Современные методы анализа данных и машинного обучения

Тема 6. Лекция 9

Классическое машинное обучение. Деревья решений. Ансамбли

Юрий Саночкин

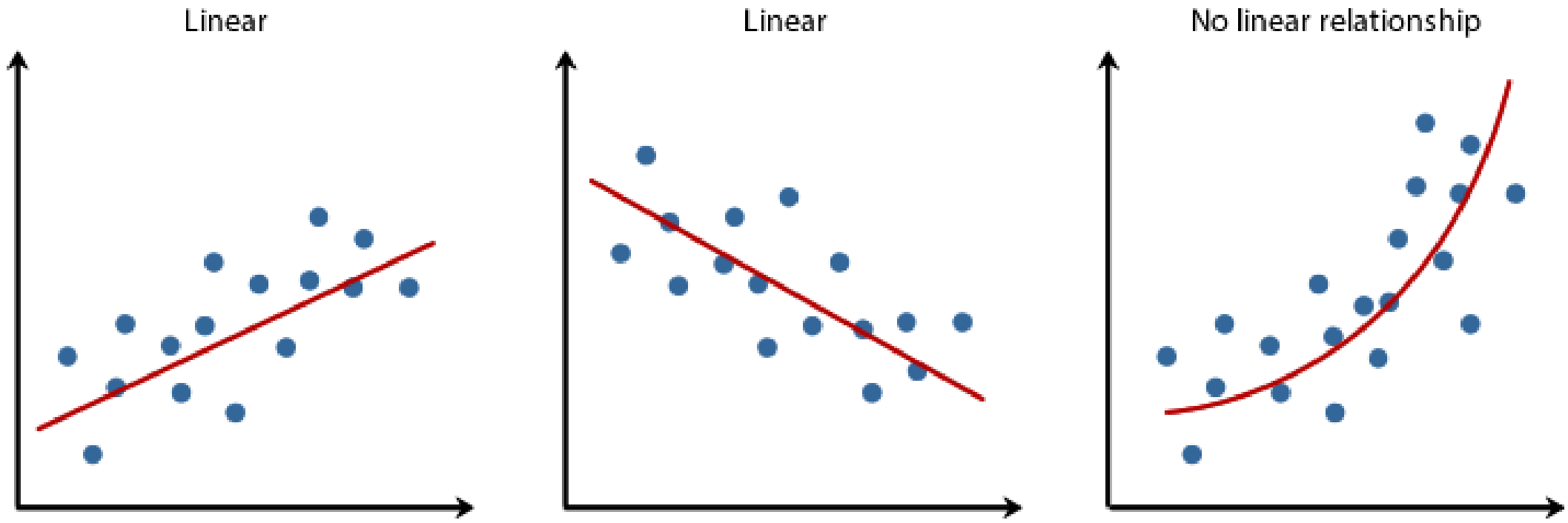
ysanochkin@hse.ru

НИУ ВШЭ, 2024

Линейные и нелинейные зависимости

- Поскольку сегодня мы с вами будем активно обсуждать совершенно новые модели машинного обучения, давайте, прежде чем начать, сделаем для них небольшой сравнительный обзор с точки зрения линейности/нелинейности зависимостей, которые они могут выявлять.
- Кстати, что такое линейные/нелинейные зависимости?

Линейные и нелинейные зависимости



Линейные и нелинейные зависимости

- Поскольку сегодня мы с вами будем активно обсуждать совершенно новые модели машинного обучения, давайте, прежде чем начать, сделаем для них небольшой сравнительный обзор с точки зрения линейности/нелинейности зависимостей, которые они могут выявлять.
- Какие из моделей машинного обучения, среди тех, что мы обсуждали, выявляют линейные, а какие — нелинейные зависимости?
- Приведите и другие примеры моделей, если знаете.

No.	Machine Learning Model	Category
1	Linear Regression (LR)	Linear
2	Linear Discriminant Analysis (LDA)	Linear
3	Support Vector Machine (SVM)	Linear
4	Quadratic Discriminant Analysis (QDA)	Non-linear
5	Random Forest (RF)	Non-linear
6	K-Nearest Neighbors (KNN)	Non-linear
7	Nearest Centroid	Linear
8	Naive Bayes	Linear
9	Perceptron	Linear
10	Decision Tree (DT)	Non-linear
11	Dummy	Non-linear
12	Neural Networks	Non-linear

No.	Machine Learning Model	Category
1	Linear Regression (LR)	Linear
2	Linear Discriminant Analysis (LDA)	Linear
3	Support Vector Machine (SVM)	Linear
4	Quadratic Discriminant Analysis (QDA)	Non-linear
5	Random Forest (RF)	Non-linear
6	K-Nearest Neighbors (KNN)	Non-linear
7	Nearest Centroid	Linear
8	Naive Bayes	Linear
9	Perceptron	Linear
10	Decision Tree (DT)	Non-linear
11	Dummy	Non-linear
12	Neural Networks	Non-linear

Наше основное обсуждение на сегодня

Решающие деревья

Решающие деревья

- Решающее дерево — один из наиболее интуитивных алгоритмов ML.
- Возможно, некоторые из вас использовали его для принятия решений ещё задолго до нашего курса :)

Решающие деревья

- Решающее дерево — один из наиболее интуитивных алгоритмов ML.
- Возможно, некоторые из вас использовали его для принятия решений ещё задолго до нашего курса :)

Дерево Решений

Решающие деревья

Задача решающего дерева —
разбить обучающую выборку на подмножества,
в которых ответ более-менее очевиден.

Решающие деревья

- Давайте для начала очень верхнеуровнево рассмотрим алгоритм построения решающего дерева, а затем пойдем уже подробно по каждому этапу, разбираясь во всех деталях.
- Итак, поехали!

Алгоритм построения решающего дерева

Алгоритм построения решающего дерева

1. Фиксируем критерий остановки обучения и критерий остановки разбиения, а также принцип выбора признака и порога для разбиения.

Алгоритм построения решающего дерева

1. Фиксируем критерий остановки обучения и критерий остановки разбиения, а также принцип выбора признака и порога для разбиения.
2. Начинаем с корня, в нём — вся обучающая выборка.

Алгоритм построения решающего дерева

1. Фиксируем критерий остановки обучения и критерий остановки разбиения, а также принцип выбора признака и порога для разбиения.
2. Начинаем с корня, в нём — вся обучающая выборка.
3. Проверяем выборку по критерию остановки обучения и оцениваем точности построенного дерева.

Алгоритм построения решающего дерева

1. Фиксируем критерий остановки обучения и критерий остановки разбиения, а также принцип выбора признака и порога для разбиения.
2. Начинаем с корня, в нём — вся обучающая выборка.
3. Проверяем выборку по критерию остановки обучения и оцениваем точности построенного дерева.
4. Перебираем признаки и пороги для всех вершин, для которых не выполнен критерий остановки разбиения, и для каждого выбираем лучшую пару (признак, порог) согласно зафиксированному принципу.

Алгоритм построения решающего дерева

5. Для каждой зафиксированной пары (признак, порог) — разбиваем данные в вершине на две части так, чтобы объекты со значениями выбранного признака меньше порога шли в левую вершину, а со значениями больше или равными порогу — в правую.

Алгоритм построения решающего дерева

5. Для каждой зафиксированной пары (признак, порог) — разбиваем данные в вершине на две части так, чтобы объекты со значениями выбранного признака меньше порога шли в левую вершину, а со значениями больше или равными порогу — в правую.
6. Повторяем шаги 3-5, пока это возможно или пока это позволяет критерий остановки обучения.

Алгоритм построения решающего дерева

- Вот такой вот прекрасный алгоритм мы с вами сформулировали и получили.
- Легкотня, правда? :)

Алгоритм построения решающего дерева

- Вот такой вот прекрасный алгоритм мы с вами сформулировали и получили.
- Легкотня, правда? :)
- Всего лишь-то и осталось разобраться с несколькими вопросами

Алгоритм построения решающего дерева

- Всего лишь-то и осталось разобраться с несколькими вопросами:
  1. Какие критерии остановки обучения существуют и какие из них нужно использовать?

Алгоритм построения решающего дерева

- Всего лишь-то и осталось разобраться с несколькими вопросами:
  1. Какие критерии остановки обучения существуют и какие из них нужно использовать?
  2. Какие критерии остановки разбиения существуют и какие из них нужно использовать?

Алгоритм построения решающего дерева

- Всего лишь-то и осталось разобраться с несколькими вопросами:
  1. Какие критерии остановки обучения существуют и какие из них нужно использовать?
  2. Какие критерии остановки разбиения существуют и какие из них нужно использовать?
  3. Какие есть способы выбрать признак и порог для разбиения?

Алгоритм построения решающего дерева

- Всего лишь-то и осталось разобраться с несколькими вопросами:
  1. Какие критерии остановки обучения существуют и какие из них нужно использовать?
  2. Какие критерии остановки разбиения существуют и какие из них нужно использовать?
  3. Какие есть способы выбрать признак и порог для разбиения?
  4. Как будет происходить разбиение признаков различных типов?

Алгоритм построения решающего дерева

- Всего лишь-то и осталось разобраться с несколькими вопросами:
  1. Какие критерии остановки обучения существуют и какие из них нужно использовать?
  2. Какие критерии остановки разбиения существуют и какие из них нужно использовать?
  3. Какие есть способы выбрать признак и порог для разбиения?
  4. Как будет происходить разбиение признаков различных типов?
  5. Как получить предсказания такого алгоритма?

Алгоритм построения решающего дерева

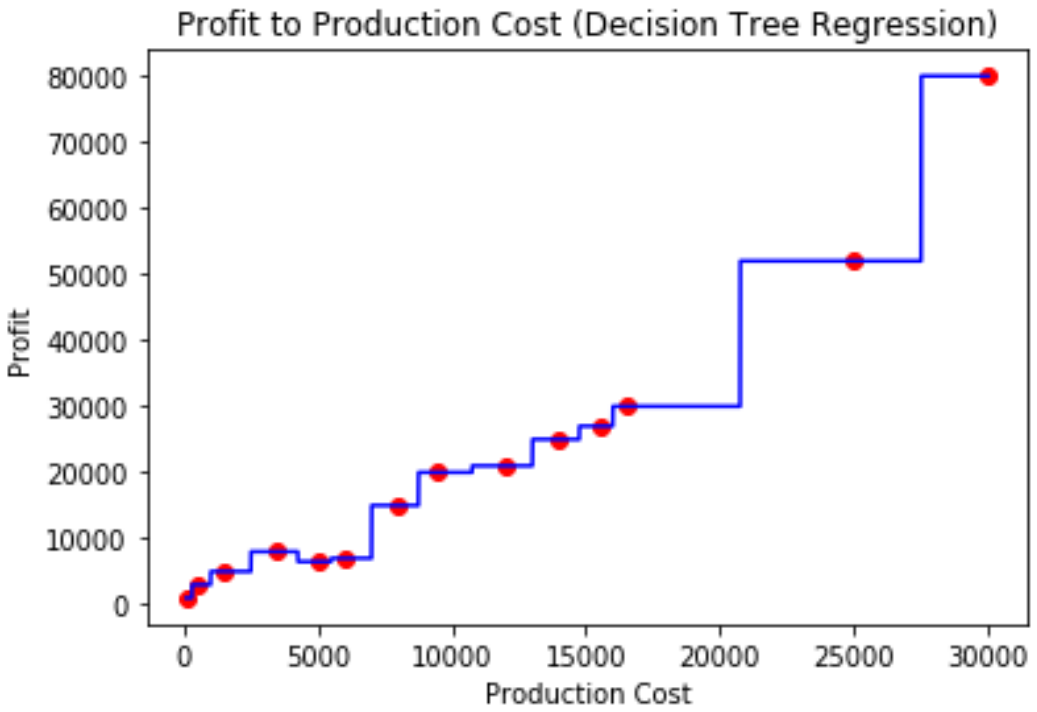
- Всего лишь-то и осталось разобраться с несколькими вопросами:
  1. Какие критерии остановки обучения существуют и какие из них нужно использовать?
  2. Какие критерии остановки разбиения существуют и какие из них нужно использовать?
  3. Какие есть способы выбрать признак и порог для разбиения?
  4. Как будет происходить разбиение признаков различных типов?
  5. Как получить предсказания такого алгоритма?

- И, конечно же, вновь, как самые последовательные люди, мы начнем отвечать на эти вопросы в обратном порядке :)

Как получить предсказания алгоритма?

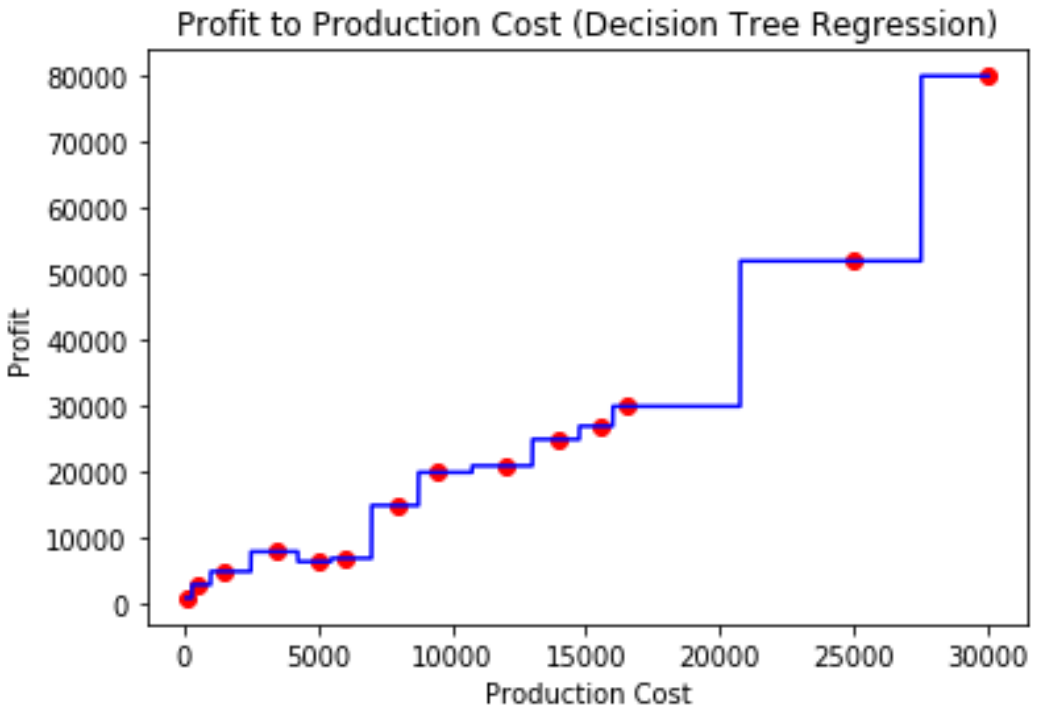
Как получить предсказания алгоритма?

Регрессия: среднее значение в листе
(или медиана, или любая другая статистика)

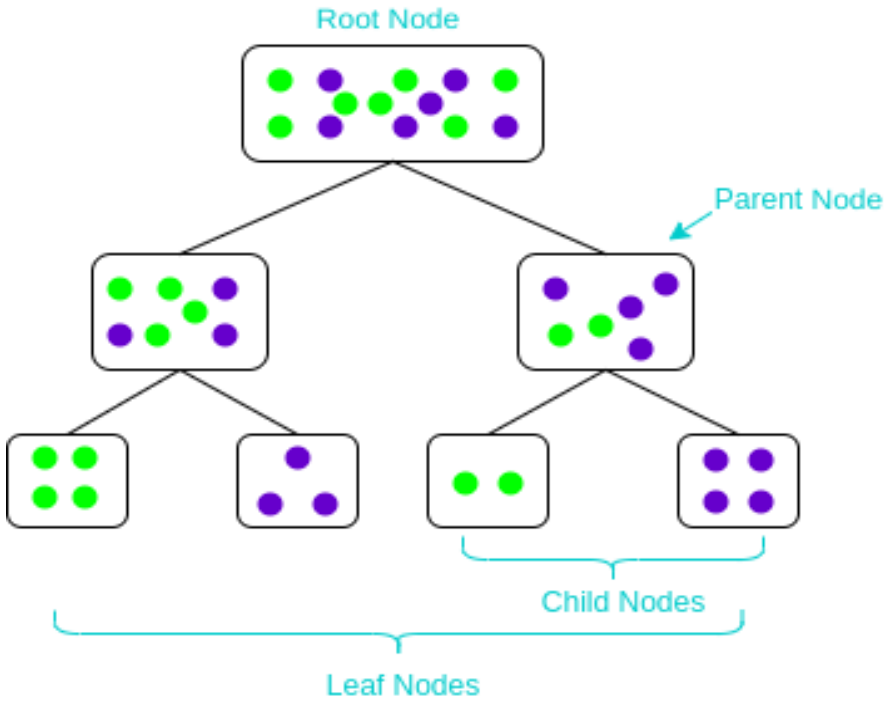


Как получить предсказания алгоритма?

Регрессия: среднее значение в листе
(или медиана, или любая другая статистика)

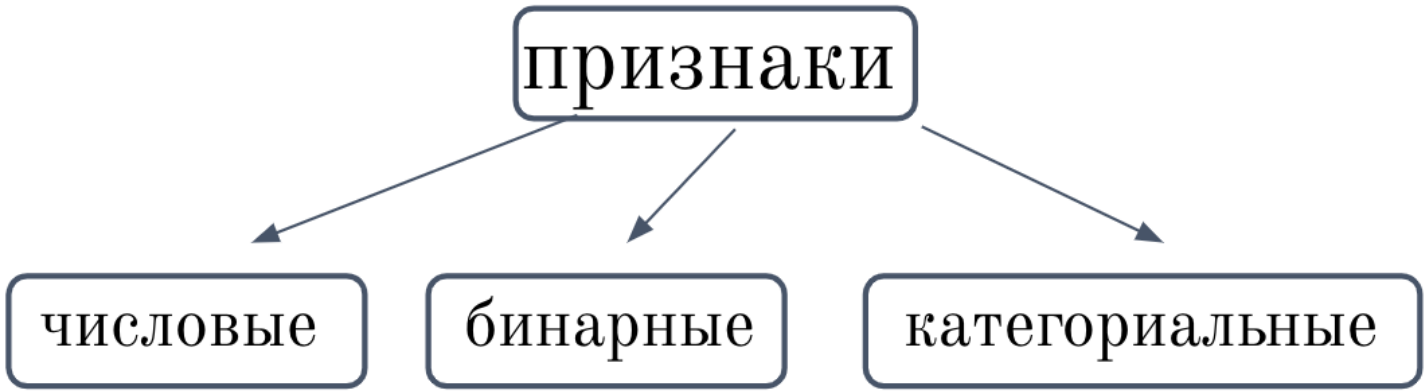


Классификация: самый частотный
класс в листе (или его частота).

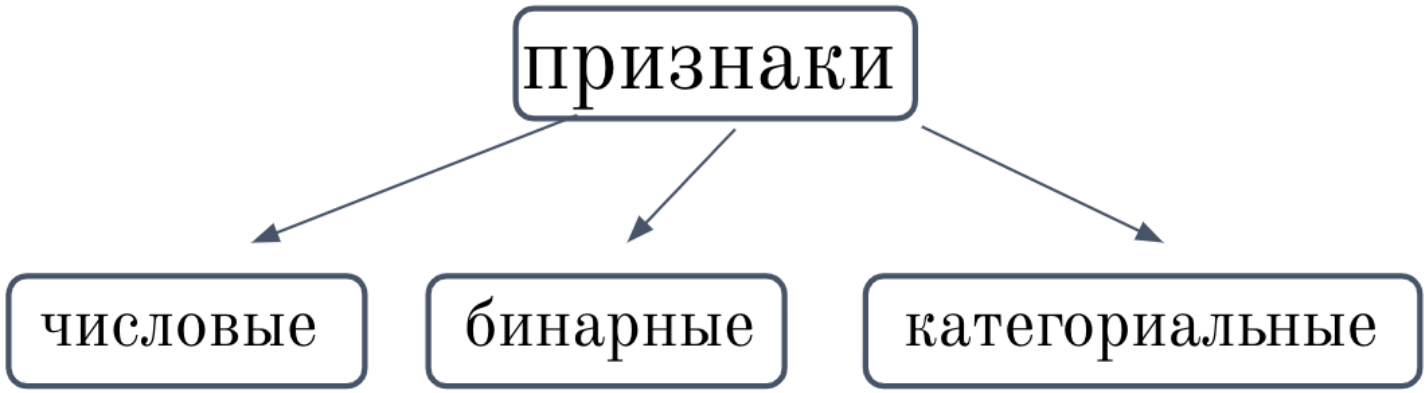


Как происходит разбиение признаков различных типов?

