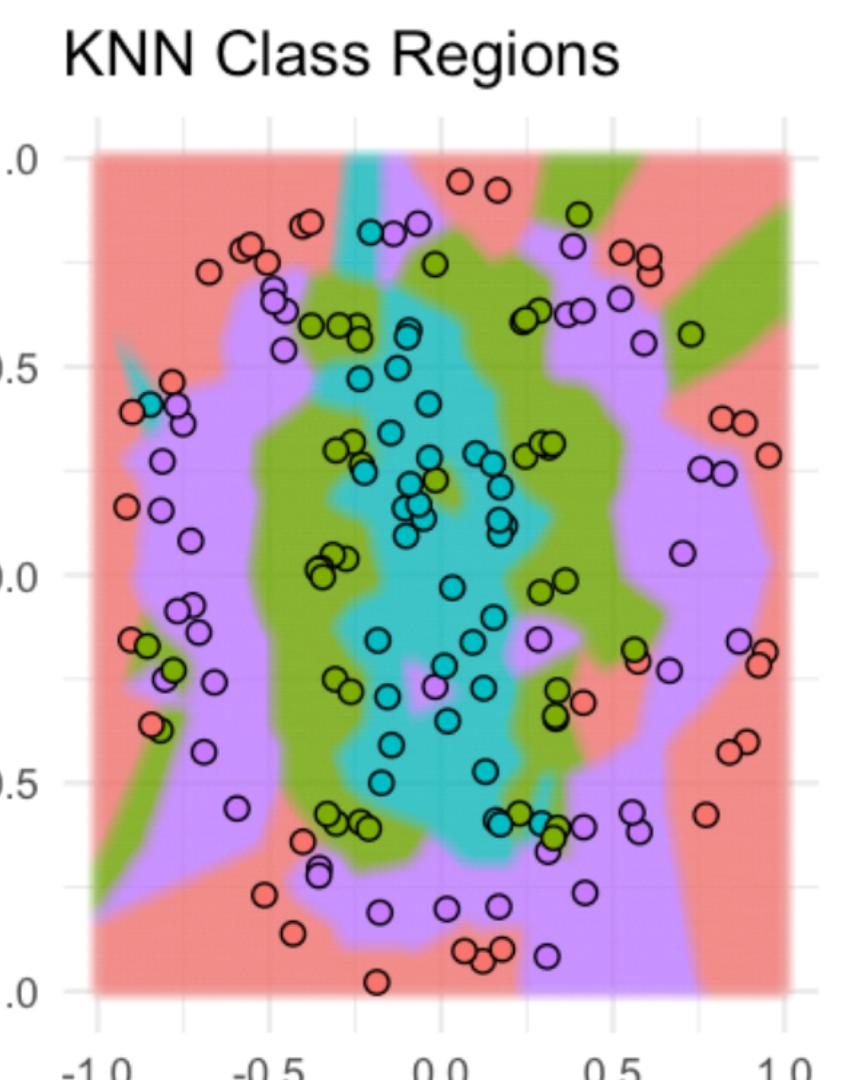


4 класса, есть
“локальная” и
“радиальная”
структура в данных

| Стекинг (Stacking)

