СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Александр Калиниченко

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

Модуль 2. Методы искусственного интеллекта

Лекция 7. Случайные леса

АНСАМБЛИ

Хорошим примером ансамблей считается теорема Кондорсе «о **жюри присяжных**» (1784)

Если каждый член жюри присяжных имеет независимое мнение, и если вероятность правильного решения члена жюри больше 0.5, то тогда вероятность правильного решения присяжных в целом возрастает с увеличением количества членов жюри и стремится к единице. Если же вероятность быть правым у каждого из членов жюри меньше 0.5, то вероятность принятия правильного решения присяжными в целом монотонно уменьшается и стремится к нулю с увеличением количества присяжных

N – количество присяжных

p – вероятность правильного решения присяжного

μ – вероятность правильного решения всего жюри

m – минимальное большинство членов жюри, m=floor(N/2)+1

C_N^i – число сочетаний из N по

$$\mu = \sum_{i=m}^{N} C_N^i p^i (1-p)^{N-i}$$

Если p>0.5, то μ>p

Если N→∞, то μ→1

"МУДРОСТЬ ТОЛПЫ"

В 1906 году на одном из рынков проводилась лотерея для крестьян

Их собралось около 800 человек, и они пытались угадать вес быка, который стоял перед ними. Бык весил **1198** фунтов. Ни один крестьянин не угадал точный вес быка, но если посчитать среднее от их предсказаний, то получим **1197** фунтов (ошибка 1 фунт!)

Эту идею уменьшения ошибки применили и в машинном обучении

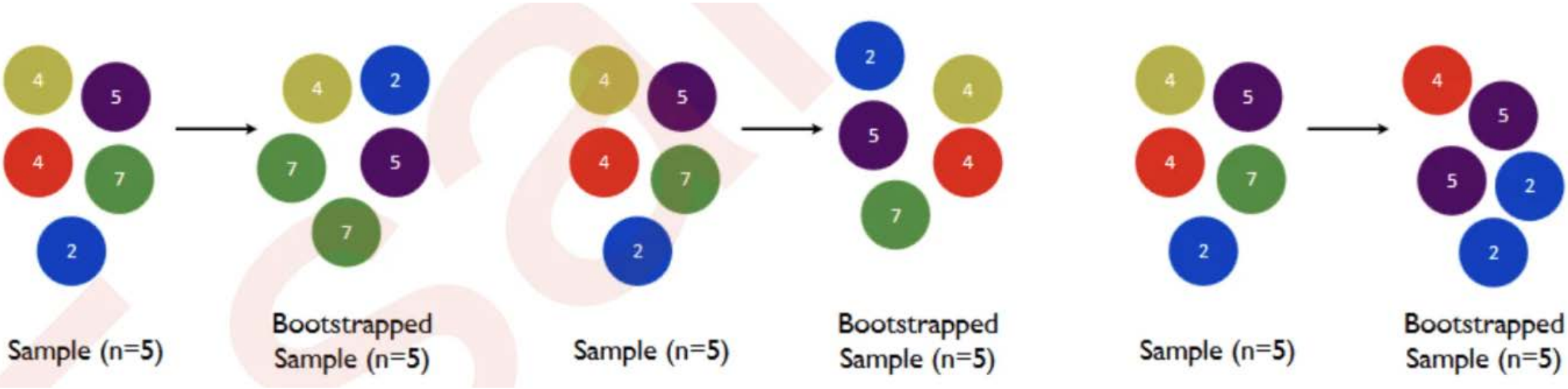
БУТСТРЭП

Метод бутстрэпа заключается в следующем

Пусть имеется выборка X размера N. Равномерно возьмем из выборки N объектов с возвращением. Это означает, что мы будем N раз выбирать произвольный объект выборки (считаем, что каждый объект «достается» с одинаковой вероятностью $1/N$), причем каждый раз мы выбираем из всех исходных N объектов. Можно представить себе мешок, из которого достают шарики: выбранный на каком-то шаге шарик возвращается обратно в мешок, и следующий выбор опять делается равновероятно из того же числа шариков

Отметим, что из-за возвращения среди них окажутся повторы. Обозначим новую выборку через X_1. Повторяя процедуру M раз, сгенерируем M подвыборок $X_1,\ldots,X_M$. Теперь мы имеем достаточно большое число выборок и можем оценивать различные статистики исходного распределения