Как происходит разбиение признаков различных типов?

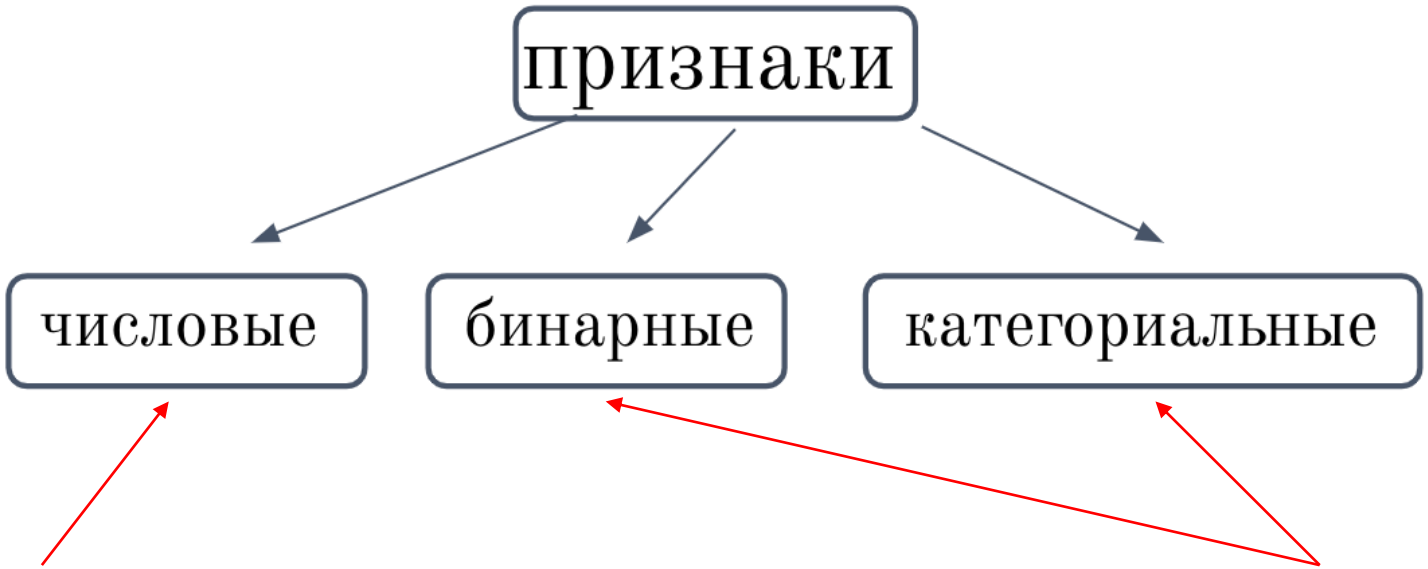


Как происходит разбиение признаков различных типов?



Разбиваются по порогу:
Если признак меньше порога, то объект отправляется в левую группу, если больше или равен порога, то — в правую.

Как происходит разбиение признаков различных типов?



Разбиваются по порогу:
Если признак меньше порога, то объект отправляется в левую группу, если больше или равен порога, то — в правую.

Разбиение один против всех:
Если значение признака равно выбранному порогу, то объект отправляется в левую группу, иначе — отправляется в правую.

Как выбрать признак и порог разбиения?

- Данный вопрос является, по сути, одним из самых важных в теории решающих деревьев.

Как выбрать признак и порог разбиения?

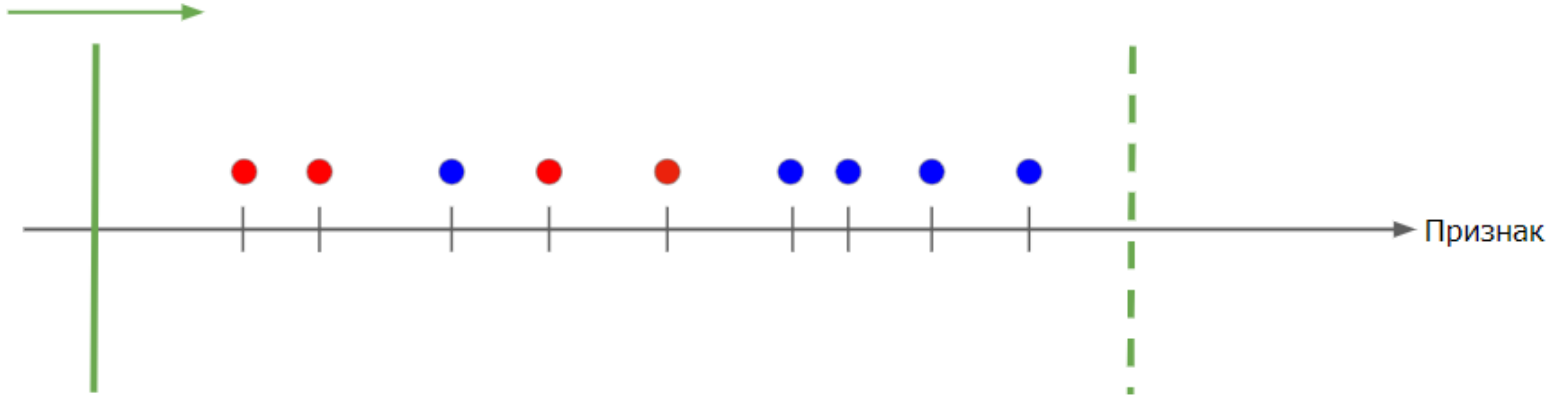
- Данный вопрос является, по сути, одним из самых важных в теории решающих деревьев.
- Давайте для начала предположим, что за нас уже выбрали признак и от нас требуется выбрать только порог.
- Мы преследуем цель сделать так, чтобы элементы выборок в одних дочерних вершинах отличались не сильно. То есть мы хотим сделать выборки однородными.

Как выбрать признак и порог разбиения?

- Сам процесс проверки различных порогов прост – мы переберем все возможные и выберем лучший с точки зрения уменьшения неоднородности в получившихся группах.

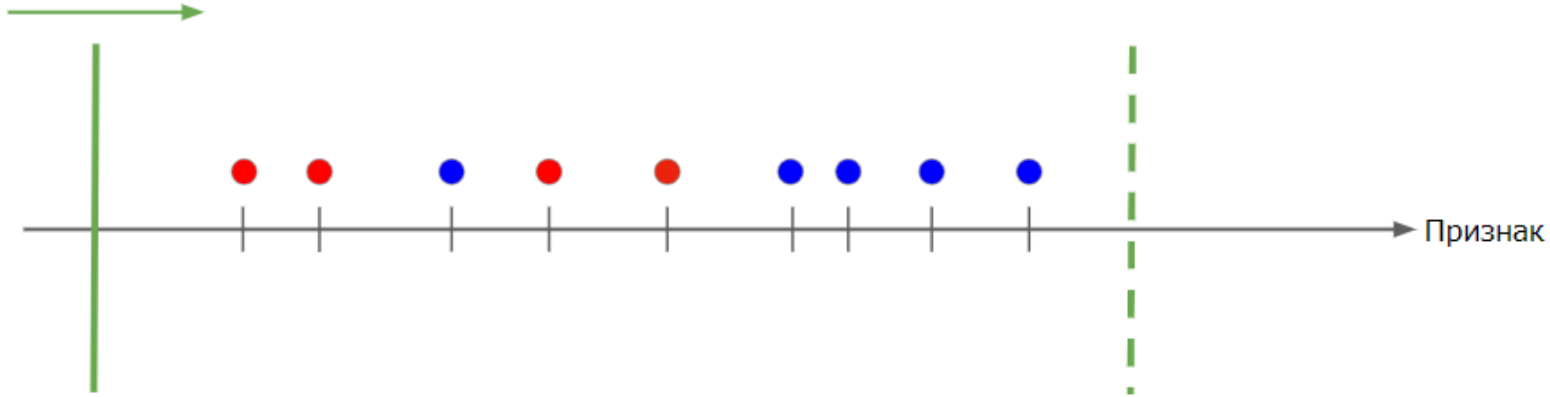
Как выбрать признак и порог разбиения?

- Сам процесс проверки различных порогов прост – мы переберем все возможные и выберем лучший с точки зрения уменьшения неоднородности в получившихся группах.



Как выбрать признак и порог разбиения?

- Сам процесс проверки различных порогов прост – мы переберем все возможные и выберем лучший с точки зрения уменьшения неоднородности в получившихся группах.



- Но тут сразу возникает другой интересный вопрос: а что же является этим критерием однородности/неоднородности для наших данных?

Как выбрать признак и порог разбиения?

- Понятие неоднородности для задачи регрессии и задачи классификации отличаются, поэтому и рассматривать мы их будем отдельно.

Как выбрать признак и порог разбиения?

- Понятие неоднородности для задачи регрессии и задачи классификации отличаются, поэтому и рассматривать мы их будем отдельно.
- Для задачи регрессии неоднородность будет считаться, по сути, просто как дисперсия таргета в выборке!
- Именно дисперсия будет говорить нам о мере хаотичности/похожести наших данных.

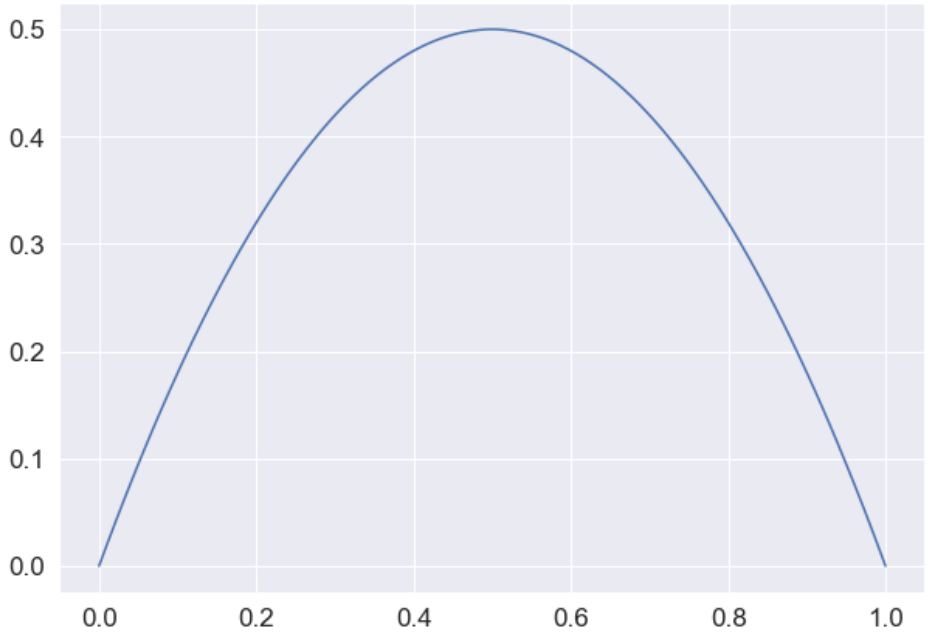
Как выбрать признак и порог разбиения?

- Для задачи классификации легко понять, что неоднородность выборки должна быть связана с долей классов в выборке: если доминирует любой из классов, то неоднородность должна быть низкая, а если распределение по классам примерно одинаковое, то напротив — высокая.
- Поэтому для подсчета однородности выборки в основном используют две метрики.

Как выбрать признак и порог разбиения?

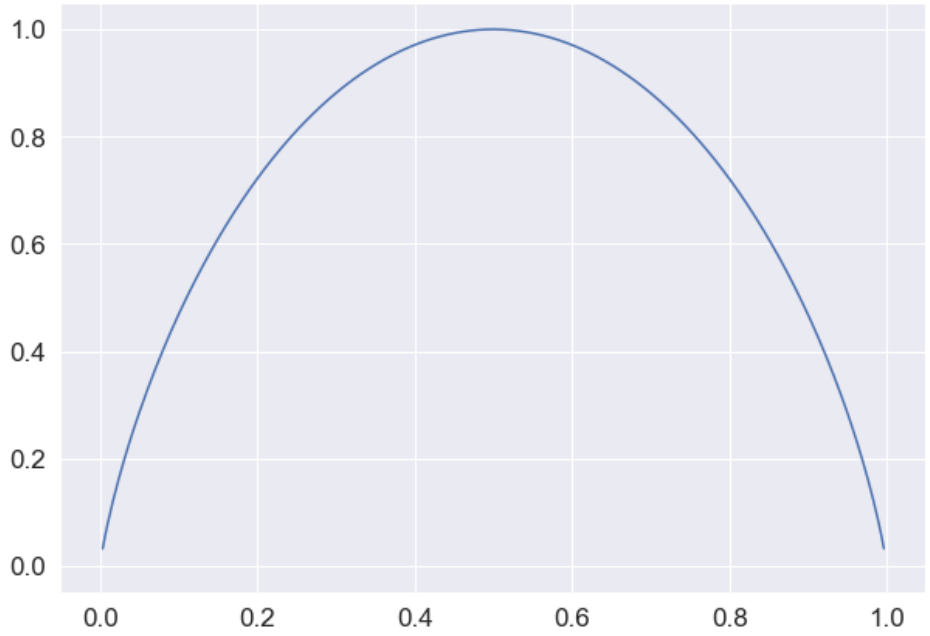
Индекс Джини

$$Gini(G) = 1 - \sum_{i=1}^{n} \left(\frac{N_i}{N}\right)^2$$



Энтропия

$$H(G) = \sum_{i=1}^{n} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right)$$



Как выбрать признак и порог разбиения?

- Теперь мы знаем, как считать неоднородность в выборке, и готовы ответить на вопрос — как же построить разбиение выборки.

Как выбрать признак и порог разбиения?

- Теперь мы знаем, как считать неоднородность в выборке, и готовы ответить на вопрос — как же построить разбиение выборки.
- На самом деле, ответ находится на поверхности — нужно сделать такое разбиение, которое сильнее всего уменьшит неоднородность исходной выборки.

Как выбрать признак и порог разбиения?

- Теперь мы знаем, как считать неоднородность в выборке, и готовы ответить на вопрос — как же построить разбиение выборки.
- На самом деле, ответ находится на поверхности — нужно сделать такое разбиение, которое сильнее всего уменьшит неоднородность исходной выборки.
- Если выписать это формульно:

$$Q(R) = H(R) - \frac{|R_l|}{|R|} H(R_l) - \frac{|R_r|}{|R|} H(R_r) \qquad Q(R) \to \max$$

Как выбрать признак и порог разбиения?

- Теперь мы знаем, как считать неоднородность в выборке, и готовы ответить на вопрос — как же построить разбиение выборки.
- На самом деле, ответ находится на поверхности — нужно сделать такое разбиение, которое сильнее всего уменьшит неоднородность исходной выборки.
- Если выписать это формульно:

Прокомментируйте все обозначения в формуле!
Что показывает и для чего нужен дробный коэффициент у второго и третьего слагаемого?

$$Q(R) = H(R) - \frac{|R_l|}{|R|} H(R_l) - \frac{|R_r|}{|R|} H(R_r) \qquad Q(R) \to \max$$

Как выбрать признак и порог разбиения?

- Итак, мы с вами научились выбирать лучшее разбиение для зафиксированного признака, и единственное, что нам, по сути, осталось понять, — это то, как выбирать сам признак для дальнейшего поиска оптимального порога разбиения.

Как выбрать признак и порог разбиения?

- Итак, мы с вами научились выбирать лучшее разбиение для зафиксированного признака, и единственное, что нам, по сути, осталось понять, — это то, как выбирать сам признак для дальнейшего поиска оптимального порога разбиения.
- Но, если честно, тут всё ещё тривиальнее: мы просто будем перебирать для каждого признака все возможные пороги и выберем тот признак, для которого посчитанная величина $Q(R)$ будет максимальной!

Какие существуют критерии остановки обучения и разбиения?

- Последний вопрос, который нам нужно обсудить и прояснить, в отношении деревьев решений — касается критериев остановки обучения и разбиения. Давайте посмотрим на несколько самых распространенных из них.

Какие существуют критерии остановки обучения и разбиения?

- Последний вопрос, который нам нужно обсудить и прояснить, в отношении деревьев решений — касается критериев остановки обучения и разбиения. Давайте посмотрим на несколько самых распространенных из них.
  - **Ранняя остановка** — алгоритм будет остановлен, как только будет достигнуто заданное значение некоторого критерия, например процентной доли правильно распознанных примеров. Единственное преимущество — снижение времени обучения. Главный недостаток — ущерб точности дерева.

Какие существуют критерии остановки обучения и разбиения?

- Последний вопрос, который нам нужно обсудить и прояснить, в отношении деревьев решений — касается критериев остановки обучения и разбиения. Давайте посмотрим на несколько самых распространенных из них.
  - **Ранняя остановка** — алгоритм будет остановлен, как только будет достигнуто заданное значение некоторого критерия, например процентной доли правильно распознанных примеров. Единственное преимущество — снижение времени обучения. Главный недостаток — ущерб точности дерева.
  - **Ограничение глубины дерева** — задание максимального числа разбиений в ветвях, по достижении которого обучение останавливается. Данный метод также ведёт к снижению точности дерева.

Какие существуют критерии остановки обучения и разбиения?

- Последний вопрос, который нам нужно обсудить и прояснить, в отношении деревьев решений — касается критериев остановки обучения и разбиения. Давайте посмотрим на несколько самых распространенных из них.
  - **Задание минимально допустимого числа примеров в узле** — запретить алгоритму создавать узлы с числом примеров меньше заданного (например, 5). Это позволит избежать создания тривиальных разбиений и, соответственно, малозначимых правил.

Какие существуют критерии остановки обучения и разбиения?

- Последний вопрос, который нам нужно обсудить и прояснить, в отношении деревьев решений — касается критериев остановки обучения и разбиения. Давайте посмотрим на несколько самых распространенных из них.
  - **Задание минимально допустимого числа примеров в узле** — запретить алгоритму создавать узлы с числом примеров меньше заданного (например, 5). Это позволит избежать создания тривиальных разбиений и, соответственно, малозначимых правил.
  - **Остановка после исчезновения всех неоднородностей в листьях** — то есть остановка перед тем, как дальнейшее разбиение не принесет эффект.

Решающие деревья

- Ну что же — мы обсудили все вопросы и нюансы, касающиеся построения решающих деревьев, и теперь готовы взглянуть на алгоритм построения по-новому.
- Давайте еще раз освежим его в памяти!

Алгоритм построения решающего дерева

1. Фиксируем критерий остановки обучения и критерий остановки разбиения, а также принцип выбора признака и порога для разбиения.
2. Начинаем с корня, в нём — вся обучающая выборка.
3. Проверяем выборку по критерию остановки обучения и оцениваем точности построенного дерева.
4. Перебираем признаки и пороги для всех вершин, для которых не выполнен критерий остановки разбиения, и для каждого выбираем лучшую пару (признак, порог) согласно зафиксированному принципу.