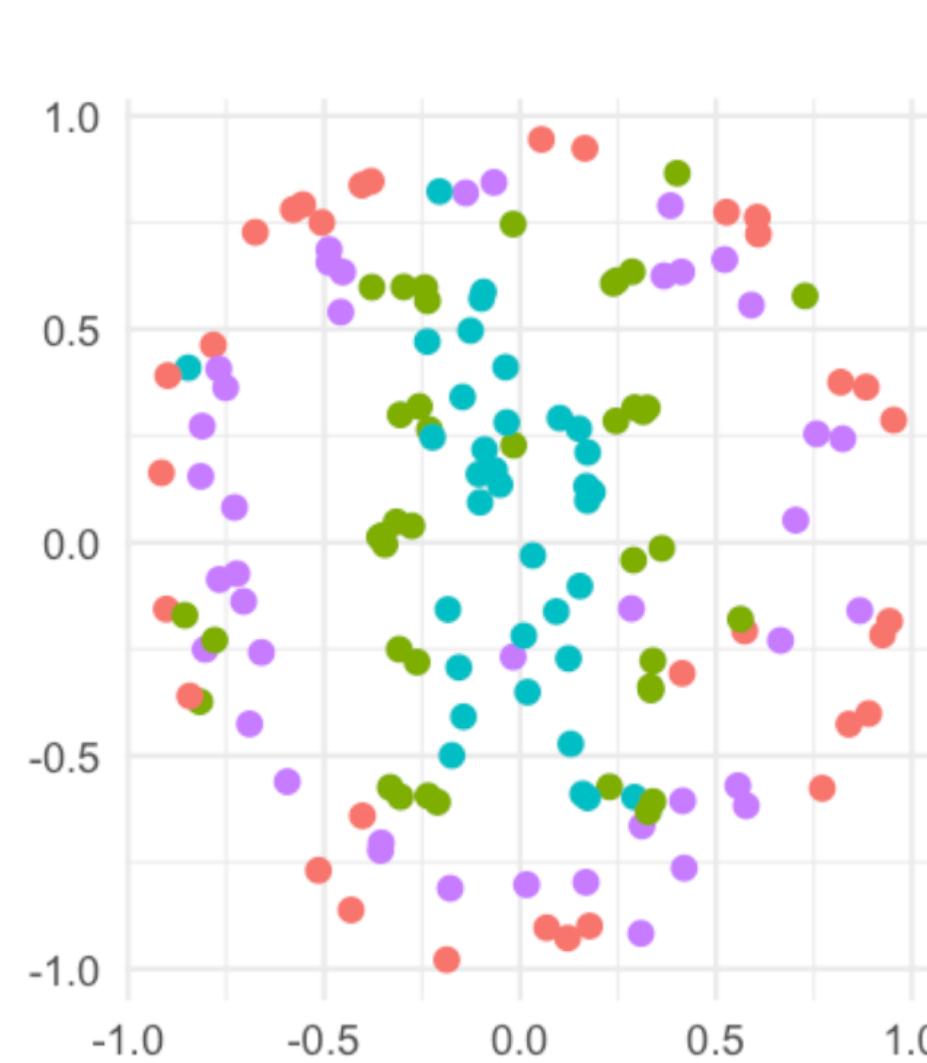


4 класса, есть
“локальная” и
“радиальная”
структура в данных