ИЛЛЮСТРАЦИЯ БУТСТРЭПА



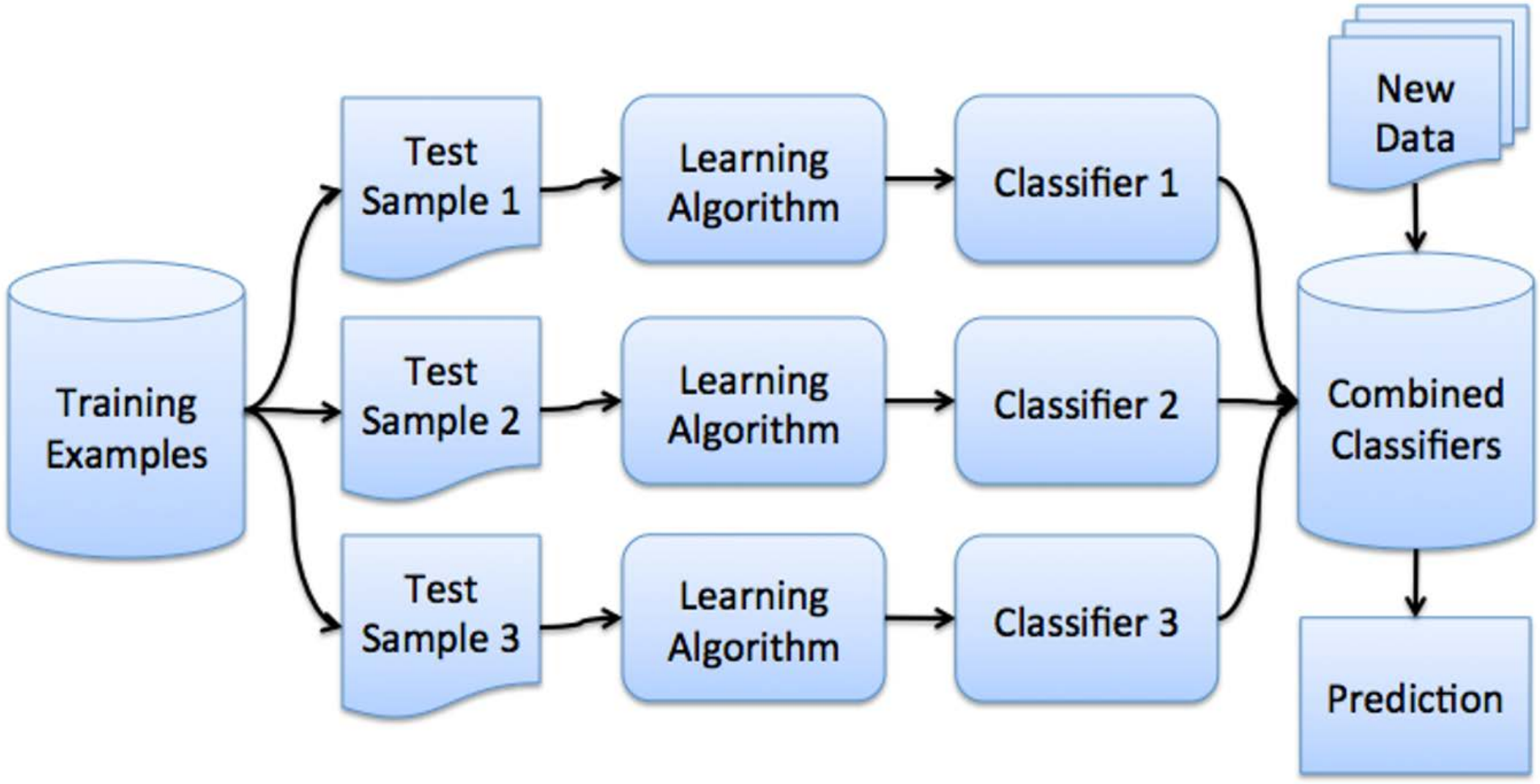
БЭГГИНГ

Bagging (от Bootstrap aggregation) — это один из первых и самых простых видов ансамблей. Он был придуман Ле́о Бре́йманом в 1994 году. Бэггинг основан на статистическом методе бутстрэпа, который позволяет оценивать многие статистики сложных распределений.