Алгоритм построения решающего дерева

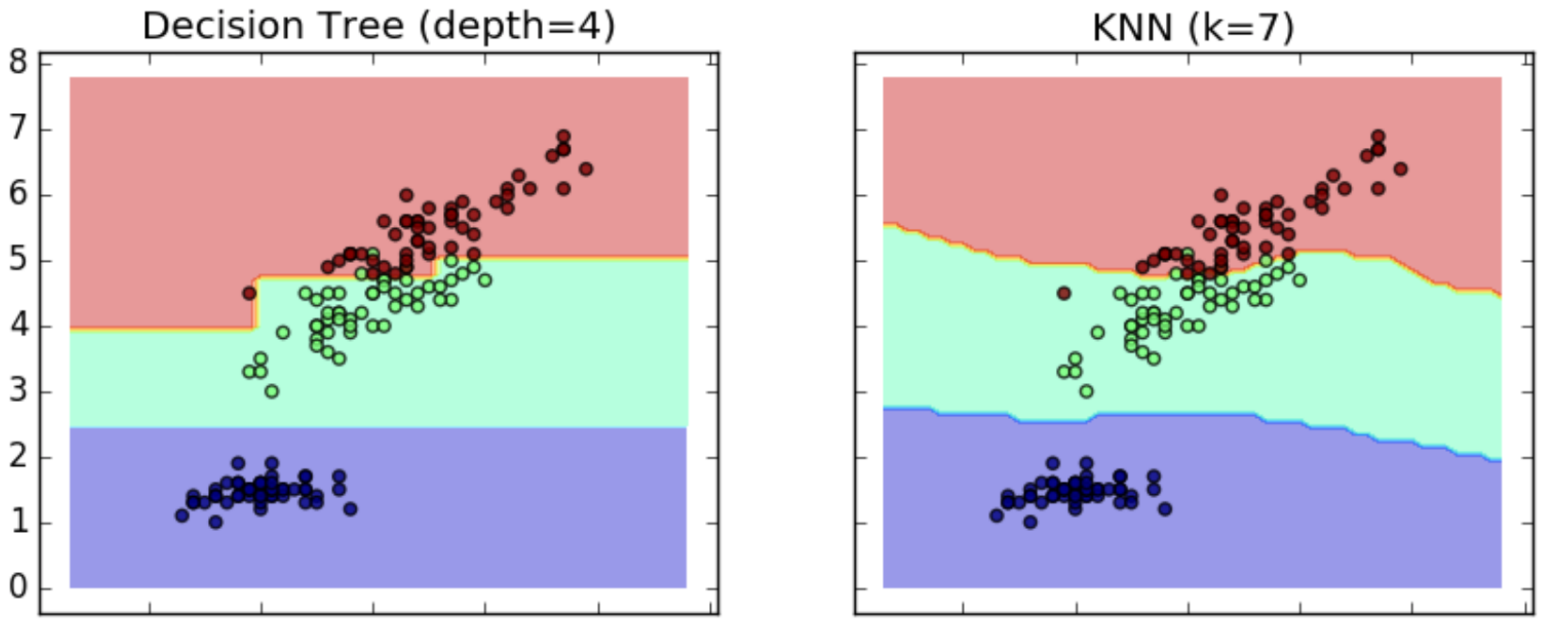
5. Для каждой зафиксированной пары (признак, порог) — разбиваем данные в вершине на две части так, чтобы объекты со значениями выбранного признака меньше порога шли в левую вершину, а со значениями больше или равными порогу — в правую.
6. Повторяем шаги 3-5, пока это возможно или пока это позволяет критерий остановки обучения.

Решающие деревья

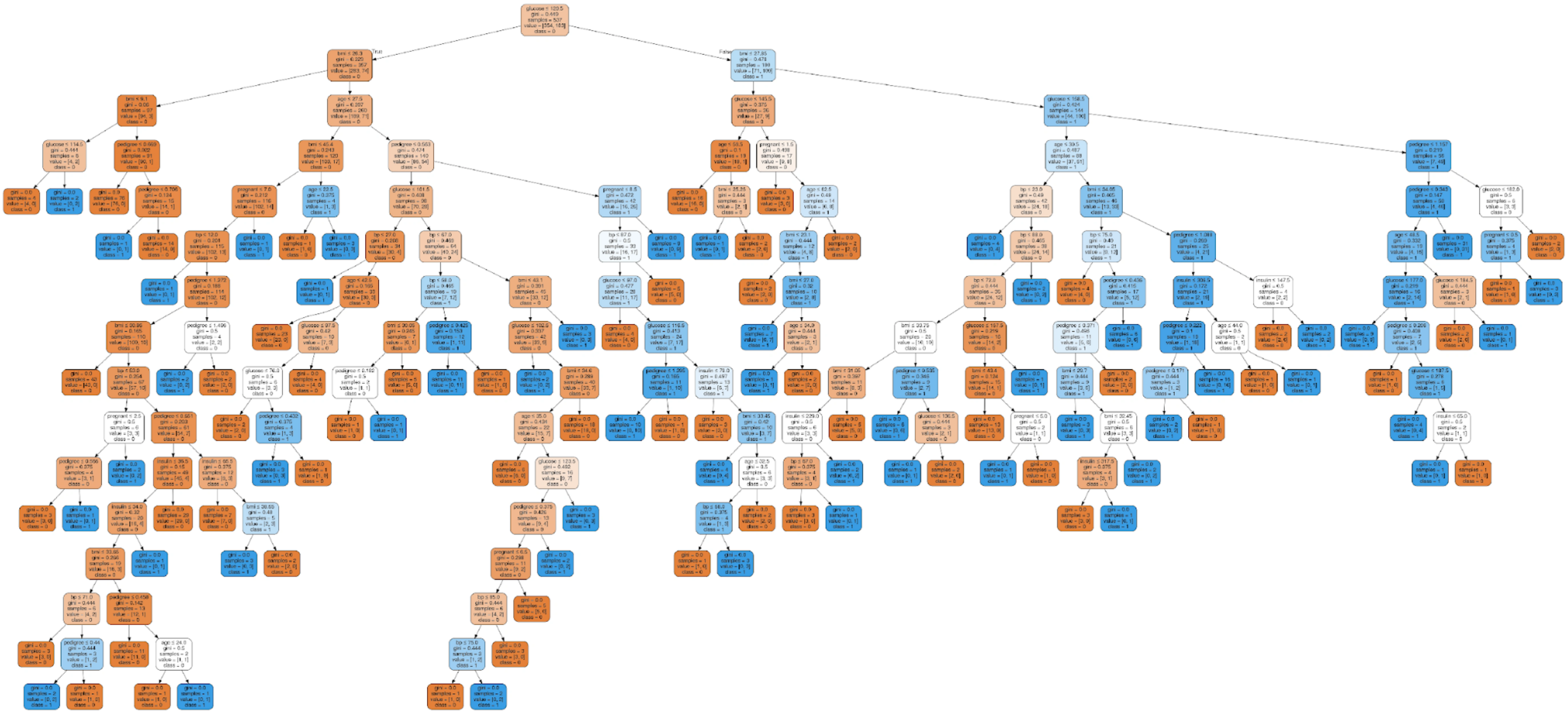
- Теперь, когда идея всего этого стала для нас прорисовываться гораздо чётче, давайте посмотрим на несколько примеров и иллюстраций.

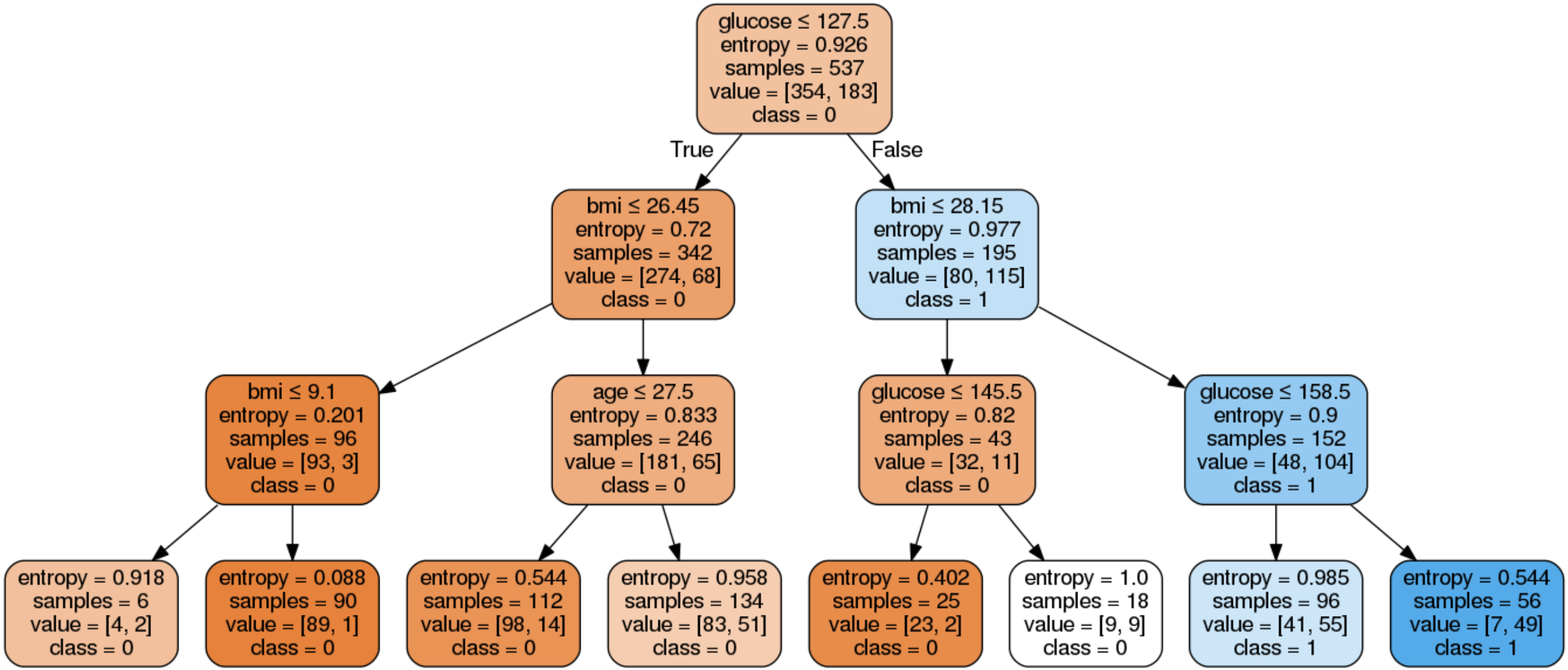
Решающие деревья

- Теперь, когда идея всего этого стала для нас прорисовываться гораздо чётче, давайте посмотрим на несколько примеров и иллюстраций.



Сравнение работы неглубокого решающего дерева и kNN (k = 7) в задаче 3-классовой классификации. Такие графики называются decision surface plots.


glucose ≤ 120.5
gini = 0.449
samples = 537
value = [354, 183]
class = 0
True
False
bmi ≤ 27.85
gini = 0.478
samples = 180
value = [71, 109]
class = 1


glucose ≤ 127.5
entropy = 0.926
samples = 537
value = [354, 183]
class = 0
True
False
bmi ≤ 26.45
entropy = 0.72
samples = 342
value = [274, 68]
class = 0
bmi ≤ 28.15
entropy = 0.977
samples = 195
value = [80, 115]
class = 1
bmi ≤ 9.1
entropy = 0.201
samples = 96
value = [93, 3]
class = 0
age ≤ 27.5
entropy = 0.833
samples = 246
value = [181, 65]
class = 0
glucose ≤ 145.5
entropy = 0.82
samples = 43
value = [32, 11]
class = 0
glucose ≤ 158.5
entropy = 0.9
samples = 152
value = [48, 104]
class = 1
entropy = 0.918
samples = 6
value = [4, 2]
class = 0
entropy = 0.088
samples = 90
value = [89, 1]
class = 0
entropy = 0.544
samples = 112
value = [98, 14]
class = 0
entropy = 0.958
samples = 134
value = [83, 51]
class = 0
entropy = 0.402
samples = 25
value = [23, 2]
class = 0
entropy = 1.0
samples = 18
value = [9, 9]
class = 0
entropy = 0.985
samples = 96
value = [41, 55]
class = 1
entropy = 0.544
samples = 56
value = [7, 49]
class = 1

Решающие деревья. Плюсы

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!
- Не нужно предобрабатывать признаки.

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!
- Не нужно предобрабатывать признаки.
  - Для разбиения не важен масштаб признаков и их природа!

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!
- Не нужно предобрабатывать признаки.
  - Для разбиения не важен масштаб признаков и их природа!
- Очень легко интерпретировать предсказания.

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!
- Не нужно предобрабатывать признаки.
  - Для разбиения не важен масштаб признаков и их природа!
- Очень легко интерпретировать предсказания.
  - Для нового объекта можно отследить, в какой лист он попадёт, и проанализировать, почему так.

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!
- Не нужно предобрабатывать признаки.
  - Для разбиения не важен масштаб признаков и их природа!
- Очень легко интерпретировать предсказания.
  - Для нового объекта можно отследить, в какой лист он попадёт, и проанализировать, почему так.
- Сложность предсказания — в худшем случае $O(d)$, где d — глубина.

Решающие деревья. Плюсы

- Один из самых простых алгоритмов.
  - Надёжный как швейцарские часы!
- Не нужно предобрабатывать признаки.
  - Для разбиения не важен масштаб признаков и их природа!
- Очень легко интерпретировать предсказания.
  - Для нового объекта можно отследить, в какой лист он попадёт, и проанализировать, почему так.
- Сложность предсказания — в худшем случае $O(d)$, где d — глубина.
  - Более того, для того чтобы получить ответ на объекте, понадобится не более чем d шагов в дереве.

Решающие деревья. Минусы

Решающие деревья. Минусы

- Трудно подобрать гиперпараметры.

Решающие деревья. Минусы

- Трудно подобрать гиперпараметры.
  - Если глубина слишком маленькая, то модель недообучается. Если глубина слишком большая, то модель переобучается.
  - С другими гиперпараметрами — ситуация аналогичная. Найти золотую середину нелегко.

Решающие деревья. Минусы

- Трудно подобрать гиперпараметры.
  - Если глубина слишком маленькая, то модель недообучается. Если глубина слишком большая, то модель переобучается.
  - С другими гиперпараметрами — ситуация аналогичная. Найти золотую середину нелегко.
- Небольшие вариации (или расхождения) в данных могут привести уже к другому дереву решений.

Решающие деревья. Минусы

- Трудно подобрать гиперпараметры.
  - Если глубина слишком маленькая, то модель недообучается. Если глубина слишком большая, то модель переобучается.
  - С другими гиперпараметрами — ситуация аналогичная. Найти золотую середину нелегко.
- Небольшие вариации (или расхождения) в данных могут привести уже к другому дереву решений.
  - Частично этот эффект можно уменьшить с помощью алгоритмов ансамблей: бэггинга и бустинга.

Решающие деревья. Области применения

- Банковское дело.
  - Оценка кредитоспособности клиентов банка при выдаче кредитов.
- Промышленность.
  - Контроль за качеством продукции (выявление дефектов), испытания без разрушений (например, проверка качества сварки) и т.д.
- Медицина.
  - Диагностика заболеваний.
- Молекулярная биология.
  - Анализ строения аминокислот.

Решающие деревья. Области применения

- Банковское дело.
  - Оценка кредитоспособности клиентов банка при выдаче кредитов.
- Промышленность.
  - Контроль за качеством продукции (выявление дефектов), испытания без разрушений (например, проверка качества сварки) и т.д.
- Медицина.
  - Диагностика заболеваний.
- Молекулярная биология.
  - Анализ строения аминокислот.

И многие другие области, где требуются быстрые и интерпретируемые алгоритмы!

Ансамблевые методы

Ансамблевые методы

- Как вы понимаете, что такое ансамбли в машинном обучении?

Ансамблевые методы

- Как вы понимаете, что такое ансамбли в машинном обучении?
- Ансамблевые методы — это парадигма машинного обучения, где несколько моделей (часто называемых «слабыми учениками») обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов.

Ансамблевые методы

- Как вы понимаете, что такое ансамбли в машинном обучении?
- Ансамблевые методы — это парадигма машинного обучения, где несколько моделей (часто называемых «слабыми учениками») обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов.
- Основная гипотеза и предпосылка состоит в том, что при правильном сочетании слабых моделей мы можем получить более точные и/или надежные модели.

Ансамблевые методы

Главная идея ансамблевых методов — достаточно много слабых алгоритмов вместе образуют сильный.

Ансамблевые методы

- Для того чтобы разобраться подробно с интуицией, которая скрывается под капотом классических ансамблевых методов, нам прежде необходимо обсудить одну важную теоретическую концепцию.
- Для начала давайте разберемся со следующим вопросом: на какие компоненты (составные части) можно разложить ошибку любого предсказания модели машинного обучения?

Ансамблевые методы

- Для того чтобы разобраться подробно с интуицией, которая скрывается под капотом классических ансамблевых методов, нам прежде необходимо обсудить одну важную теоретическую концепцию.
- Для начала давайте разберемся со следующим вопросом: на какие компоненты (составные части) можно разложить ошибку любого предсказания модели машинного обучения?
- Данные компоненты называются: bias, variance и noise.

Bias-Variance Tradeoff

- Теоретическая концепция, которую мы упоминали ранее и которая лежит в основе многих ансамблевых методов, называется Bias-Variance Tradeoff — или по-русски (можно попробовать перевести дословно) — «торг между смещением и дисперсией».

Bias-Variance Tradeoff

- Теоретическая концепция, которую мы упоминали ранее и которая лежит в основе многих ансамблевых методов, называется Bias-Variance Tradeoff — или по-русски (можно попробовать перевести дословно) — «торг между смещением и дисперсией».
- Для понимания данной концепции давайте вернемся к вопросу, который мы обсуждали на прошлом слайде, и посмотрим, как можно записать представление ошибки предсказания модели машинного обучения формульно:

$$\mathbb{E}(y - \hat{y})^2 = (y - \mathbb{E}\hat{y})^2 + \mathbb{D}\hat{y} + \sigma^2$$

Bias-Variance Tradeoff

- Теоретическая концепция, которую мы упоминали ранее и которая лежит в основе многих ансамблевых методов, называется Bias-Variance Tradeoff — или по-русски (можно попробовать перевести дословно) — «торг между смещением и дисперсией».
- Для понимания данной концепции давайте вернемся к вопросу, который мы обсуждали на прошлом слайде, и посмотрим, как можно записать представление ошибки предсказания модели машинного обучения формульно:

Прокомментируйте все обозначения!

$$\mathbb{E}(y - \hat{y})^2 = (y - \mathbb{E}\hat{y})^2 + \mathbb{D}\hat{y} + \sigma^2$$

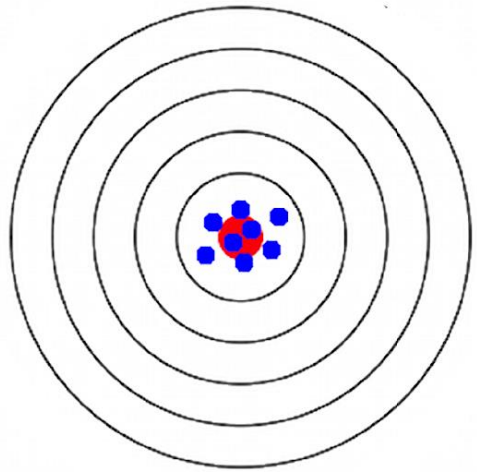
Bias-Variance Tradeoff

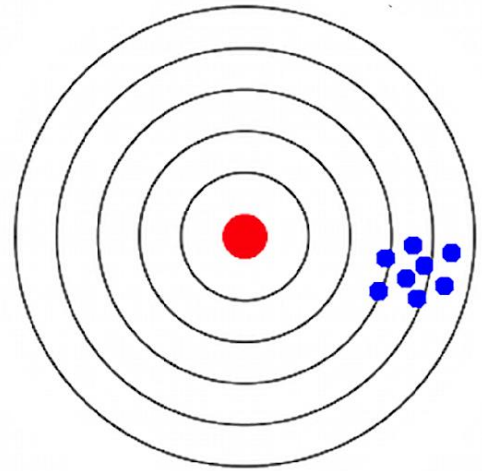
$$\mathbb{E}(y - \hat{y})^2 = (y - \mathbb{E}\hat{y})^2 + \mathbb{D}\hat{y} + \sigma^2$$

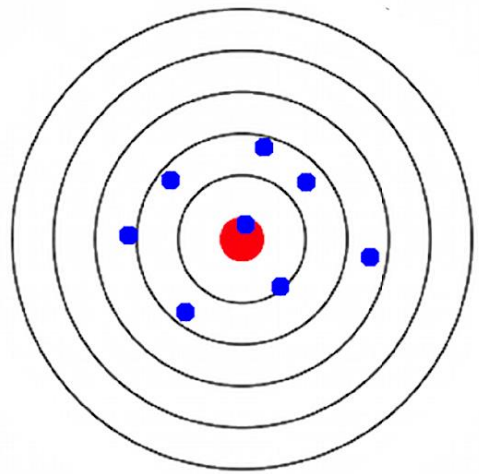
Bias-Variance Tradeoff

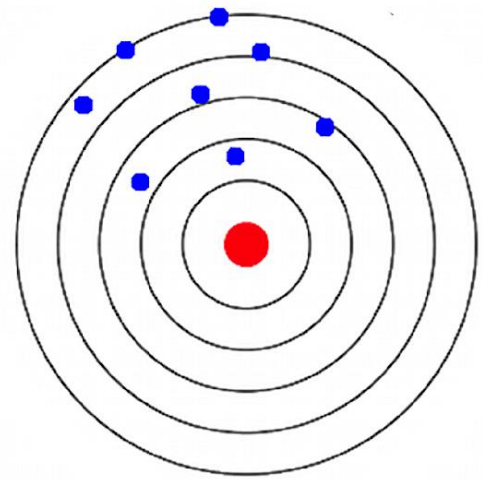
$$\underset{Expected\ error}{\mathbb{E}(y - \hat{y})^2} = \underset{Bias^2}{(y - \mathbb{E}\hat{y})^2} + \underset{Variance}{\mathbb{D}\hat{y}} + \underset{Noise}{\sigma^2}$$

- Здесь:
  - $\hat{y}$ — предсказание алгоритма $\hat{y} = f(X)$.
  - Разбросом ($Variance$) мы назвали дисперсию ответов алгоритма.
  - Смещением ($Bias$) — матожидание разности между истинным ответом и ответом, выданным алгоритмом.