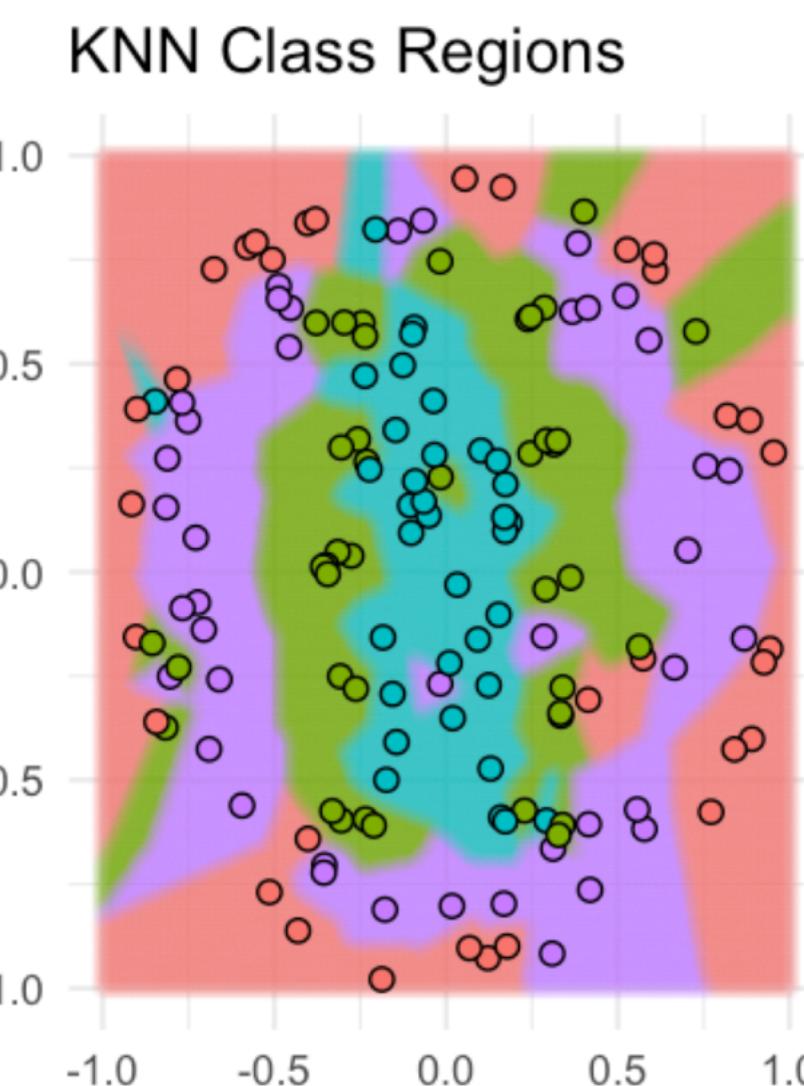


kNN видит
“локальную”
структуре

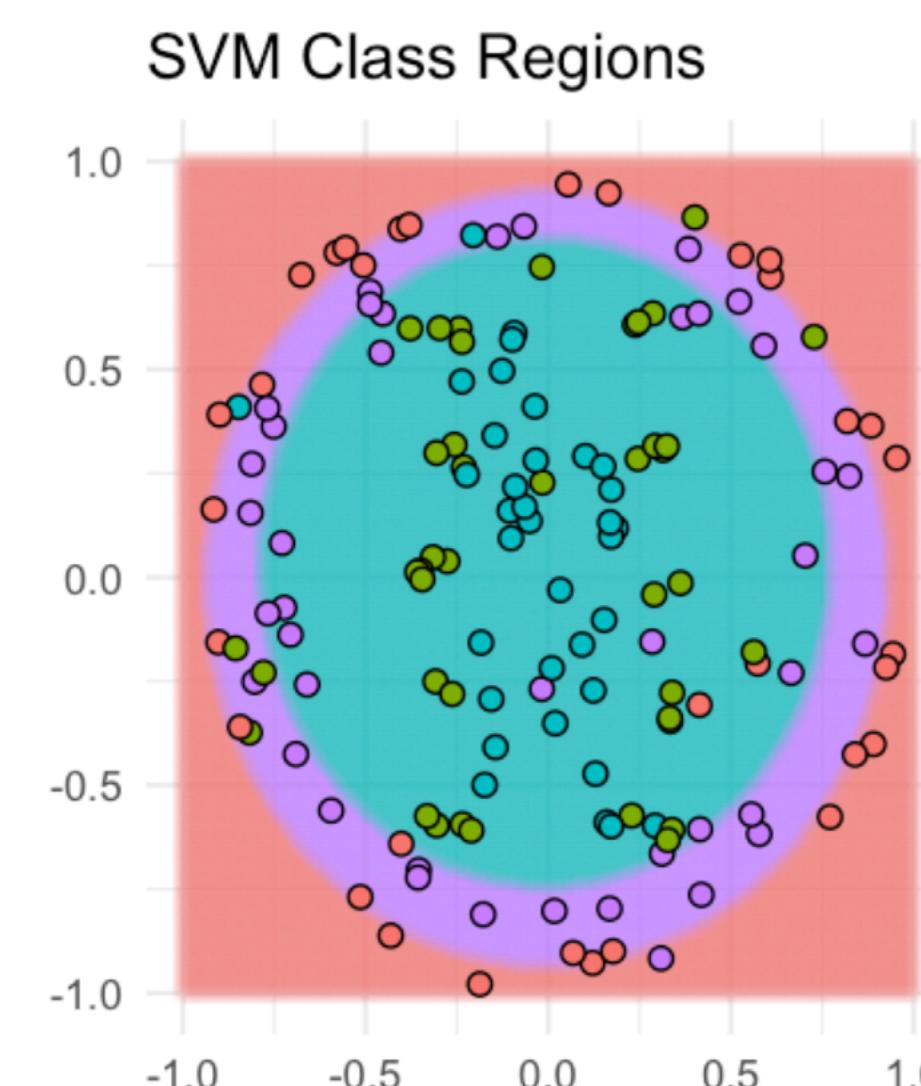
| Стекинг (Stacking)