Пусть имеется обучающая выборка X. С помощью бутстрэпа сгенерируем из неё выборки X1,…,XM. Теперь на каждой выборке обучим свой классификатор ai(x). Итоговый классификатор будет **усреднять** ответы всех этих алгоритмов (в случае классификации это соответствует голосованию):

$$a(x) = \frac{1}{M} \sum_{i=1}^{M} a_i(x)$$

ИЛЛЮСТРАЦИЯ БЭГГИНГА



ЭФФЕКТИВНОСТЬ БЭГГИНГА

Бэггинг позволяет снизить дисперсию обучаемого классификатора, уменьшая величину, на сколько ошибка будет отличаться, если обучать модель на разных наборах данных, или другими словами, *предотвращает переобучение*

Эффективность бэггинга достигается благодаря тому, что базовые алгоритмы, обученные по различным подвыборкам, получаются достаточно различными, и их ошибки взаимно компенсируются при голосовании, а также за счёт того, что объекты-выбросы могут не попадать в некоторые обучающие подвыборки

Бэггинг эффективен на малых выборках, когда исключение даже малой части обучающих объектов приводит к построению существенно различных базовых классификаторов. В случае больших выборок обычно генерируют подвыборки существенно меньшей длины

Следует отметить, что рассмотренный пример не очень применим на практике, поскольку мы сделали предположение о некоррелированности ошибок, что редко выполняется. Если это предположение неверно, то уменьшение ошибки оказывается не таким значительным. В следующих лекциях мы рассмотрим более сложные методы объединения алгоритмов в композицию, которые позволяют добиться высокого качества в реальных задачах

ОШИБКА OUT-OF-BAG

Каждое дерево строится с использованием разных образцов бутстрэпа из исходных данных. Примерно 37% примеров остаются вне выборки бутстрэпа и не используются при построении *k*-го дерева.

Получается, что каждый базовый алгоритм обучается на ~63% исходных объектов. Значит, на оставшихся ~37% его можно сразу проверять.

Out-of-Bag оценка — это усредненная оценка базовых алгоритмов на тех ~37% данных, на которых они не обучались.

СЛУЧАЙНЫЙ ЛЕС

Случайный лес (Random forest) — алгоритм машинного обучения, заключающийся в использовании комитета (ансамбля) решающих деревьев.

Алгоритм сочетает в себе две основные идеи: метод бэггинга и метод случайных подпространств.

Алгоритм применяется для задач классификации, регрессии и кластеризации.

АЛГОРИТМ ОБУЧЕНИЯ КЛАССИФИКАТОРА (1)

Пусть обучающая выборка состоит из *N* образцов, размерность пространства признаков равна *M*, и задан параметр *m* (в задачах классификации обычно $m = \sqrt{M}$, а в задачах регрессии $m = M/3$), как неполное количество признаков для обучения

Наиболее распространённый способ построения дерева:

- Сгенерируем случайную подвыборку **с повторениями** размером *N* из обучающей выборки. Таким образом, некоторые образцы попадут в неё два или более раза, а в среднем $N(1 - 1/N)^N$ (при больших *N* примерно *N/e*) образцов не войдут в неё вообще. Те образцы, которые не попали в выборку, называются **out-of-bag** (неотобранные)
- Построим решающее дерево, классифицирующее образцы данной подвыборки, причём в ходе создания очередного узла дерева будем выбирать набор признаков, на основе которых производится разбиение (не из всех *M* признаков, а лишь из *m* случайно выбранных). Выбор наилучшего из этих *m* признаков может осуществляться по одному из известных критериев (прирост информации, критерий Джини)
- Дерево строится до полного исчерпания подвыборки и не подвергается процедуре прунинга (англ. pruning — отсечение ветвей)

АЛГОРИТМ ОБУЧЕНИЯ КЛАССИФИКАТОРА (2)

Классификация объектов проводится путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев