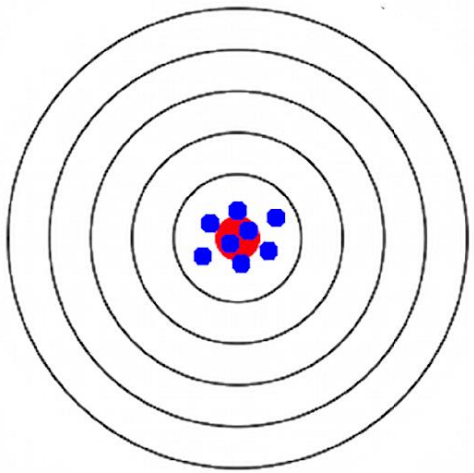


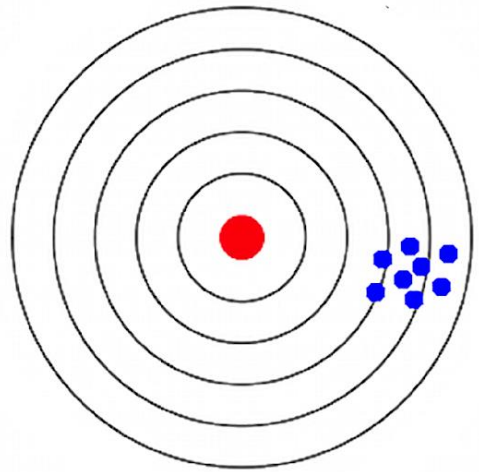


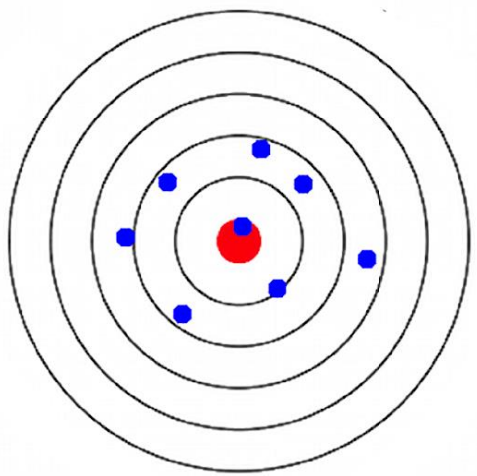


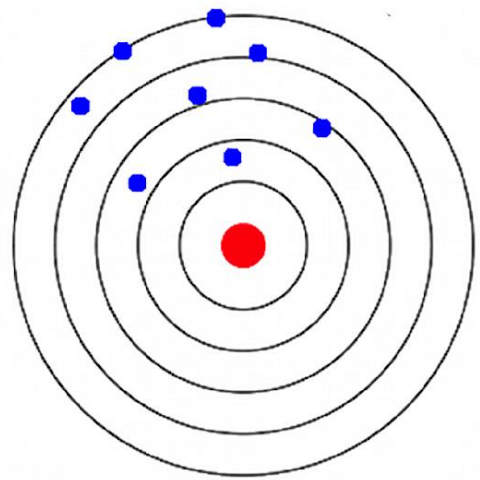


В какой из представленных ситуаций высокий Bias, но низкий Variance; в какой — низкий Bias и высокий Variance и т.д.?





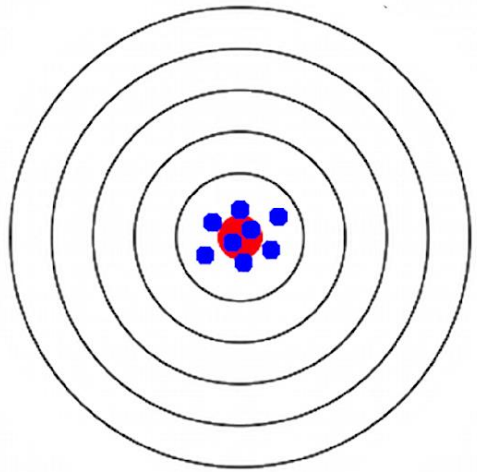


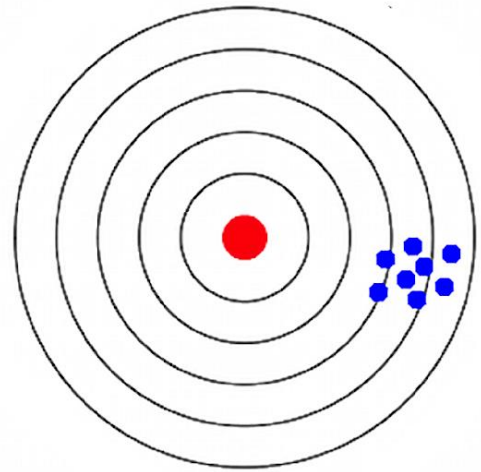


Low bias

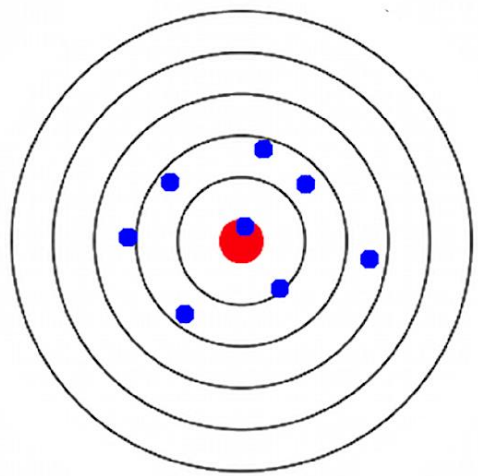
High bias

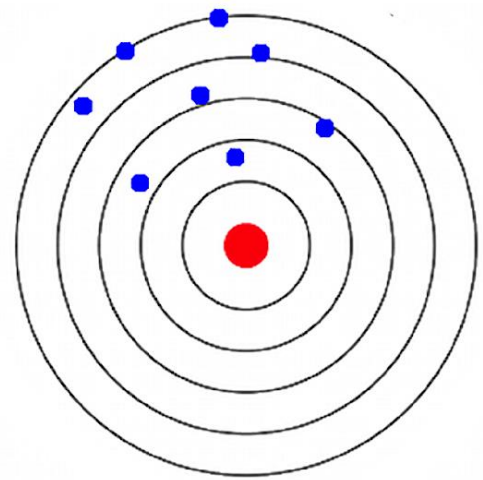
Low variance





High variance



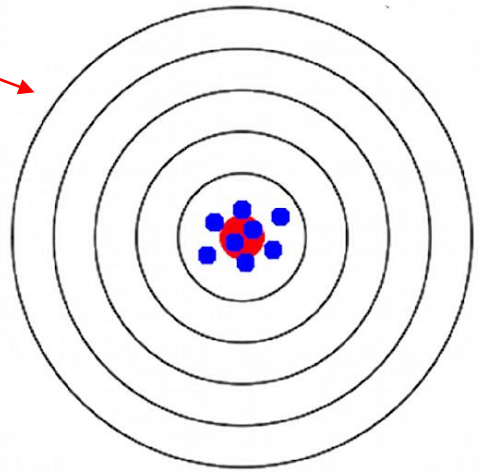


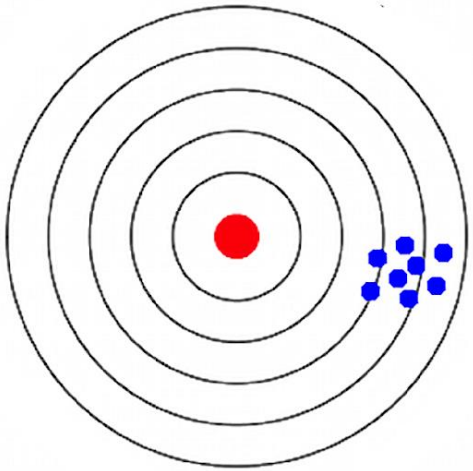
Low bias

High bias

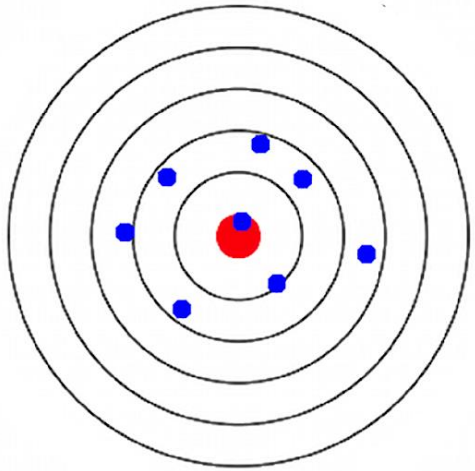
Идеальный алгоритм:

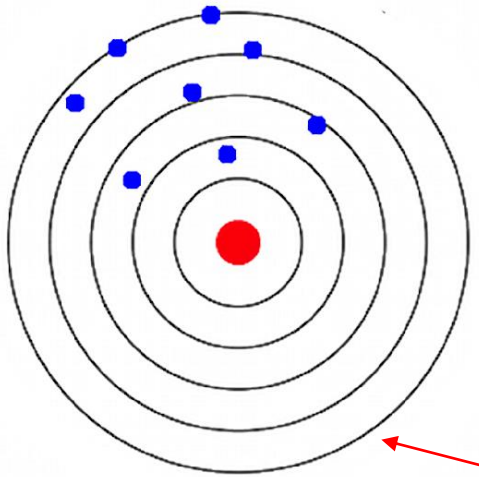
Low variance



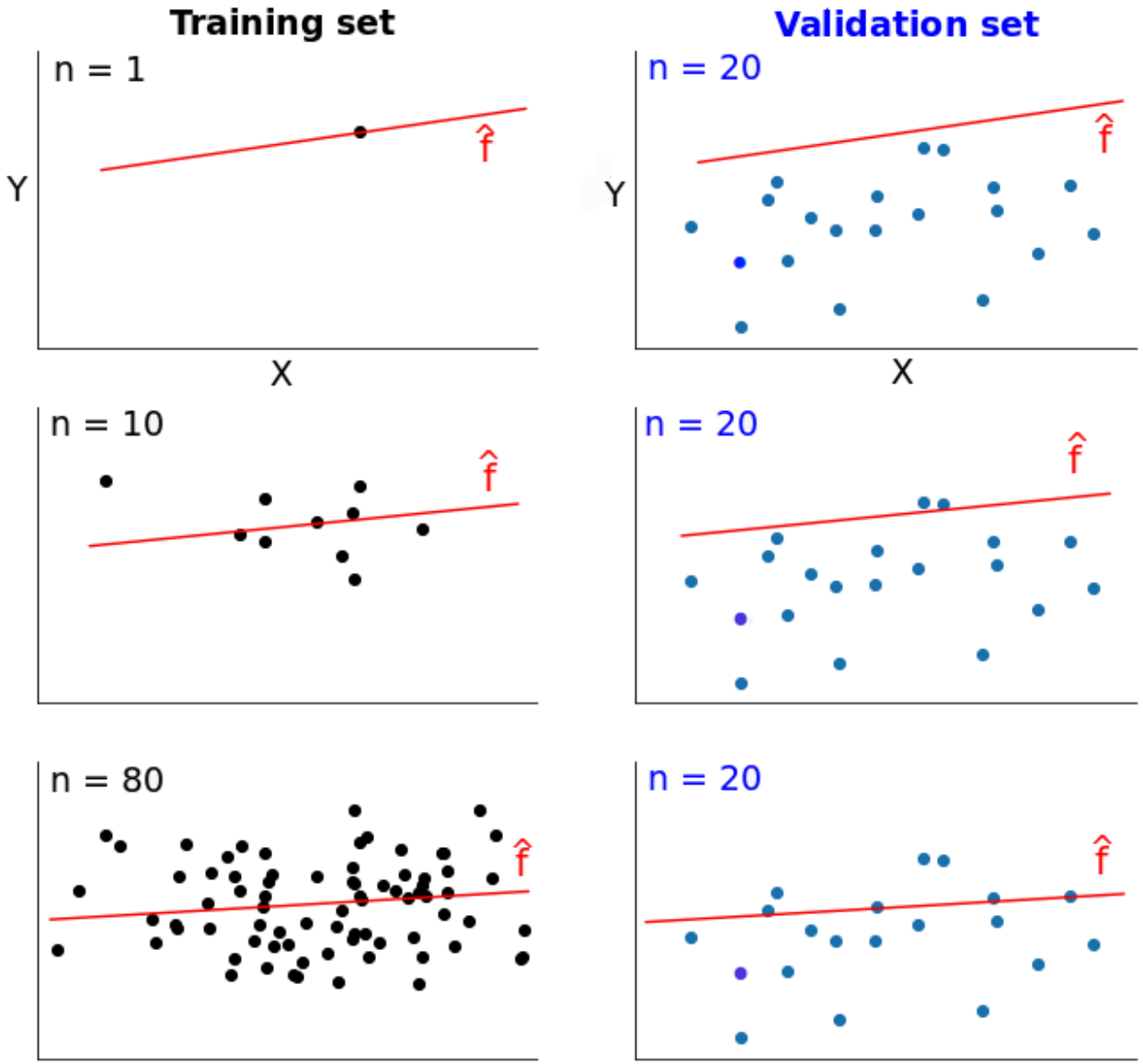


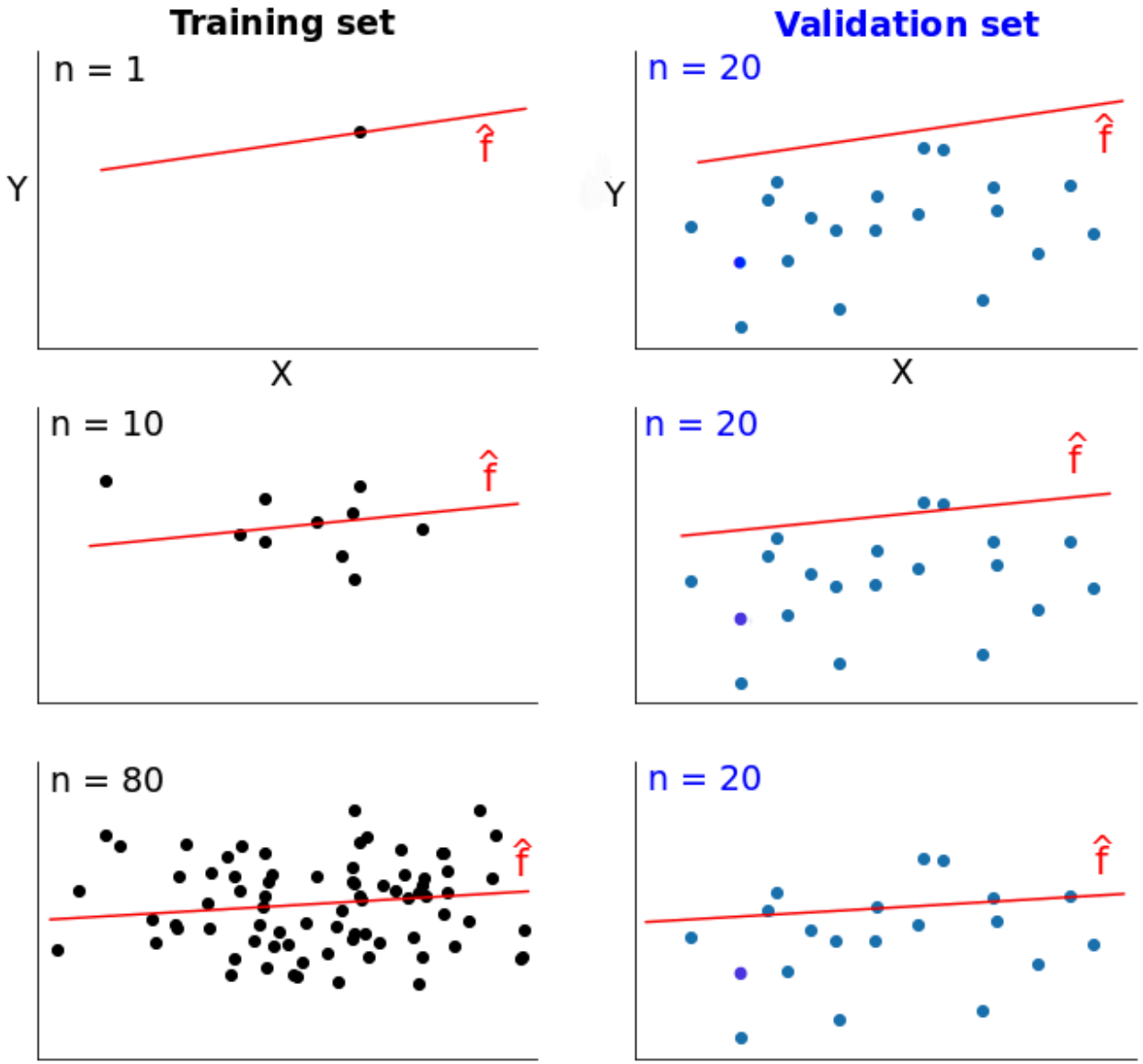
High variance



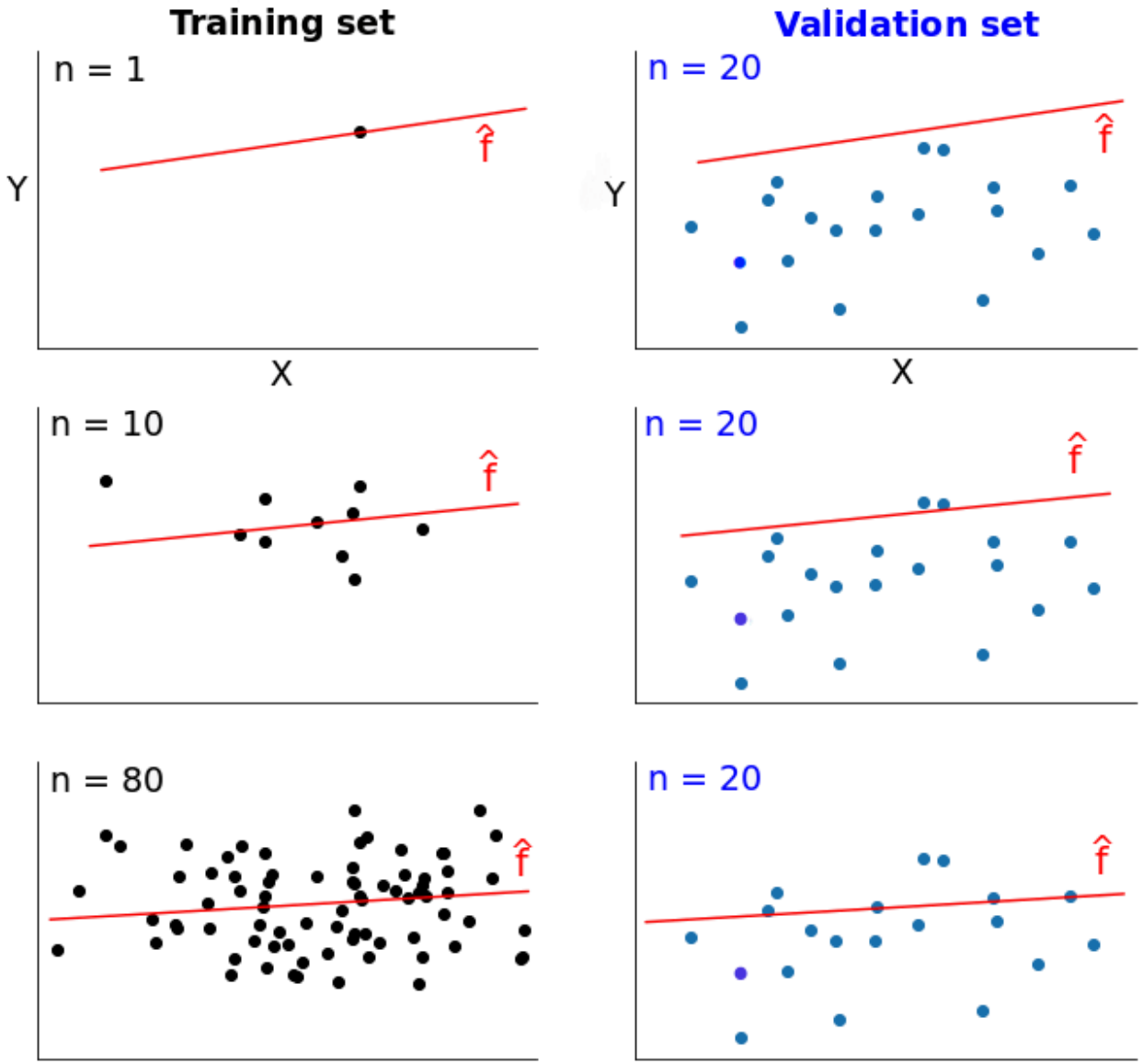


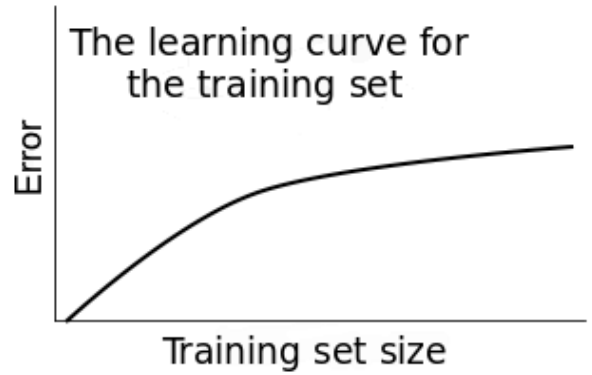
Очень плохой алгоритм:

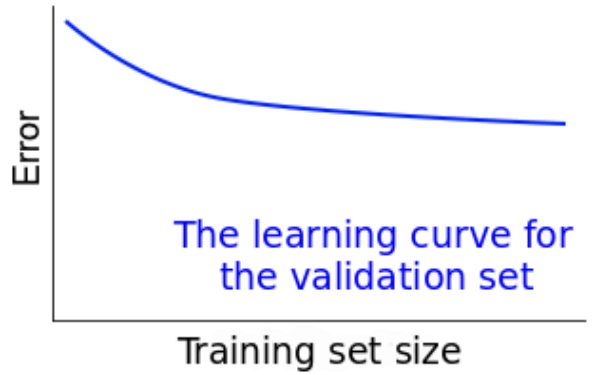

Training set
Validation set
n = 1
n = 20
Y
X
$\hat{f}$
n = 10
n = 20
n = 80
n = 20

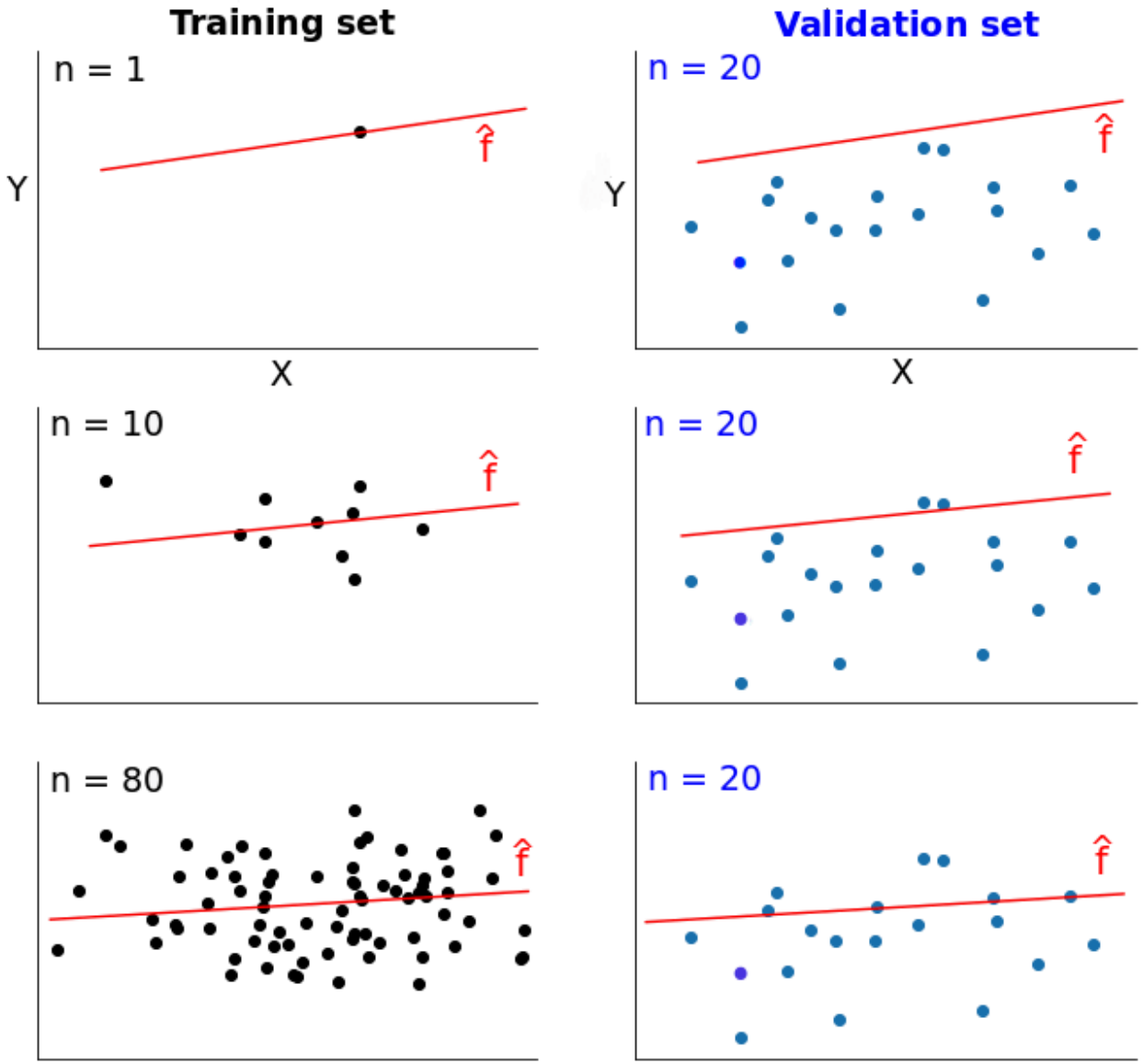


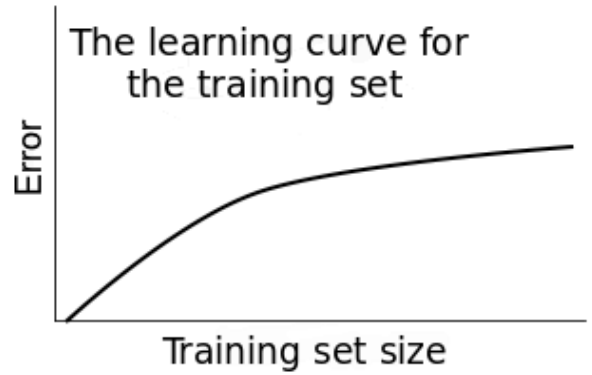
Другой пример. Ответьте на вопрос:
Как будет меняться ошибка на тренировочном и валидационном наборе?

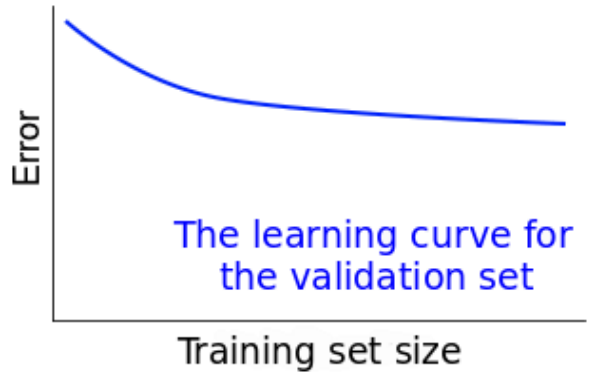

Training set
Validation set
n = 1
n = 20
Y
X
$\hat{f}$
n = 10
n = 20
n = 80
n = 20

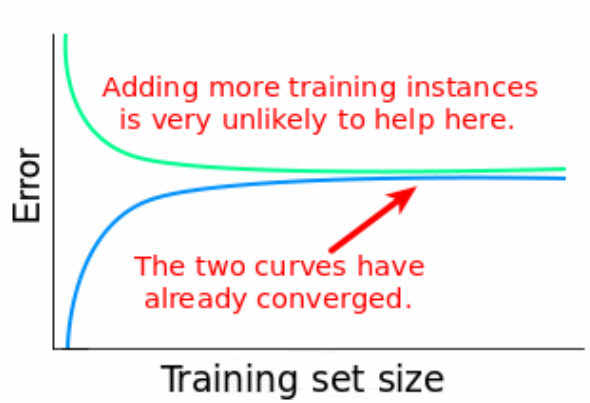

The learning curve for
the training set
Error
Training set size

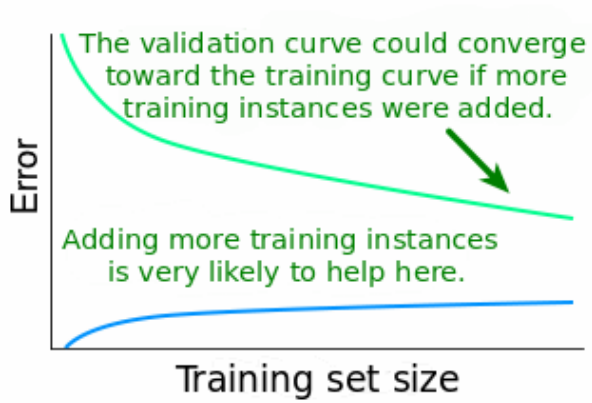

Error
The learning curve for
the validation set
Training set size

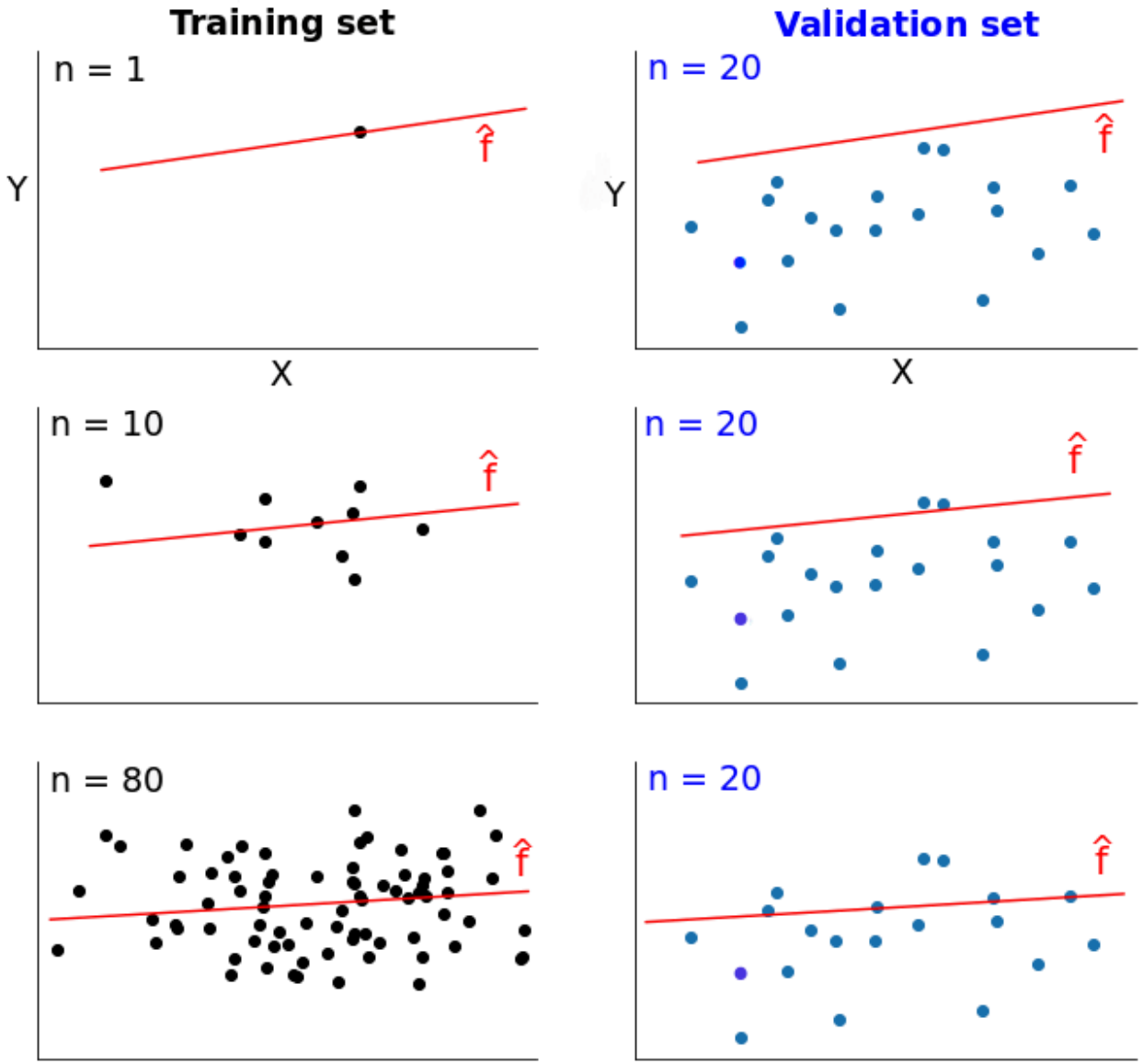

Training set
Validation set
n = 1
n = 20
Y
X
$\hat{f}$
n = 10
n = 20
n = 80
n = 20

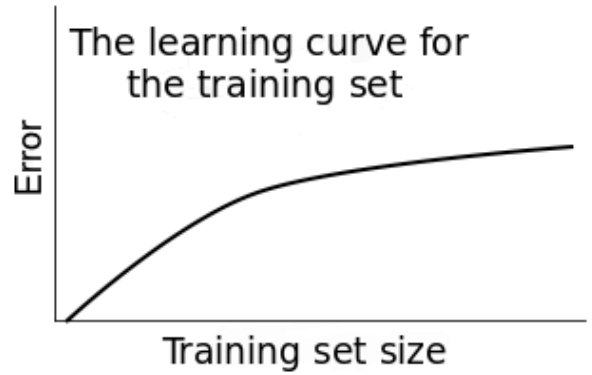

The learning curve for the training set
Error
Training set size

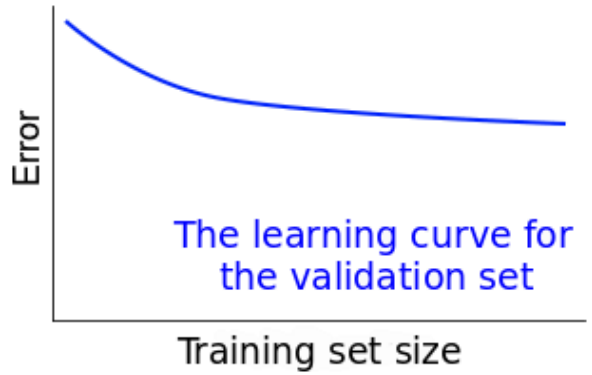

Error
The learning curve for the validation set
Training set size

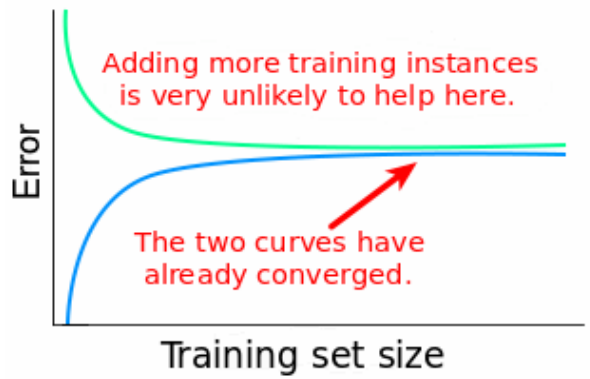

Adding more training instances is very unlikely to help here.
The two curves have already converged.
Error
Training set size

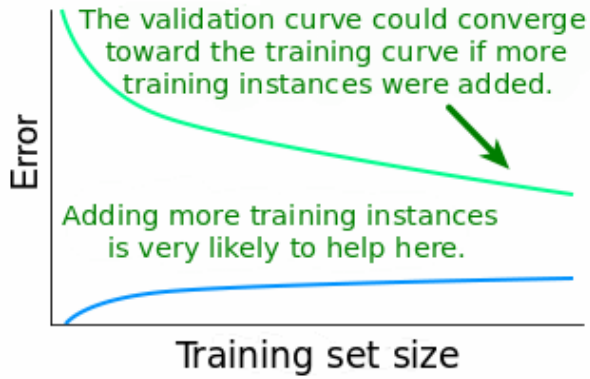

The validation curve could converge toward the training curve if more training instances were added.
Adding more training instances is very likely to help here.
Error
Training set size