4 класса, есть
“локальная” и
“радиальная”
структура в данных

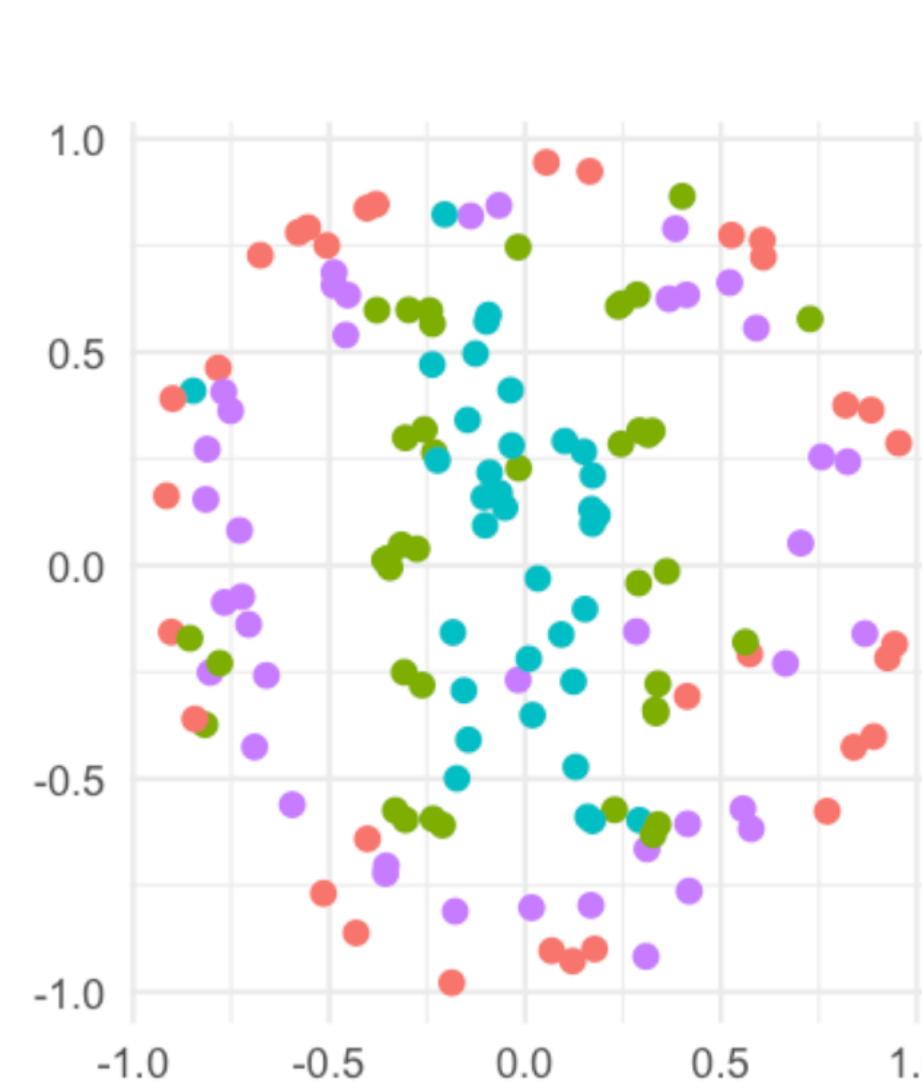


kNN видит
“локальную”
структуре