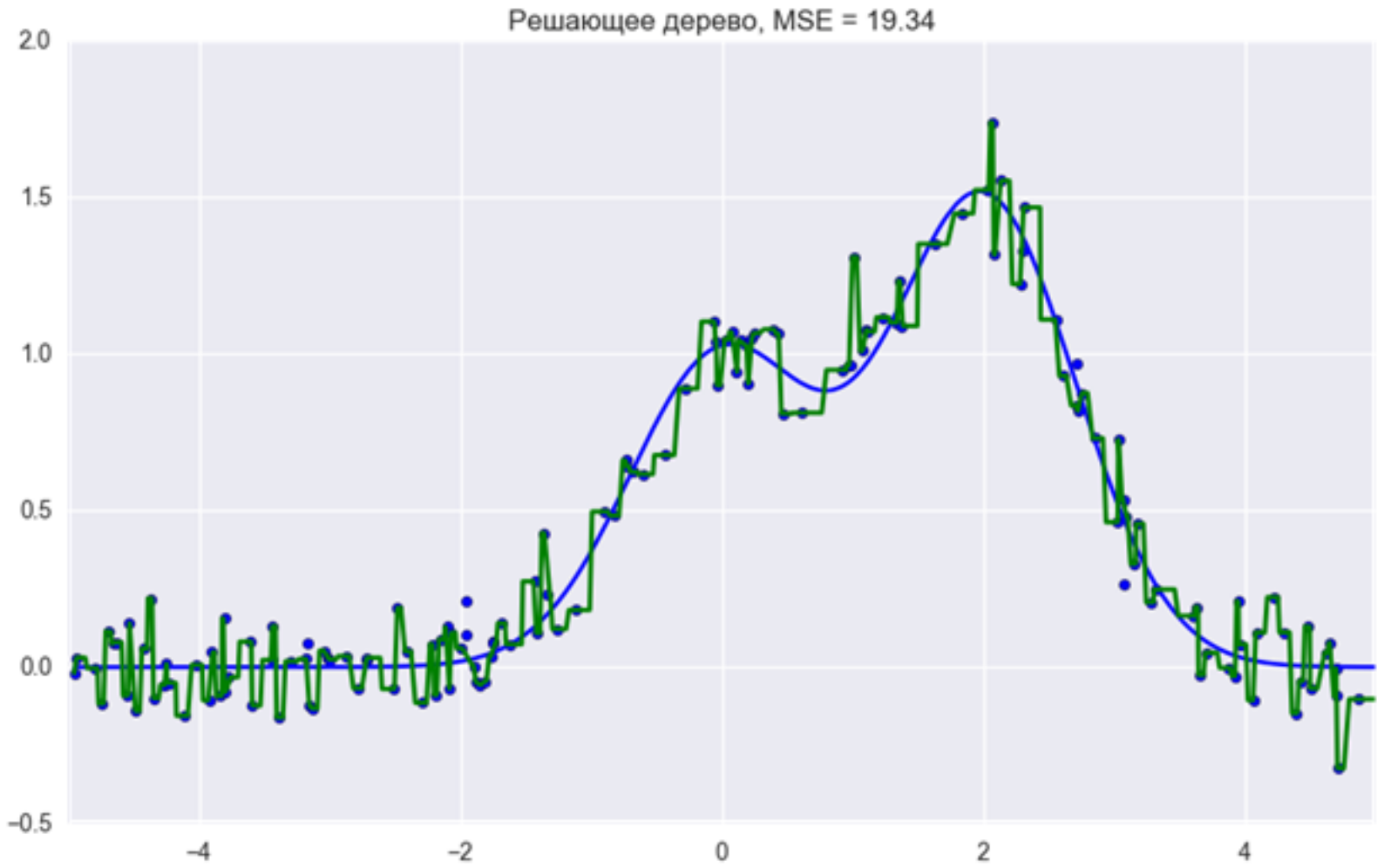
Оптимальное число деревьев подбирается таким образом, чтобы минимизировать ошибку классификатора на тестовой выборке. В случае её отсутствия, минимизируется оценка ошибки out-of-bag: тех образцов, которые не попали в обучающую подвыборку за счёт повторений (их примерно N/e)

Решающие деревья являются хорошим семейством базовых классификаторов для бэггинга, поскольку они достаточно сложны и могут достигать нулевой ошибки на любой выборке

Метод случайных подпространств позволяет снизить коррелированность между деревьями и избежать переобучения. Базовые алгоритмы обучаются на различных подмножествах признакового описания, которые также выделяются случайным образом