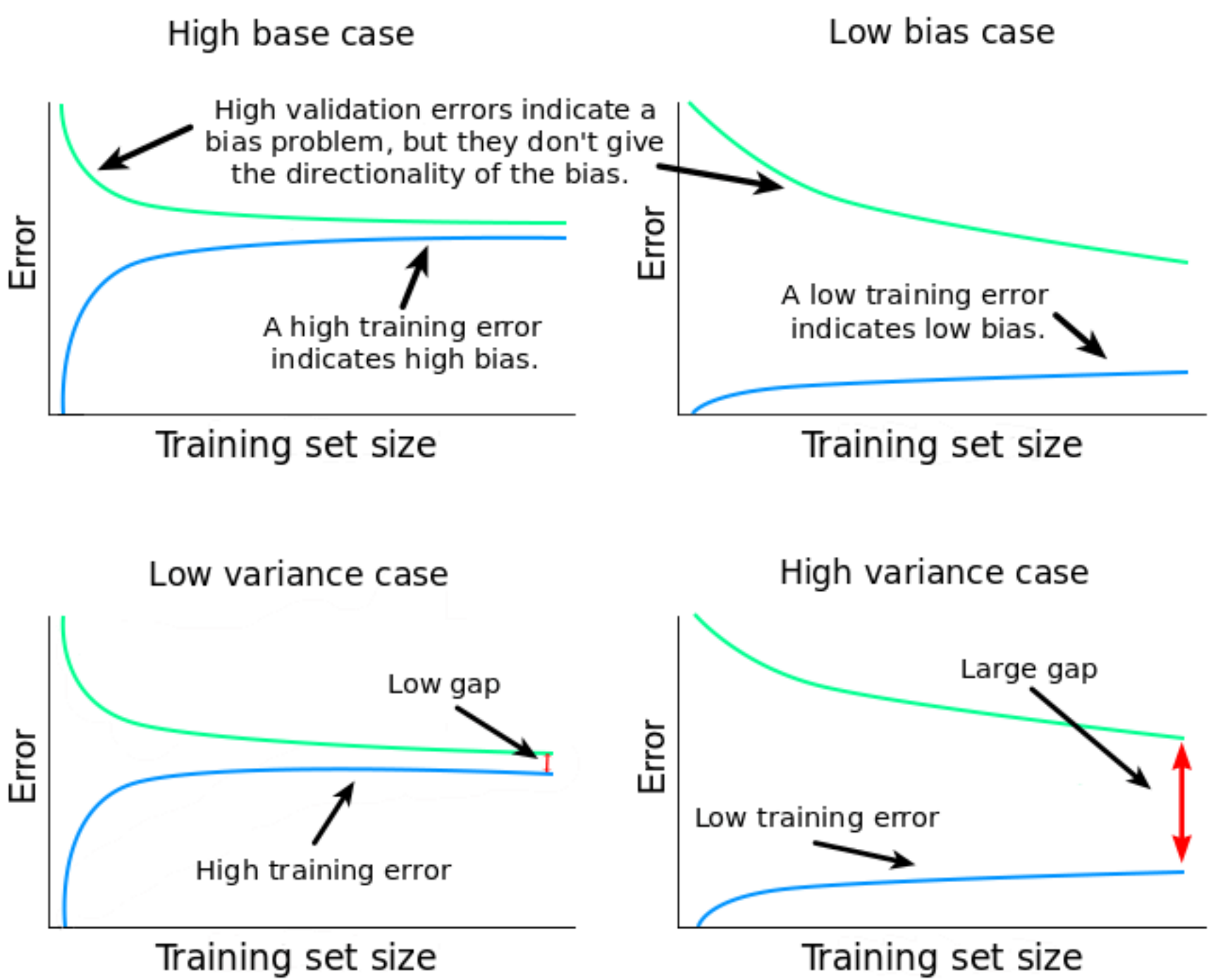


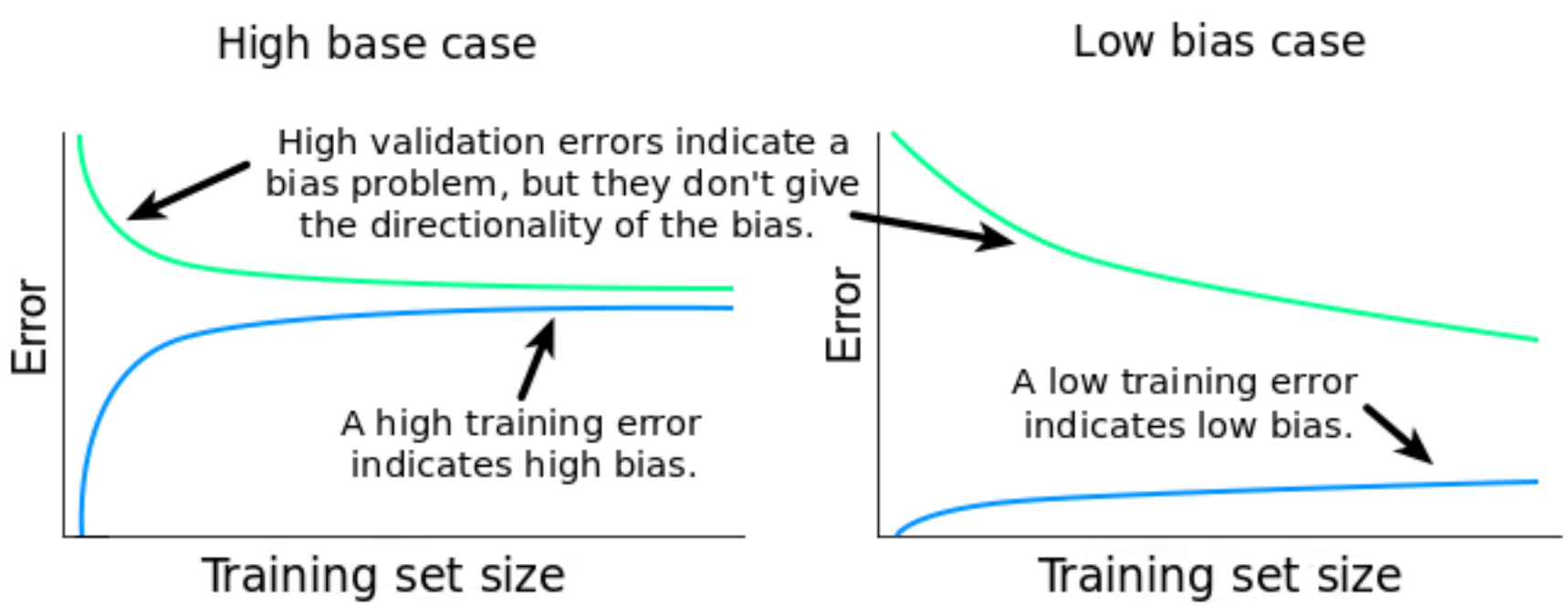




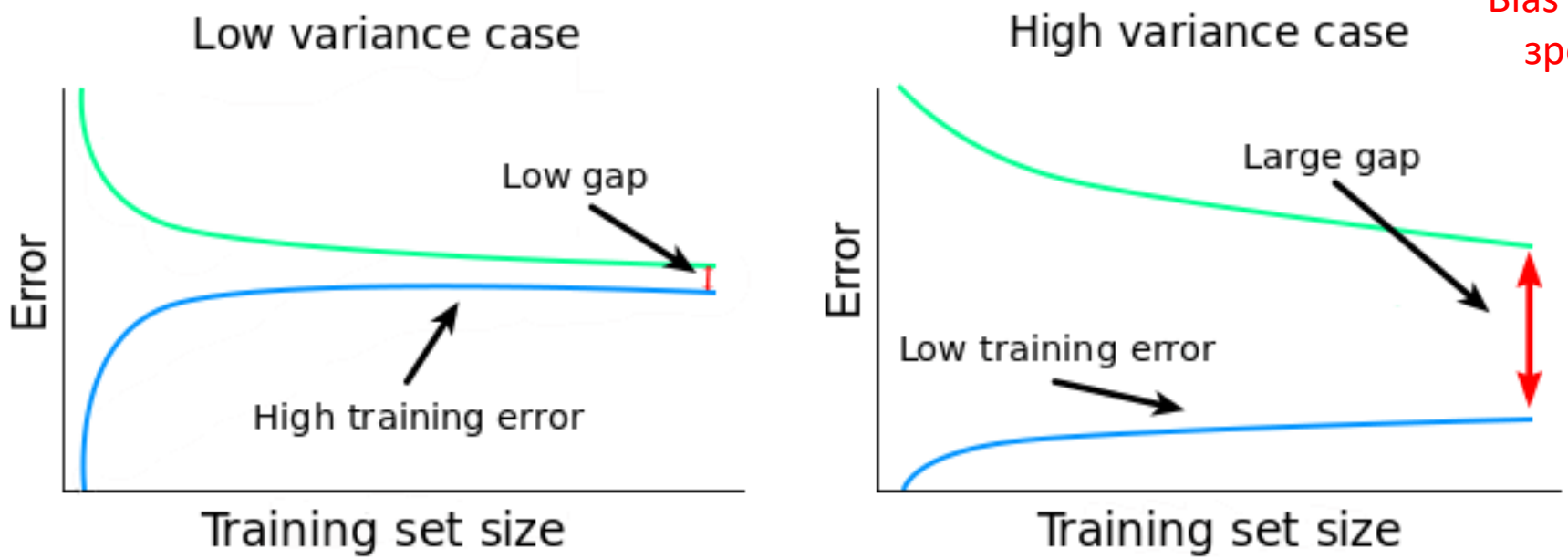


Что за ситуации иллюстрируются примерами выше?

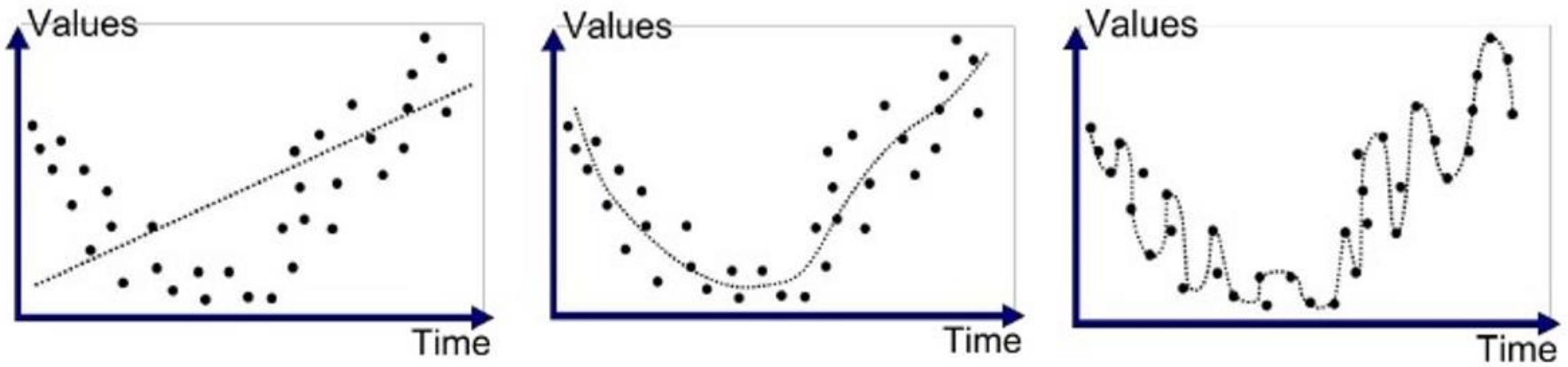

High base case
High validation errors indicate a bias problem, but they don't give the directionality of the bias.
Error
A high training error indicates high bias.
Training set size
Low bias case
Error
A low training error indicates low bias.
Training set size
Low variance case
Low gap
Error
High training error
Training set size
High variance case
Large gap
Error
Low training error
Training set size



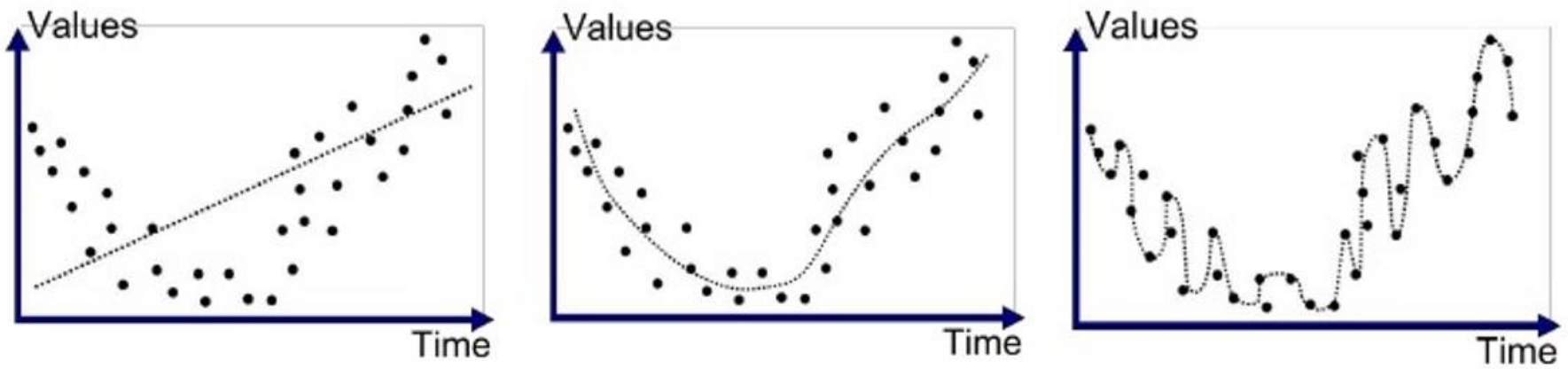
Резюмируя: что показывают Bias и Variance с точки зрения обучения?



Bias-Variance Tradeoff

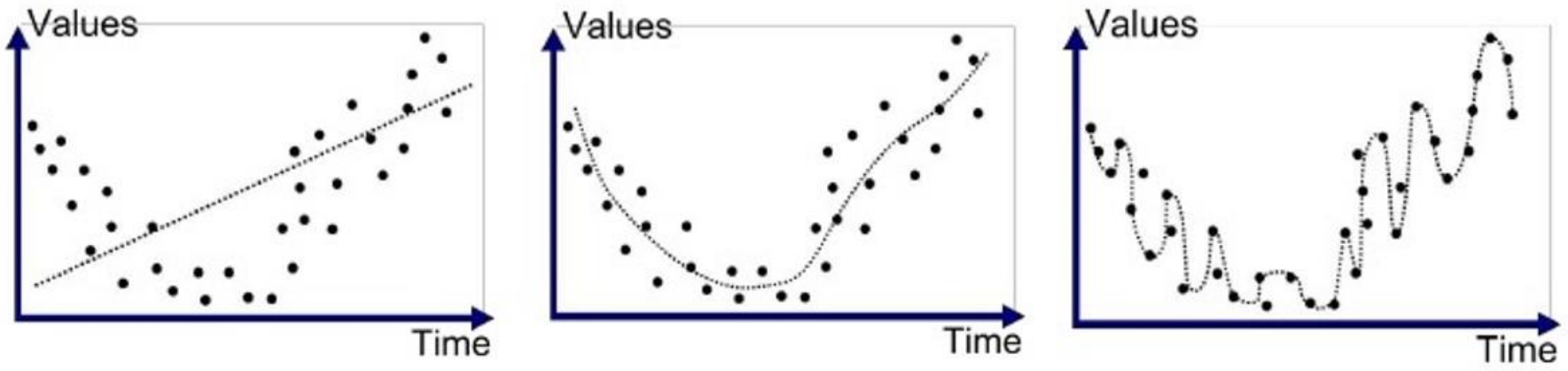


Bias-Variance Tradeoff



На каком графике изображено явное переобучение?