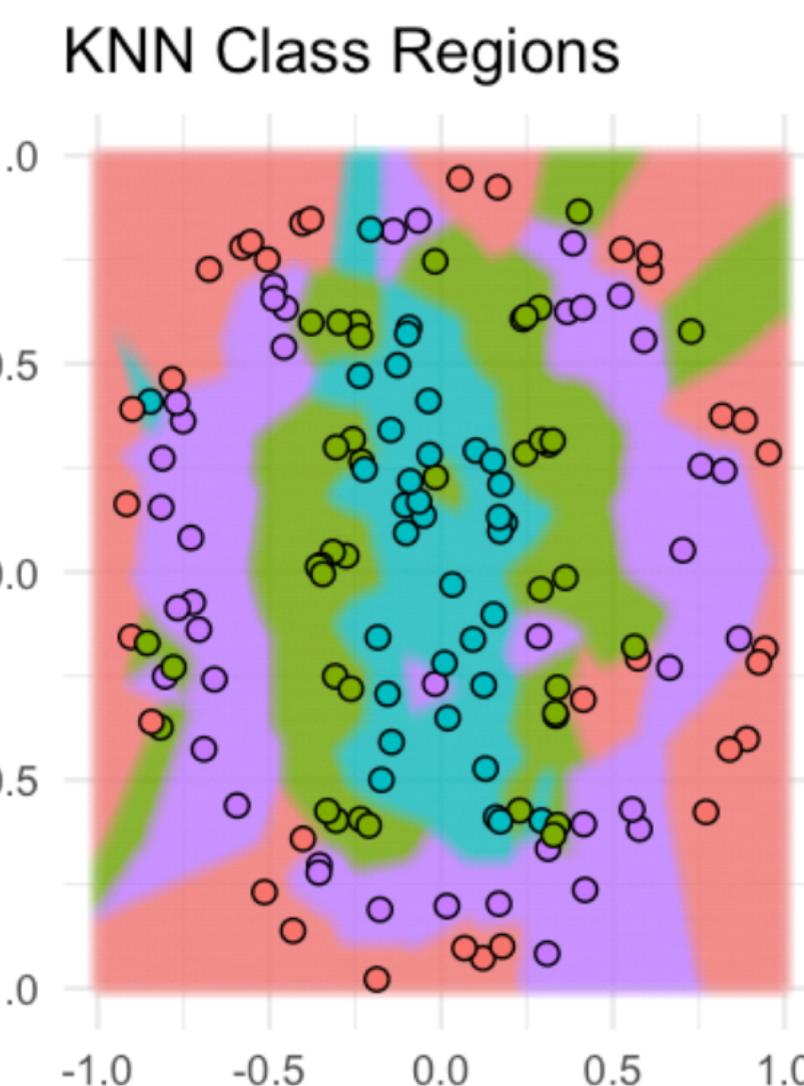


SVM видит
“радиальную”
структуре

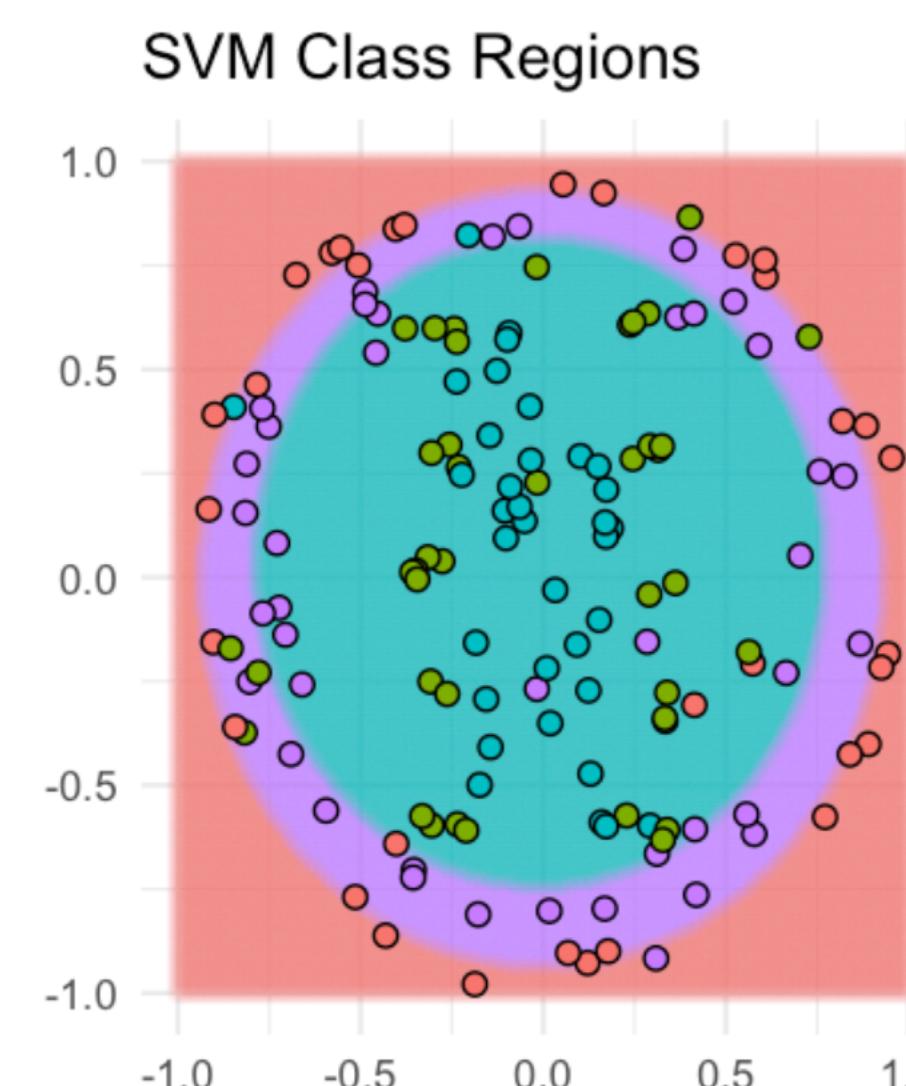
Стекинг (Stacking)