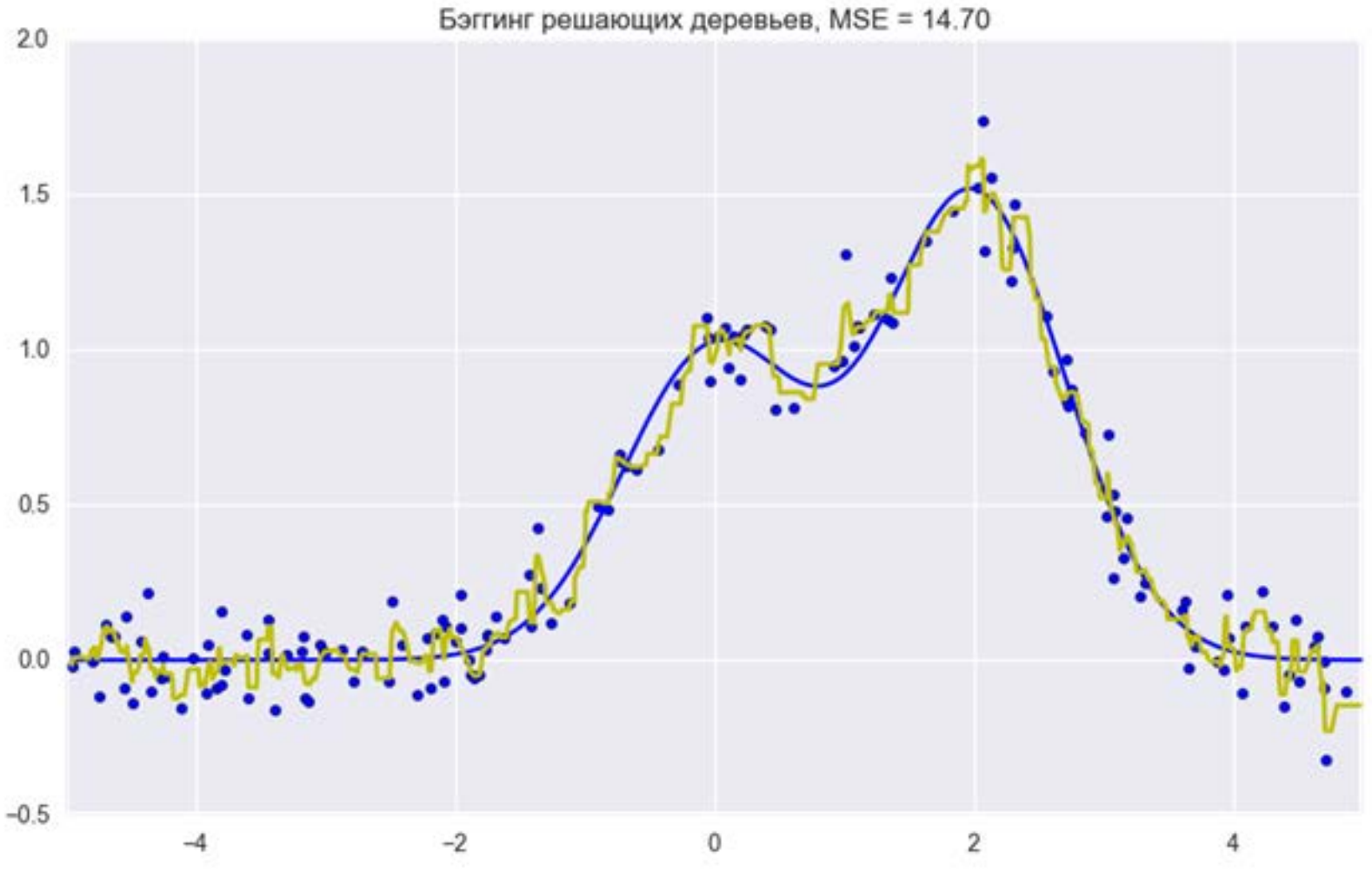
Таким образом, случайный лес — это бэггинг над решающими деревьями, при обучении которых для каждого разбиения признаки выбираются из некоторого случайного подмножества признаков

СРАВНЕНИЕ МЕТОДОВ ДЛЯ РЕГРЕССИИ (1)