Bias-Variance Tradeoff

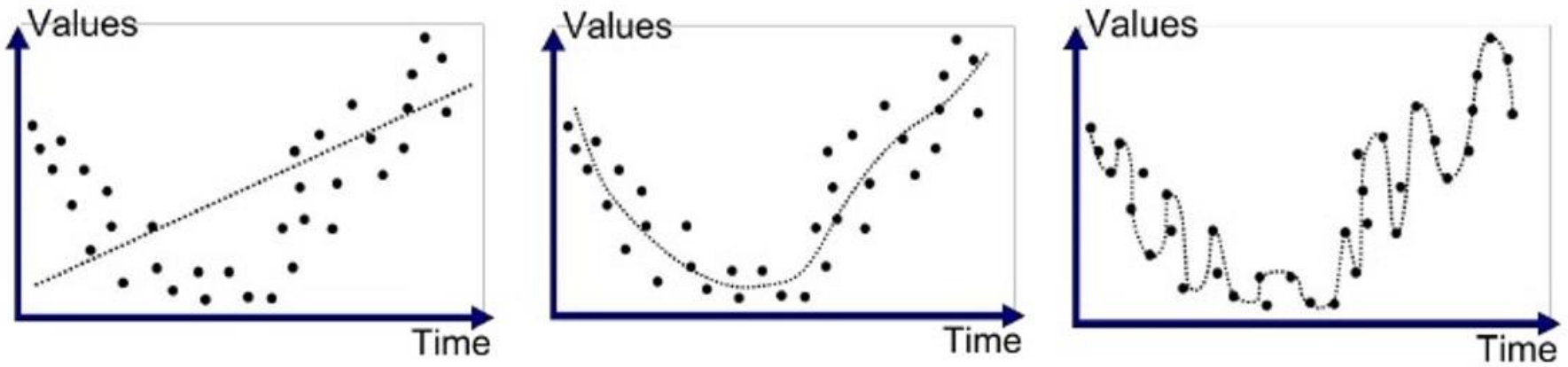


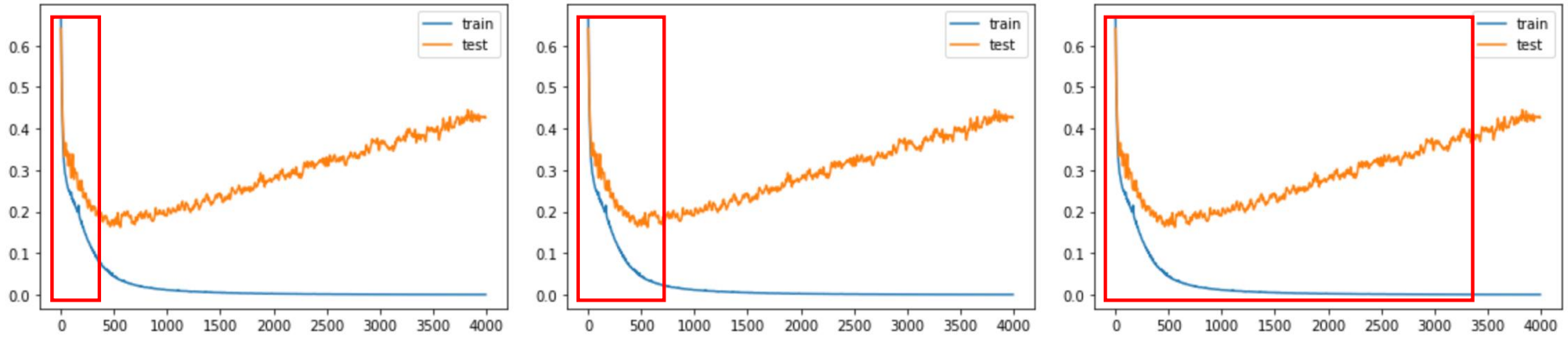
Underfitted

Good Fit/Robust

Overfitted

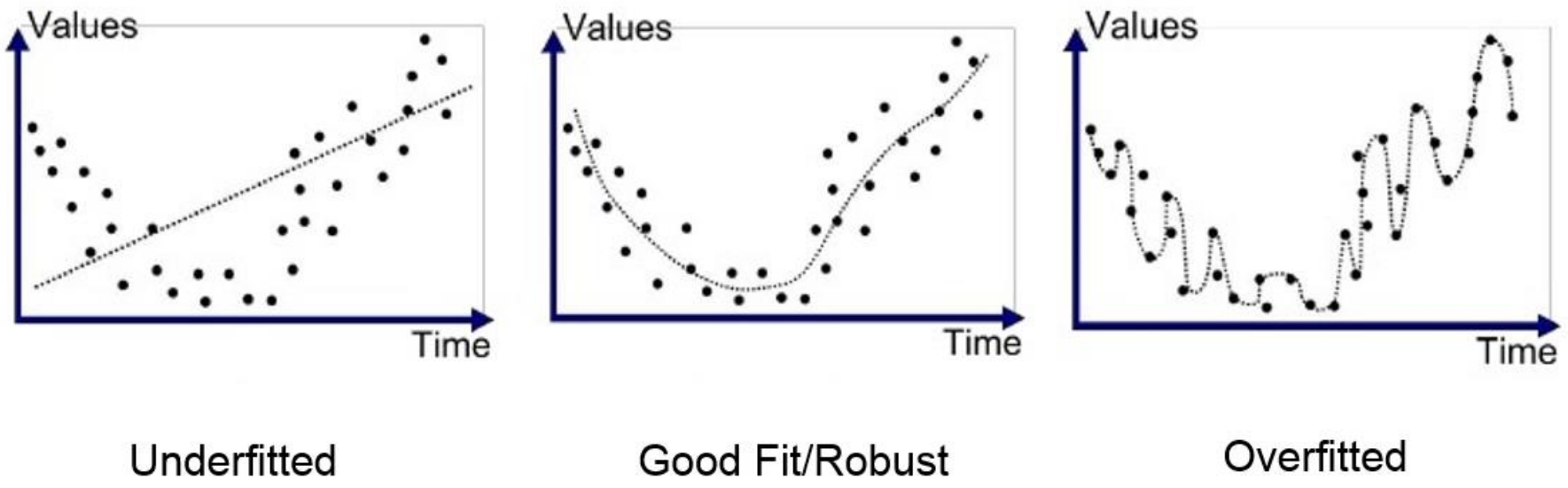
Bias-Variance Tradeoff

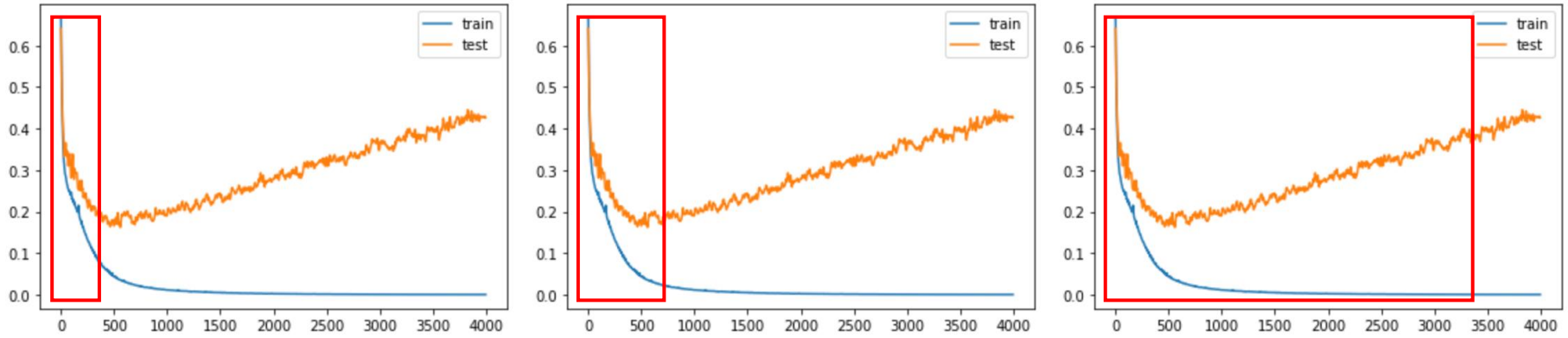


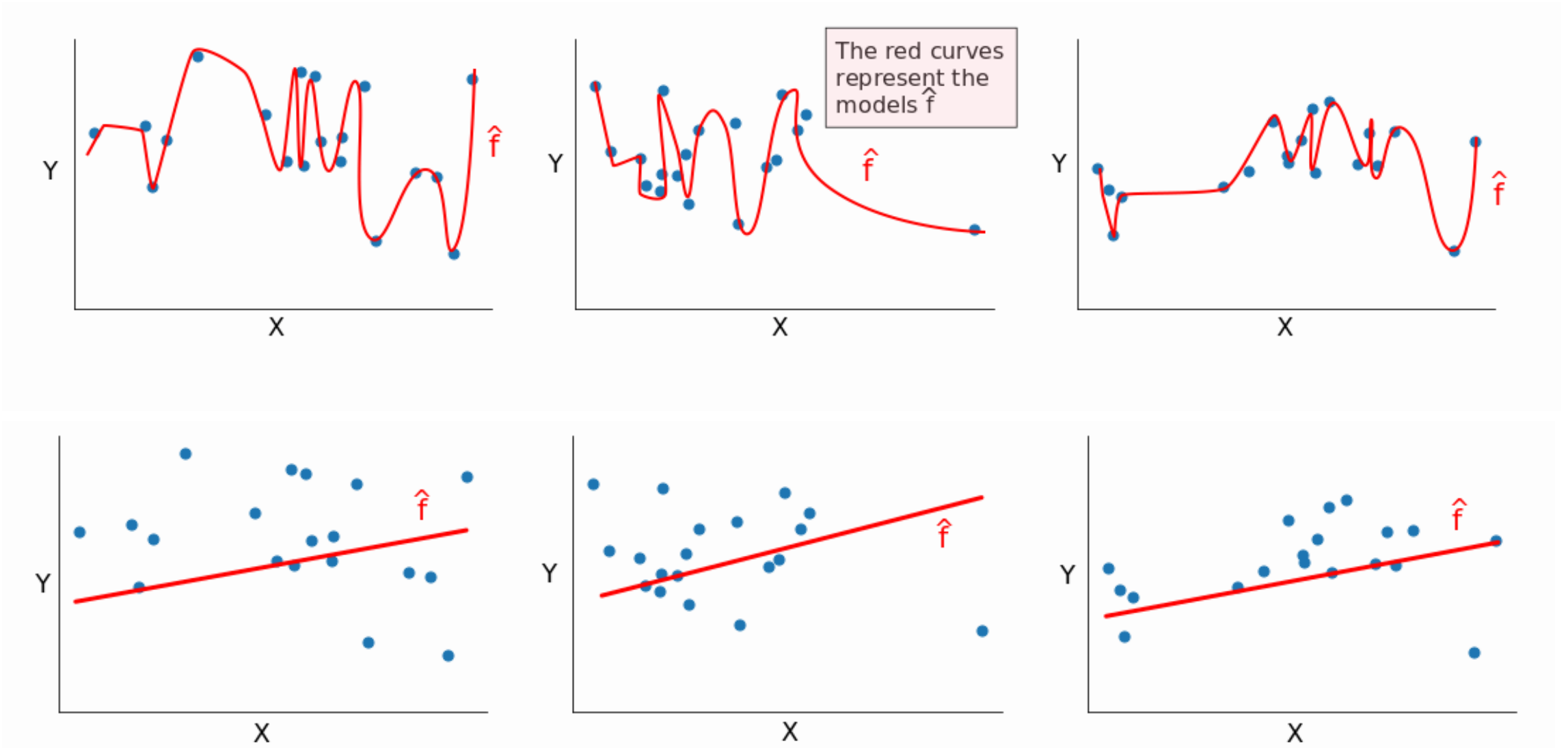


На каком этапе графика изображено явное переобучение?

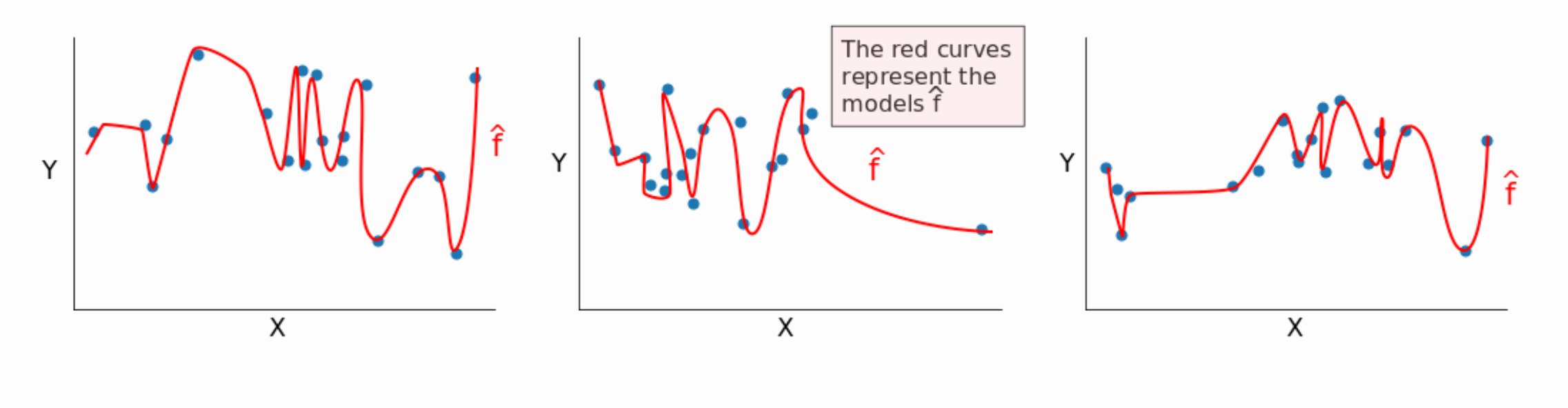
Bias-Variance Tradeoff

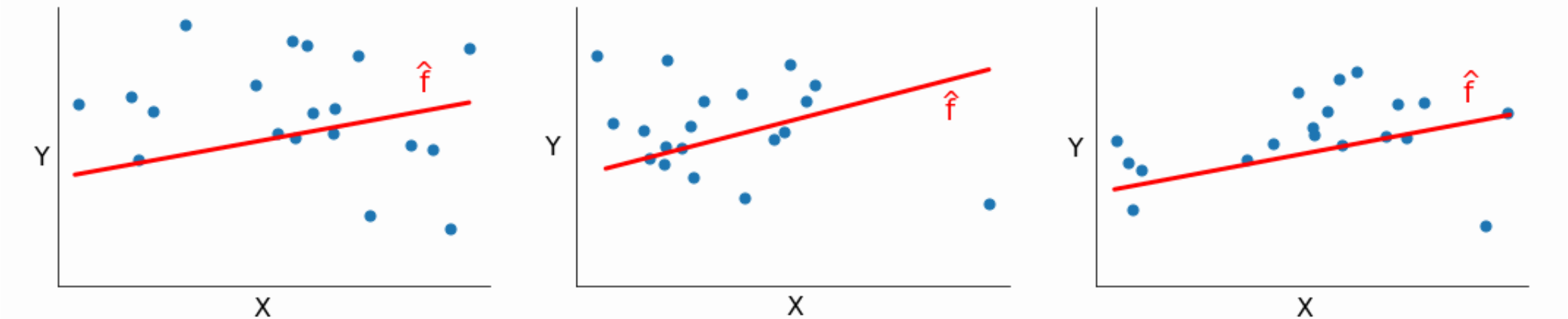


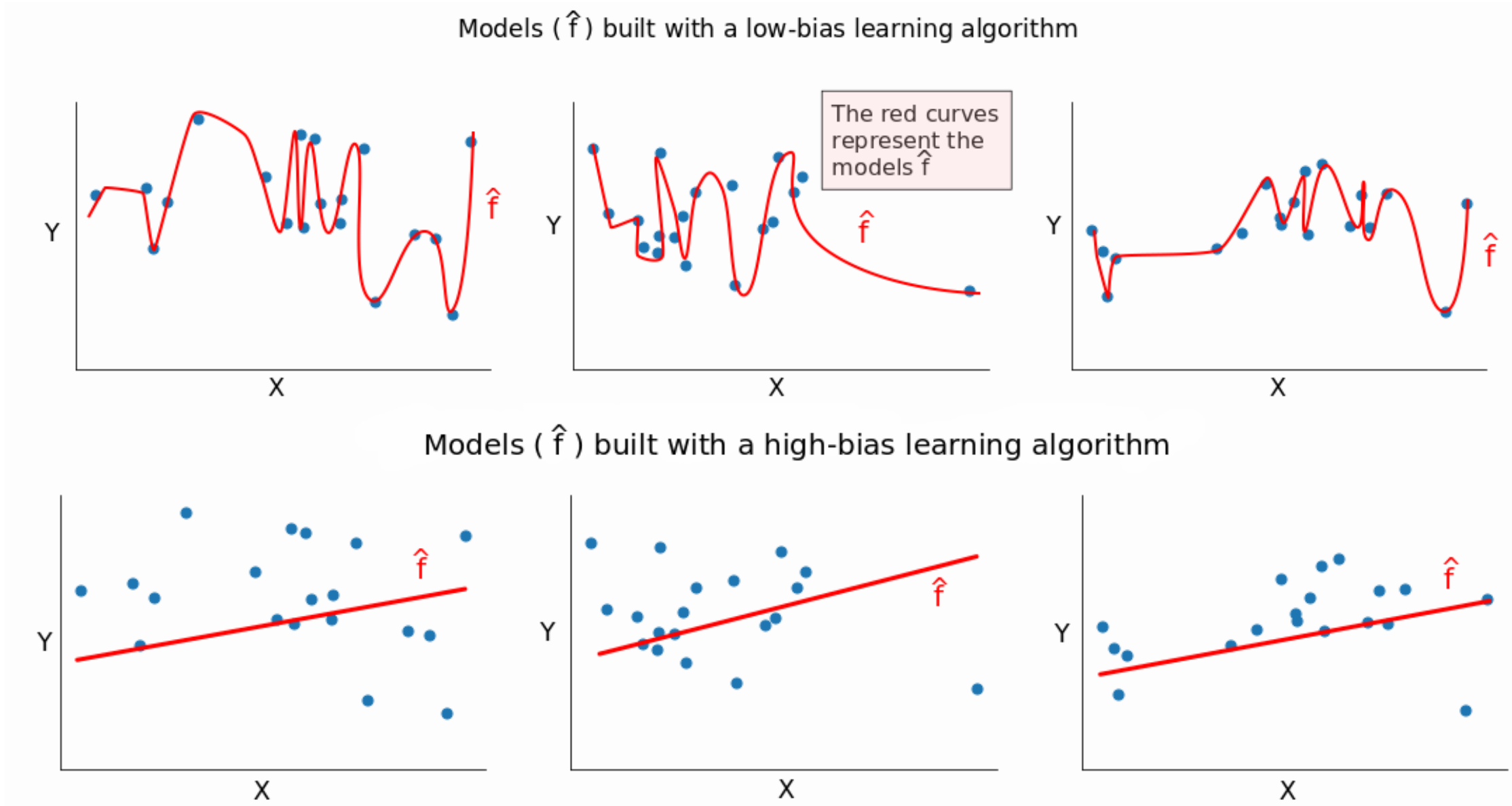



The red curves represent the models $\hat{f}$
Y
X
$\hat{f}$

Вопрос: на каких из графиков у моделей высокий bias (смещение)?






Models ($\hat{f}$) built with a low-bias learning algorithm
Y
X
$\hat{f}$
The red curves represent the models $\hat{f}$
Models ($\hat{f}$) built with a high-bias learning algorithm

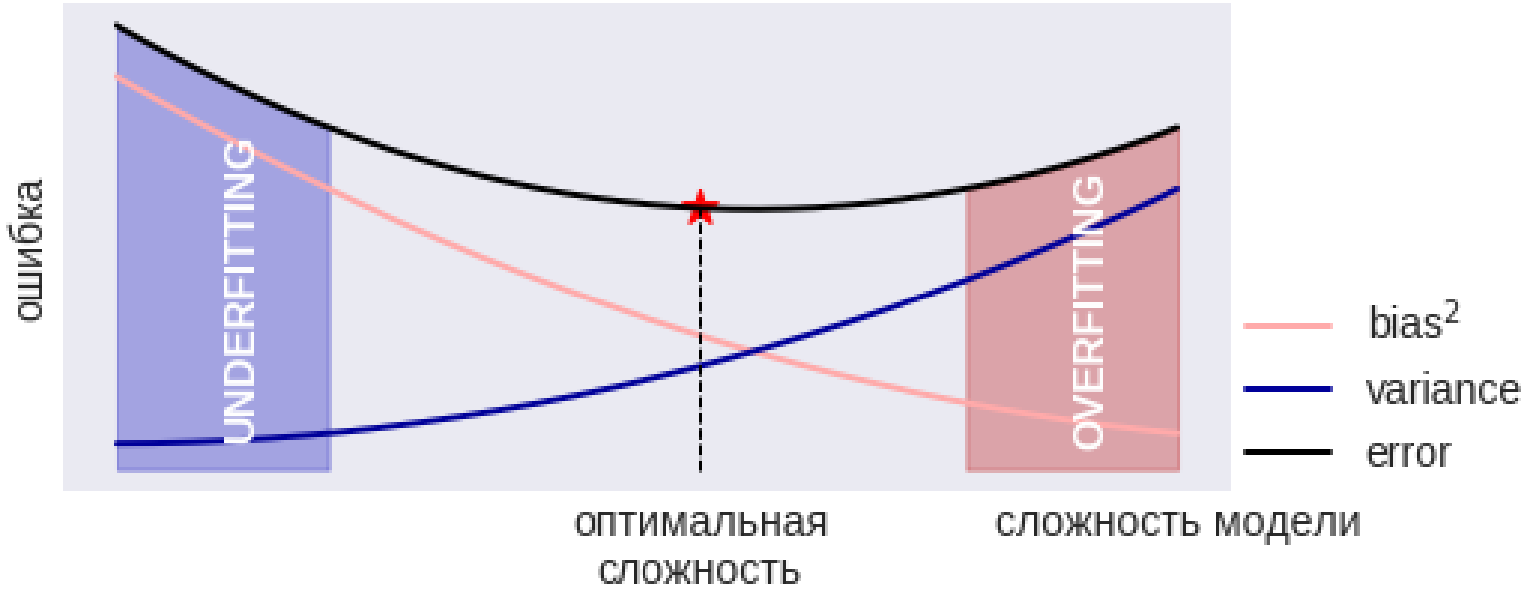
Bias-Variance Tradeoff

- Смещение — это погрешность оценки, возникающая в результате ошибочного предположения в алгоритме обучения.
  - В результате большого смещения алгоритм может пропустить связь между признаками и выводом (образуется недообучение).

- Дисперсия — это ошибка чувствительности к малым отклонениям в тренировочном наборе.
  - При высокой дисперсии алгоритм может начать трактовать случайный шум в тренировочном наборе, вместо того чтобы выявлять зависимости (образуется переобучение).

Bias-Variance Tradeoff

- Ниже представлена классическая иллюстрация изменения разброса и смещения. Именно эта дилемма и называется Bias-Variance Tradeoff.



Ансамблевые методы

- Так вот ансамблевые методы, оказывается, представляют собой всегда композиции над слабыми алгоритмами.
- Слабый алгоритм обладает плохой обобщающей способностью. Есть два основных случая:

Ансамблевые методы

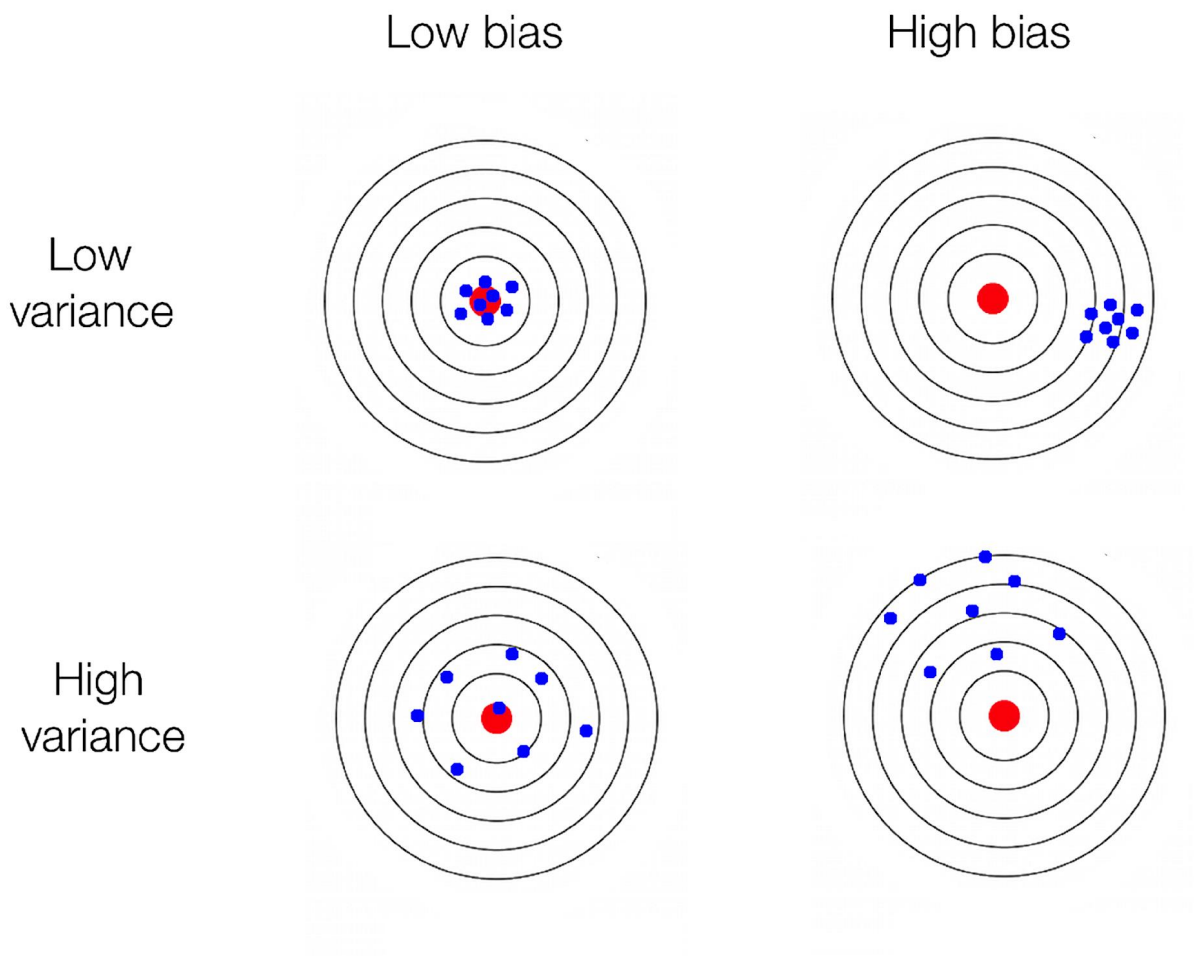
- Так вот ансамблевые методы, оказывается, представляют собой всегда композиции над слабыми алгоритмами.
- Слабый алгоритм обладает плохой обобщающей способностью. Есть два основных случая:
  - Алгоритм слишком примитивен (высокий bias, но низкий variance). Пример — решающий пень.

Ансамблевые методы

- Так вот ансамблевые методы, оказывается, представляют собой всегда композиции над слабыми алгоритмами.
- Слабый алгоритм обладает плохой обобщающей способностью. Есть два основных случая:
  - Алгоритм слишком примитивен (высокий bias, но низкий variance). Пример — решающий пень.
  - Алгоритм слишком легко переобучается (низкий bias, но высокий variance). Пример — очень глубокое решающее дерево.