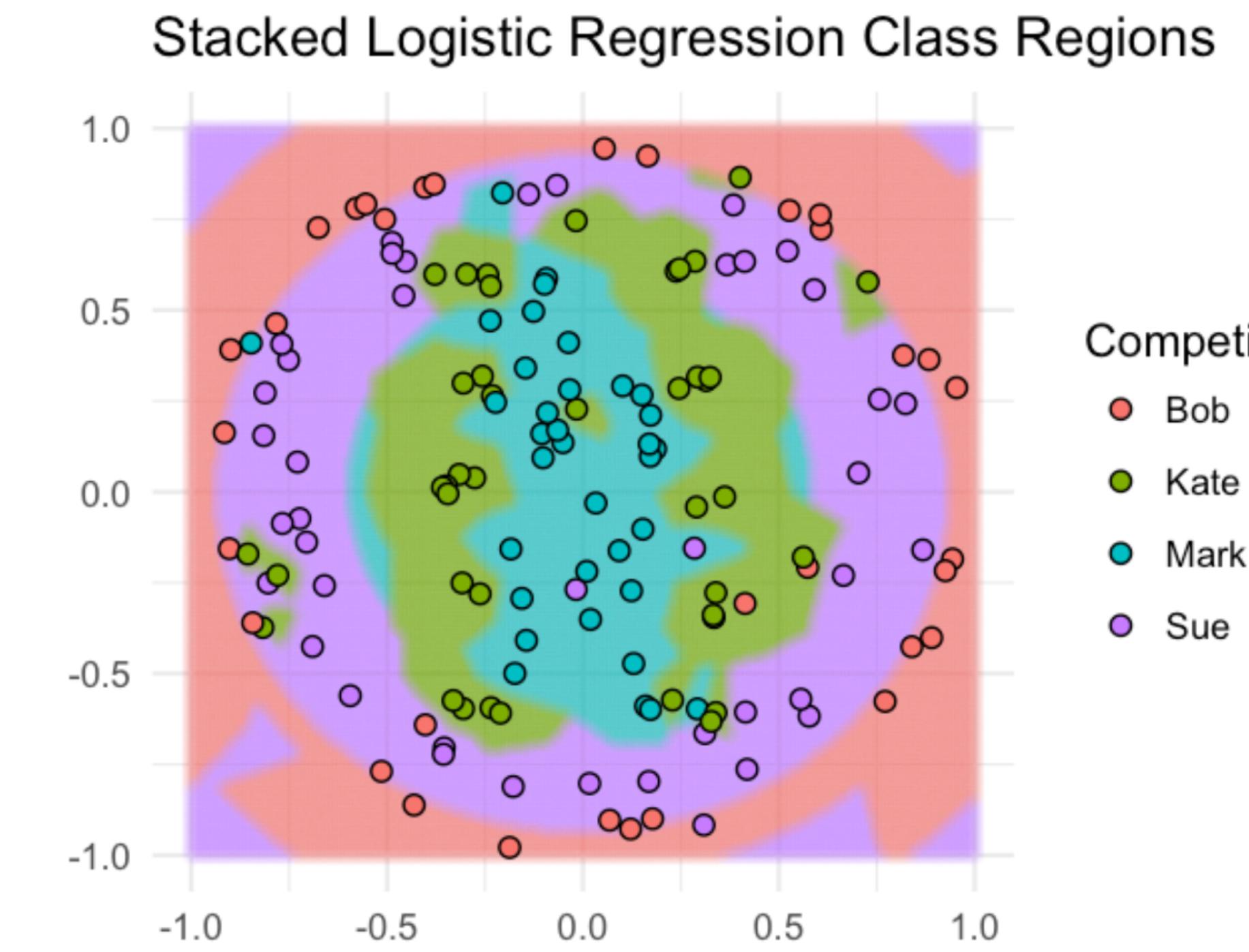
4 класса, есть
“локальная” и
“радиальная”
структура в данных



kNN видит
“локальную”
структуре



SVM видит
“радиальную”
структуре



Логистическая регрессия на ответах
kNN и SVM учитывает и локальную, и
радиальную структуру в данных

| Стекинг (Stacking)