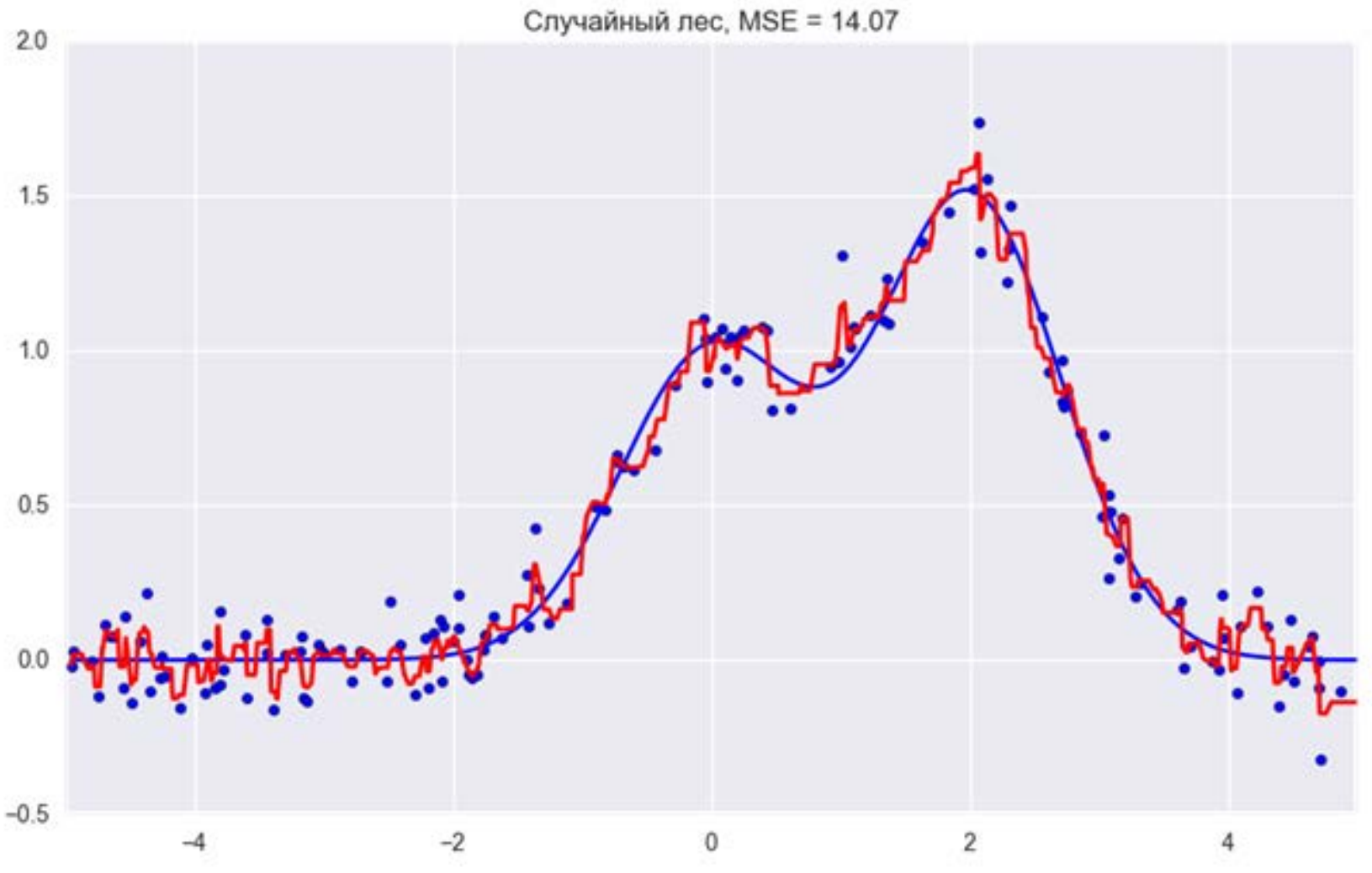
СРАВНЕНИЕ МЕТОДОВ ДЛЯ РЕГРЕССИИ (2)

Бэггинг из 10 деревьев решений



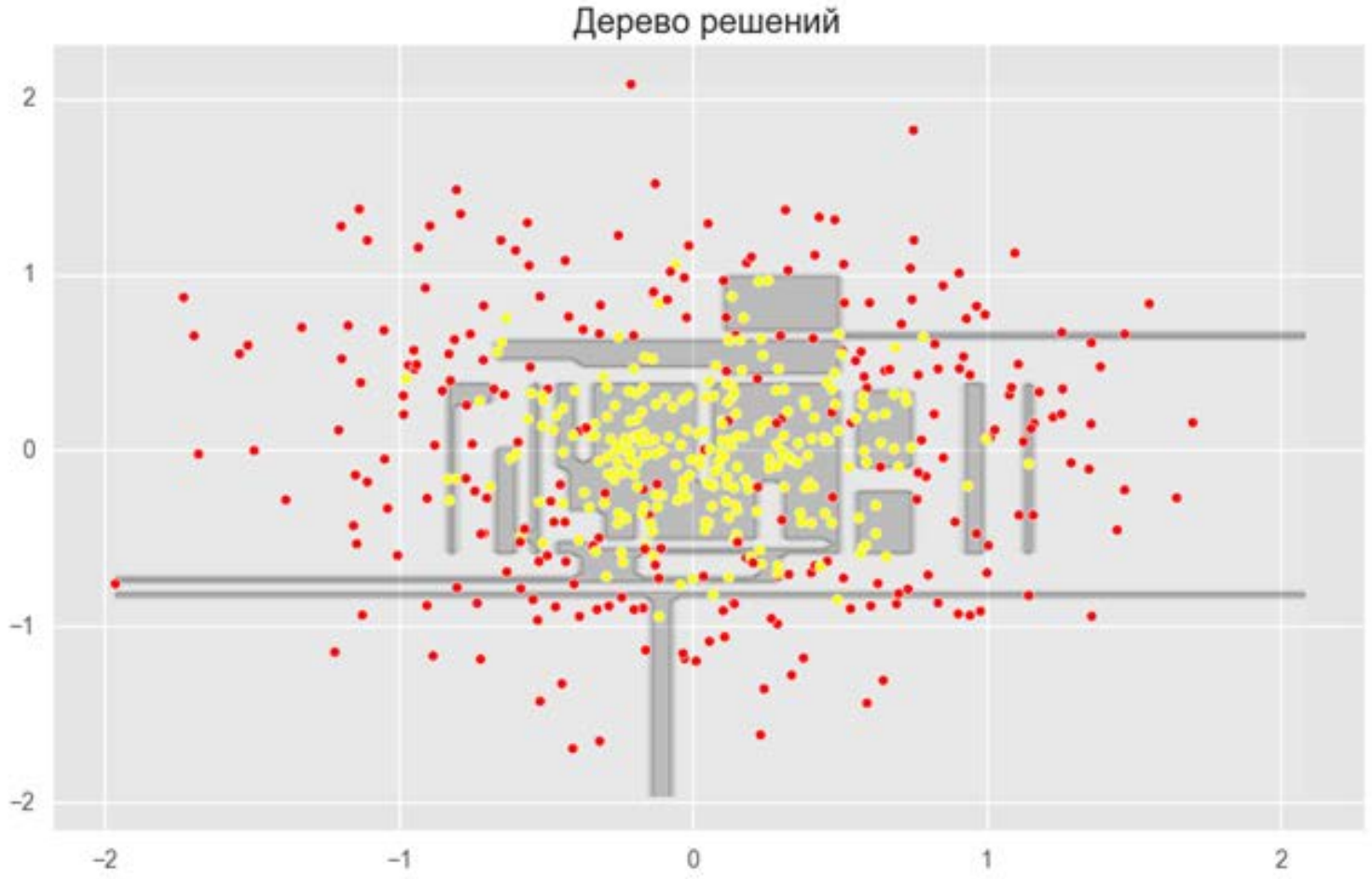
СРАВНЕНИЕ МЕТОДОВ ДЛЯ РЕГРЕССИИ (3)