- Можно делать несколько “этажей” стекинга - тогда получаются “мета-алгоритм”, “мета-мета-алгоритм”, ...
- Нужно подбирать мета-алгоритма и число фолдов
- Не гарантирует 100% повышения качества относительно лучшей модели
- Иногда вместе с мета-признаками используются исходные признаки
- Для обучения базовых моделей может потребоваться отдельная предобработка признаков (нормализация для линейных моделей, ...)

| Стекинг (Stacking)

- *Каков главный и очевидный недостаток стекинга?*

| Стекинг (Stacking)

- *Каков главный и очевидный недостаток стекинга?*
- Мета-признаки обучающей и тестовой выборок - разные!
- Мета-признаки обучающей выборки получены с помощью моделей, обученных на $(N-1)/N$ -части всей выборки
- Мета-признаки тестовой выборки получены с помощью моделей, обученных на всей обучающей выборке

| Стекинг и Kaggle

- В спортивном анализе данных стекинг - один из основных инструментов
- При объединении участников в команды делают стекинг моделей отдельных участников
- Количество моделей ограничено только временем его работы и терпеливостью пользователя

Google search results for "kaggle 1st place stacking". The search bar shows the query. Below it, a navigation bar includes "Все" (All), "Картинки" (Images), "Видео" (Videos), "Новости" (News), "Покупки" (Purchases), "Ещё" (More), "Настройки" (Settings), and "Инструменты" (Tools). A link to "1st PLACE - WINNER SOLUTION" is shown, followed by a snippet from Gilberto Titericz & Stanislav Semenov's solution. Other results include links to "1st place solution overview" on Kaggle, "1st Place Solution overview" on Kaggle, and an introduction to ensembling/stacking in Python. A Medium post about a first-place challenge in Kaggle is also listed.

kaggle 1st place stacking

Все Картинки Видео Новости Покупки Ещё Настройки Инструменты

Результатов: примерно 43 200 (0,58 сек.)

1st PLACE - WINNER SOLUTION - Gilberto Titericz & Stanislav ...
<https://www.kaggle.com/c/otto-group-product-classification-challenge/..../14335> ▾
1st PLACE SOLUTION - Gilberto Titericz & Stanislav Semenov. First, thanks to ... Also stacked 2 kmeans features using the T-sne 3 dimensions. Dataset: ...

1st place solution overview | Kaggle
<https://www.kaggle.com/c/jigsaw-toxic-comment.../52557> ▾ Перевести эту страницу
For stacking, we used a weighted mean of arithmetic averaging and stacking, which worked marginally better (~.0001) than either approach alone. For stacking ...

1st Place Solution overview | Kaggle
<https://www.kaggle.com/c/statoil-iceberg-classifier.../48241> - Перевести эту страницу
1st Place Solution overview ... We then ensembled and stacked the results and ran the predictions through a Well done taking the first place on both LBs!

Introduction to Ensembling/Stacking in Python | Kaggle
<https://www.kaggle.com/.../introduction-to-ensembling-stack...> ▾ Перевести эту страницу
2018 Kaggle Inc. Our Team Terms Privacy Contact/Support.
Вы посещали эту страницу 23.11.18.

My 1st challenge in Kaggle. How to be ranked in top 1%. - Medium
<https://medium.com/.../my-1st-challenge-in-kaggle-how-to-b...> ▾ Перевести эту страницу
29 янв. 2018 г. - Recently, I have challenged a competition in Kaggle. ... Fig1. Final stacking architecture ... The visualization of 1st place solution is really clear.

| Итоги: блендинг и стекинг

- **Преимущества**
 - Позволяют очень “дешево” повысить качество
 - Хорошо аппроксимируют данные благодаря взвешиванию базовых алгоритмов с разными сильными сторонами
 - Можно распараллелить (на фолды или модели)
- **Недостатки**
 - Требуют большого количества данных
 - Долго учатся и долго работают (в зависимости от времени работы базовых моделей)

| В следующей серии...

- Научимся оставшимся 50% умения “стакать иксджиуст”