Случайный лес из 10 деревьев решений

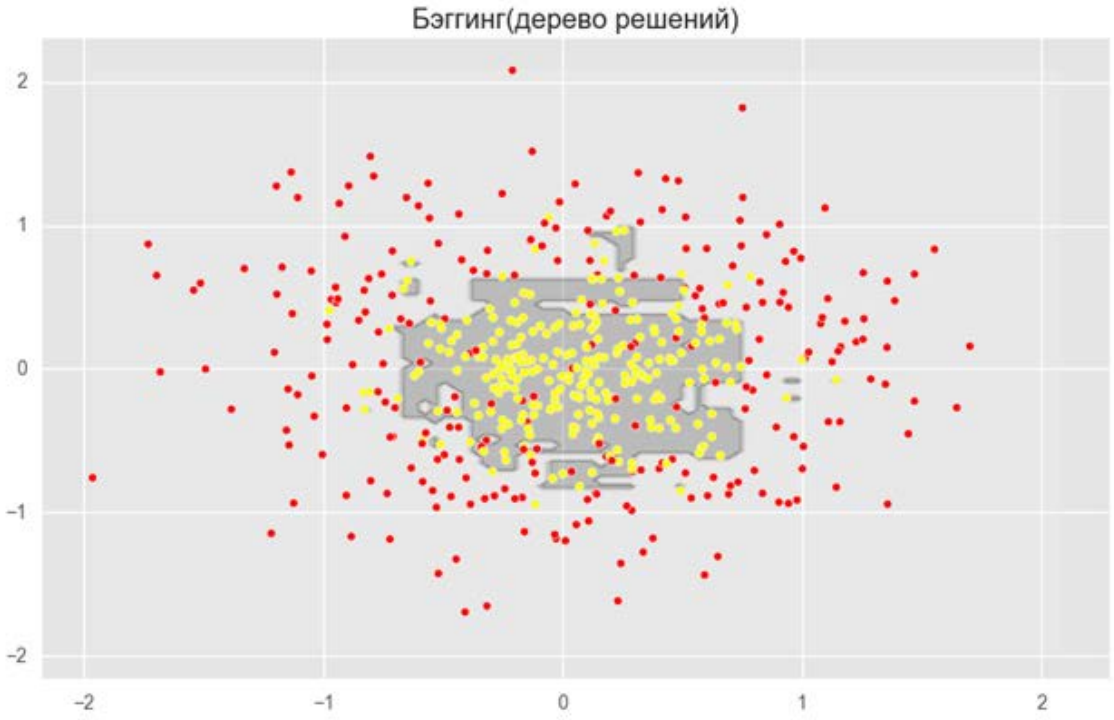


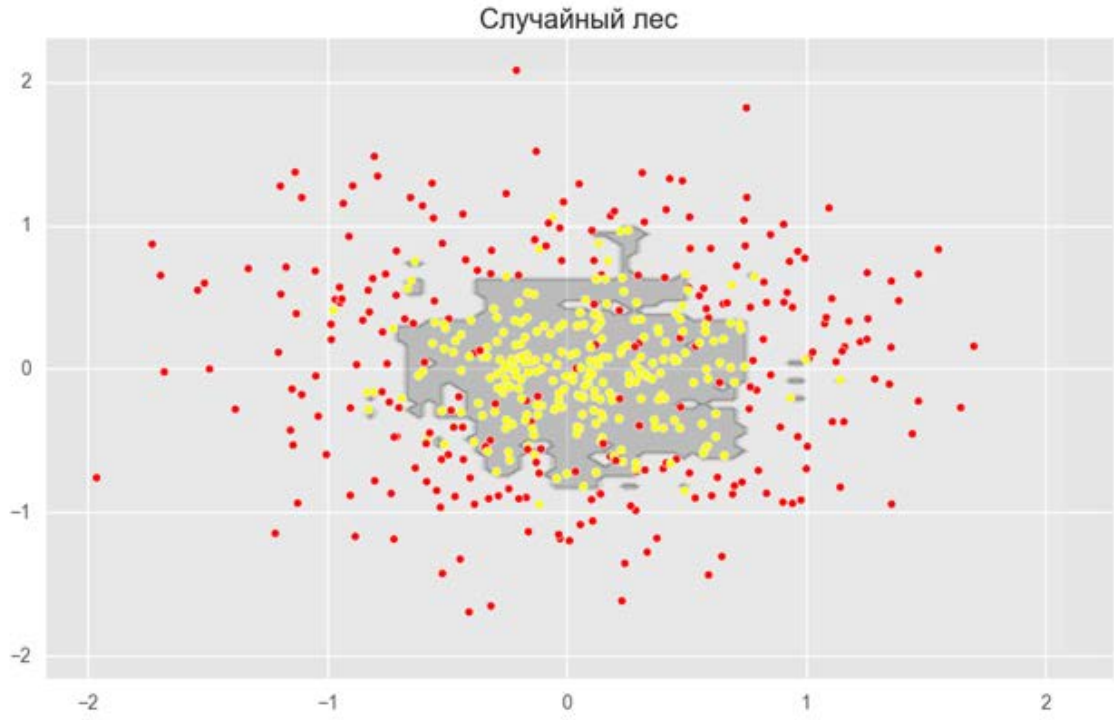
СРАВНЕНИЕ МЕТОДОВ ДЛЯ КЛАССИФИКАЦИИ (1)

Разделяющая граница дерева решений очень «рваная» и на ней много острых углов, что говорит о переобучении и слабой обобщающей способности.



СРАВНЕНИЕ МЕТОДОВ ДЛЯ КЛАССИФИКАЦИИ (2)





У бэггинга и случайного леса граница достаточно сглаженная и практически нет признаков переобучения.

ПЛЮСЫ СЛУЧАЙНОГО ЛЕСА

- Высокая точность предсказания
- Нечувствительность к выбросам в данных
- Нечувствительность к масштабированию значений признаков, что связано с выбором случайных подпространств;
- Не требует тщательной настройки параметров
- Способен эффективно обрабатывать данные с большим числом признаков и классов
- Одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки
- Редко переобучается
- Для случайного леса существуют методы оценивания значимости отдельных признаков в модели
- Хорошо работает с пропущенными данными
- Предполагает возможность сбалансировать вес каждого класса на всей выборке, либо на подвыборке каждого дерева
- Вычисляет близость между парами объектов, что может использоваться при кластеризации, обнаружении выбросов или дают интересные представления данных

МИНУСЫ СЛУЧАЙНОГО ЛЕСА

- В отличие от одного дерева, результаты случайного леса сложнее интерпретировать
- Нет формальных выводов, доступных для оценки важности переменных
- Случайный лес не умеет экстраполировать
- Алгоритм склонен к переобучению на некоторых задачах, особенно на зашумленных данных
- Для данных, включающих категориальные переменные с различным количеством уровней, случайные леса предвзяты в пользу признаков с большим количеством уровней
- Больший размер получающихся моделей