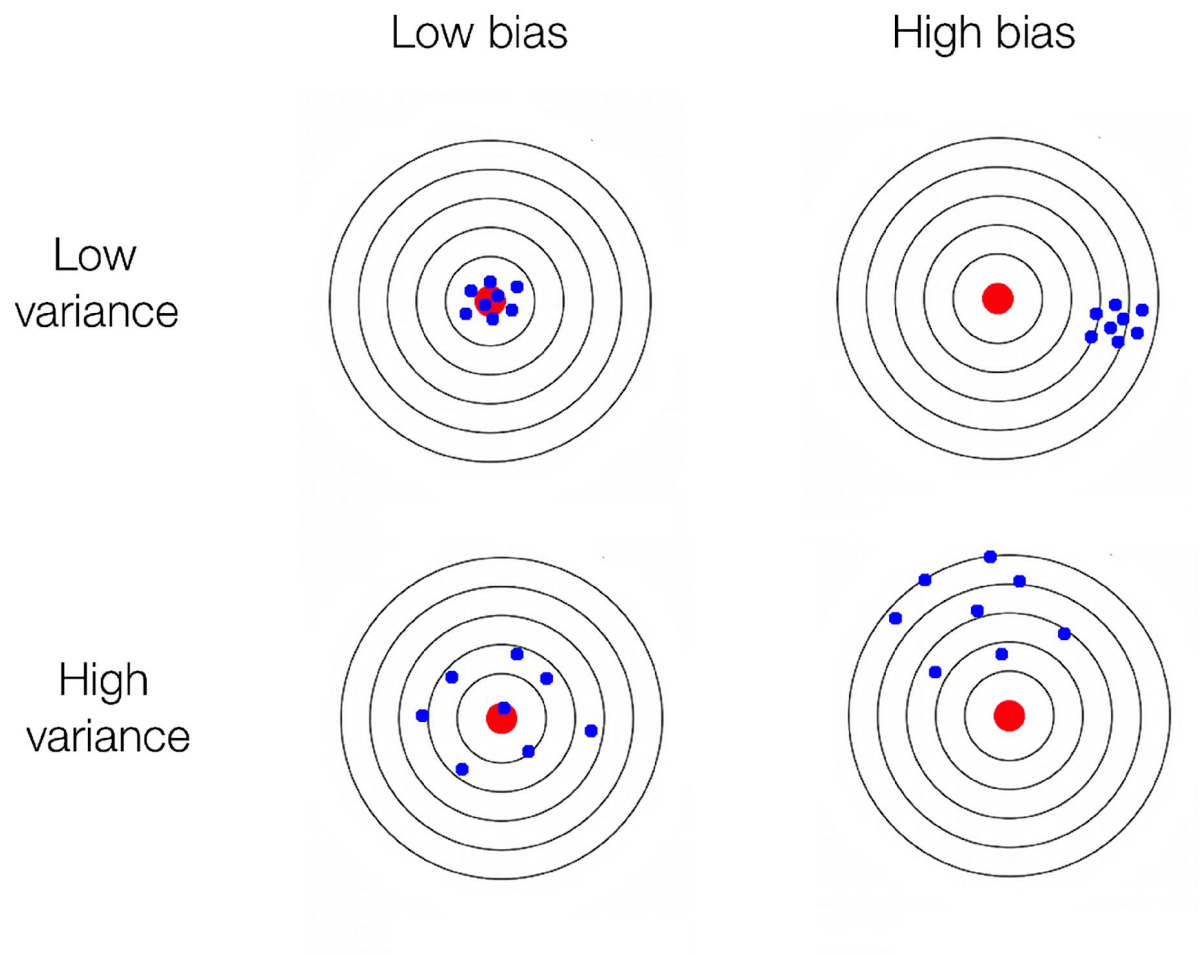
Ансамблевые методы

- Сегодня мы рассмотрим один из двух типов ансамблей над решающими деревьями — Random Forest.



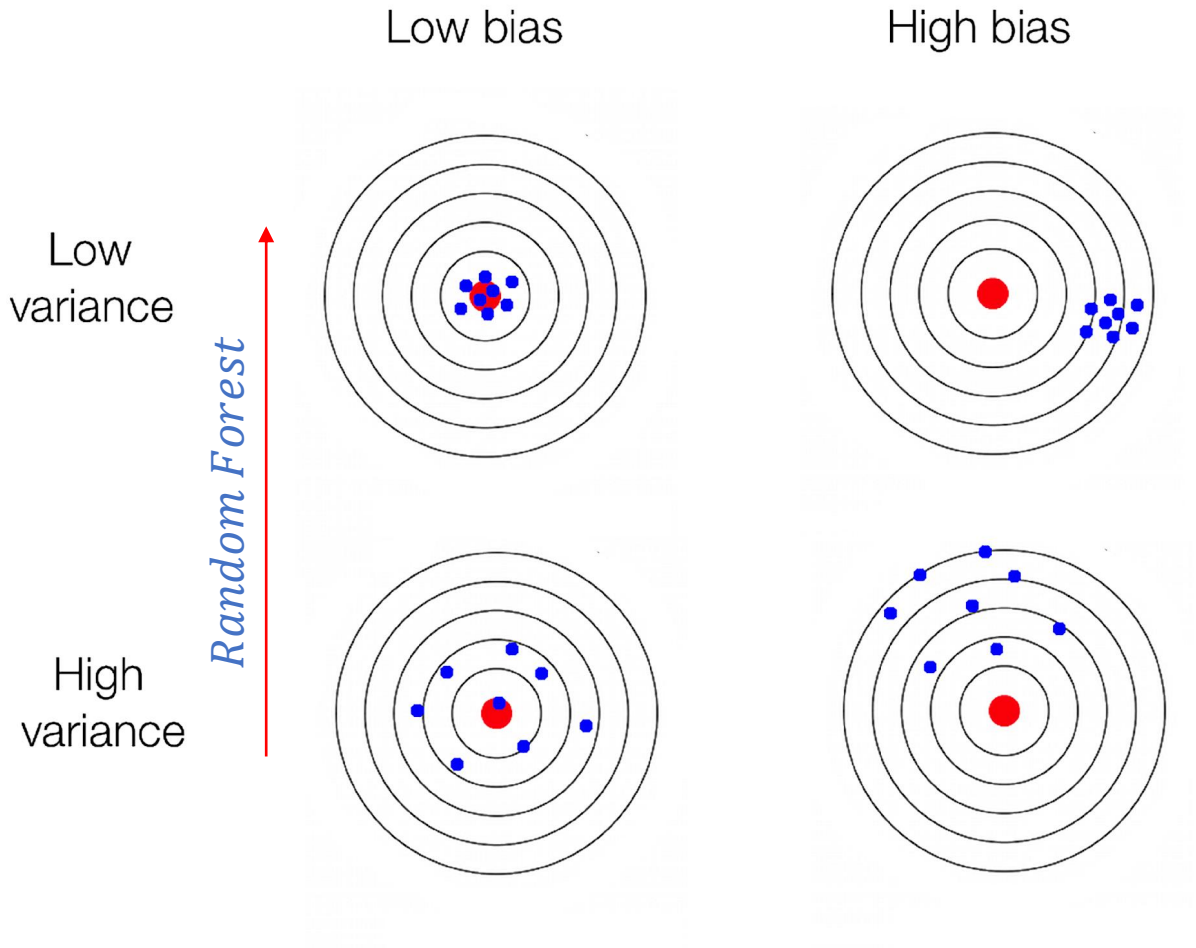
Ансамблевые методы

- Сегодня мы рассмотрим один из двух типов ансамблей над решающими деревьями — Random Forest.
  - Но, на самом деле, оба типа направлены на уменьшение bias-variance, просто начинают с разных слабых моделей — по большому счету, с противоположных сторон.



Ансамблевые методы

- Сегодня мы рассмотрим один из двух типов ансамблей над решающими деревьями — Random Forest.
  - Но, на самом деле, оба типа направлены на уменьшение bias-variance, просто начинают с разных слабых моделей — по большому счету, с противоположных сторон.



Random Forest

Random Forest

- Random Forest представляет собой обширное множество глубоких, переобученных решающих деревьев.
- В задаче регрессии ответы всех этих деревьев усредняются; в задаче классификации решение строится голосованием по большинству.

Random Forest

- Random Forest представляет собой обширное множество глубоких, переобученных решающих деревьев.
- В задаче регрессии ответы всех этих деревьев усредняются; в задаче классификации решение строится голосованием по большинству.
- Поскольку мы имеем дело с глубокими, переобученными деревьями, то у нас… (что можно сказать про смещение и разброс?)

Random Forest

- Random Forest представляет собой обширное множество глубоких, переобученных решающих деревьев.
- В задаче регрессии ответы всех этих деревьев усредняются; в задаче классификации решение строится голосованием по большинству.
- Поскольку мы имеем дело с глубокими, переобученными деревьями, то у нас большой разброс, но маленькое смещение; и ансамбль должен помочь нам уменьшить этот высокий разброс.

Random Forest

- Random Forest представляет собой обширное множество глубоких, переобученных решающих деревьев.
- В задаче регрессии ответы всех этих деревьев усредняются; в задаче классификации решение строится голосованием по большинству.
- Поскольку мы имеем дело с глубокими, переобученными деревьями, то у нас большой разброс, но маленькое смещение; и ансамбль должен помочь нам уменьшить этот высокий разброс.
- Но как?

Random Forest

- На самом деле, нам достаточно вспомнить одну замечательную теорему из теории вероятности — центральную предельную теорему!
- Кто помнит её формулировку?

Random Forest

- На самом деле, нам достаточно вспомнить одну замечательную теорему из теории вероятности — центральную предельную теорему!
- Кто помнит её формулировку?
- В одной из формулировок она звучит так: если мы усредним n одинаково распределенных независимых случайных величин, то у получившейся случайной величины будет такое же математическое ожидание, как у исходной, а дисперсия уменьшится в n раз.

Random Forest

- Но ведь ответ нашего решающего дерева — это тоже случайная величина (мы уже упоминали с вами раннее, что, по сути, любая модель машинного обучения представляет собой некое распределение из теории вероятностей, просто многомерное и очень сложное).

Random Forest

- Но ведь ответ нашего решающего дерева — это тоже случайная величина (мы уже упоминали с вами раннее, что, по сути, любая модель машинного обучения представляет собой некое распределение из теории вероятностей, просто многомерное и очень сложное).
- А раз это случайная величина, значит мы также сможем воспользоваться центральной предельной теоремой, в отношении предсказаний нашей модели, и единственное, по сути, что остается, — это постараться сделать ответы от разных деревьев независимыми!

Random Forest

- Проблема #1: Построение решающего дерева — детерминированный алгоритм. Как получить ***разные*** деревья?

Random Forest

- Проблема #1: Построение решающего дерева — детерминированный алгоритм. Как получить ***разные*** деревья?

- Решение #1:
  - Выберем с повторением n объектов из исходной выборки (т.н. bootstrap).
  - Всего это можно сделать n^n способами.
  - Будем обучать дерево на том, что получилось. Это т.н. bagging — bootstrap aggregating.

Random Forest

- Проблема #2: В каждом листе перебираются все признаки.
  - Бутстрепные выборки всё равно похожи друг на друга (т.к. взяты из одного распределения).
  - Если перебирать данные по всем признакам, то деревья тоже будут похожи, как ни крути.

Random Forest

- Проблема #2: В каждом листе перебираются все признаки.
  - Бутстрепные выборки всё равно похожи друг на друга (т.к. взяты из одного распределения).
  - Если перебирать данные по всем признакам, то деревья тоже будут похожи, как ни крути.

- Решение #2:
  - Для каждого дерева будем случайным образом отбирать m признаков ещё до его построения: m берём как $d/3$ для задачи регрессии и как $\sqrt{d}$ для задачи классификации
  - Это т.н. метод случайных подпространств.

Random Forest

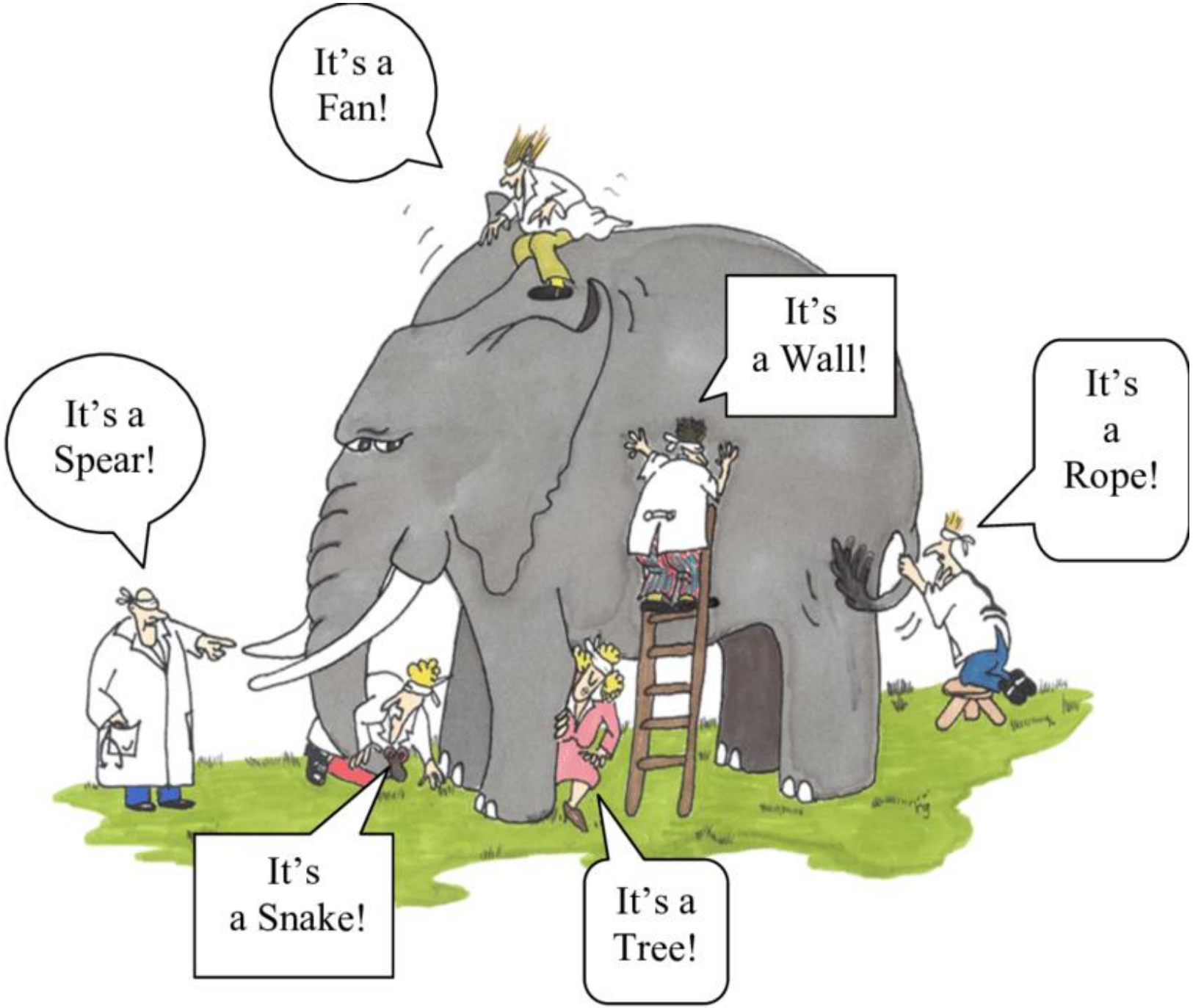
Random Forest = Bagging + Random Subspaces

Random Forest

- В случайном лесе каждое дерево уникально. Оно по построению слабо коррелирует с остальными.
- Засчёт этого деревья смотрят на разные закономерности, потому вместе могут «осмотреть» объект со всех сторон.

Random Forest

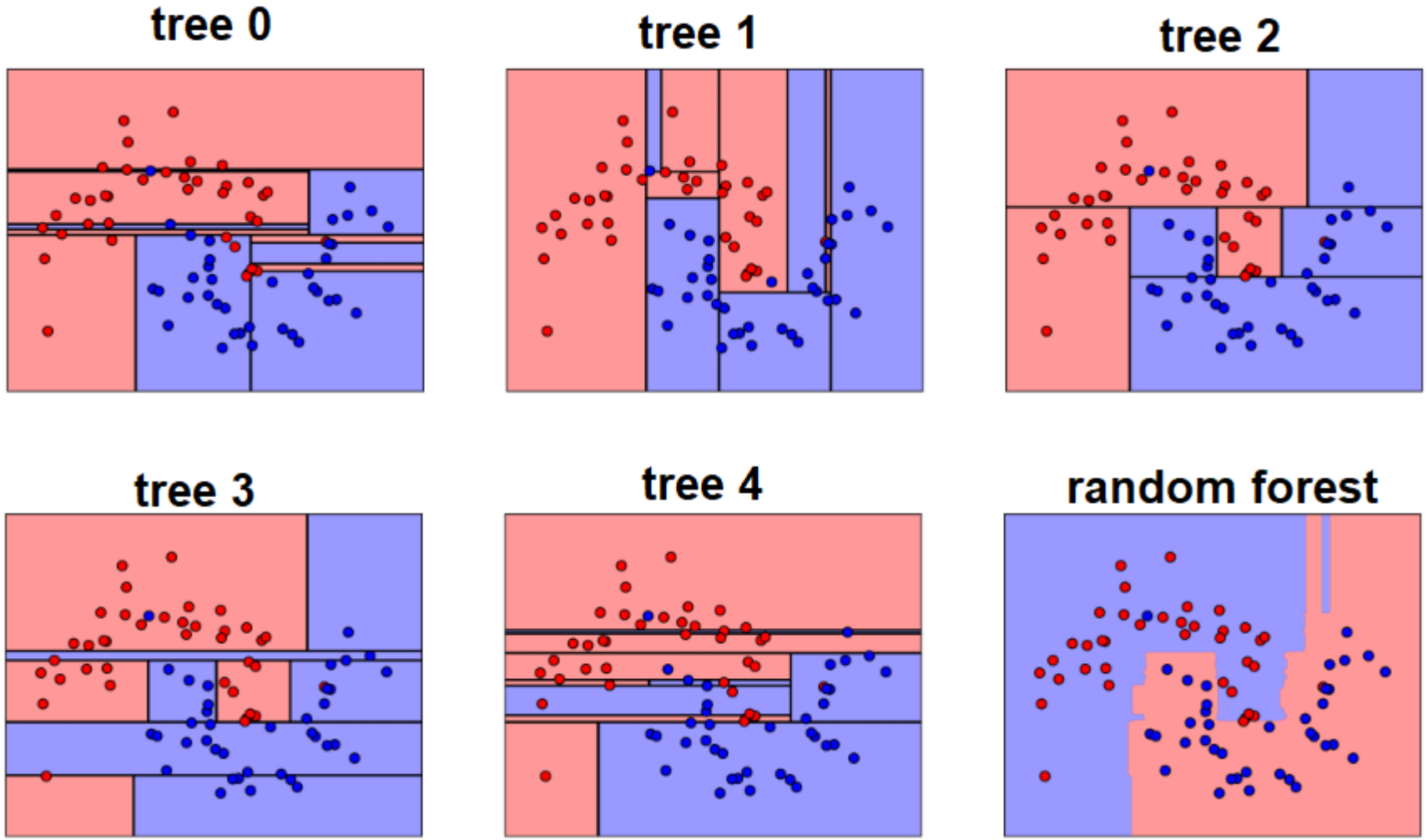
- В случайном лесе каждое дерево уникально. Оно по построению слабо коррелирует с остальными.
- Засчёт этого деревья смотрят на разные закономерности, потому вместе могут «осмотреть» объект со всех сторон.
- Используя концепцию случайного леса, мы научимся строить непохожие деревья и можем предполагать, что ответы этих деревьев будут в достаточной степени независимыми.
- Тогда усреднение ответов таких алгоритмов должно стабильно уменьшать разброс модели, при сохранении одновременно с этим — небольшого смещения (такого же, как у исходного дерева).


It's a Fan!
It's a Spear!
It's a Wall!
It's a Rope!
It's a Snake!
It's a Tree!

Random Forest

- Итого, подытожим идею случайного леса:
  - Для каждого дерева создадим подвыборку объектов и признаков.
  - Переобучим очень глубокие деревья на каждой подвыборке.
  - Агрегируем их предсказания.
  - Чем слабее коррелируют предсказания деревьев, тем сильнее уменьшится variance.
  - Bias же каждого дерева был изначально невелик. Если предсказания ещё и агрегировать, то он точно не сможет сильно вырасти.

Random Forest



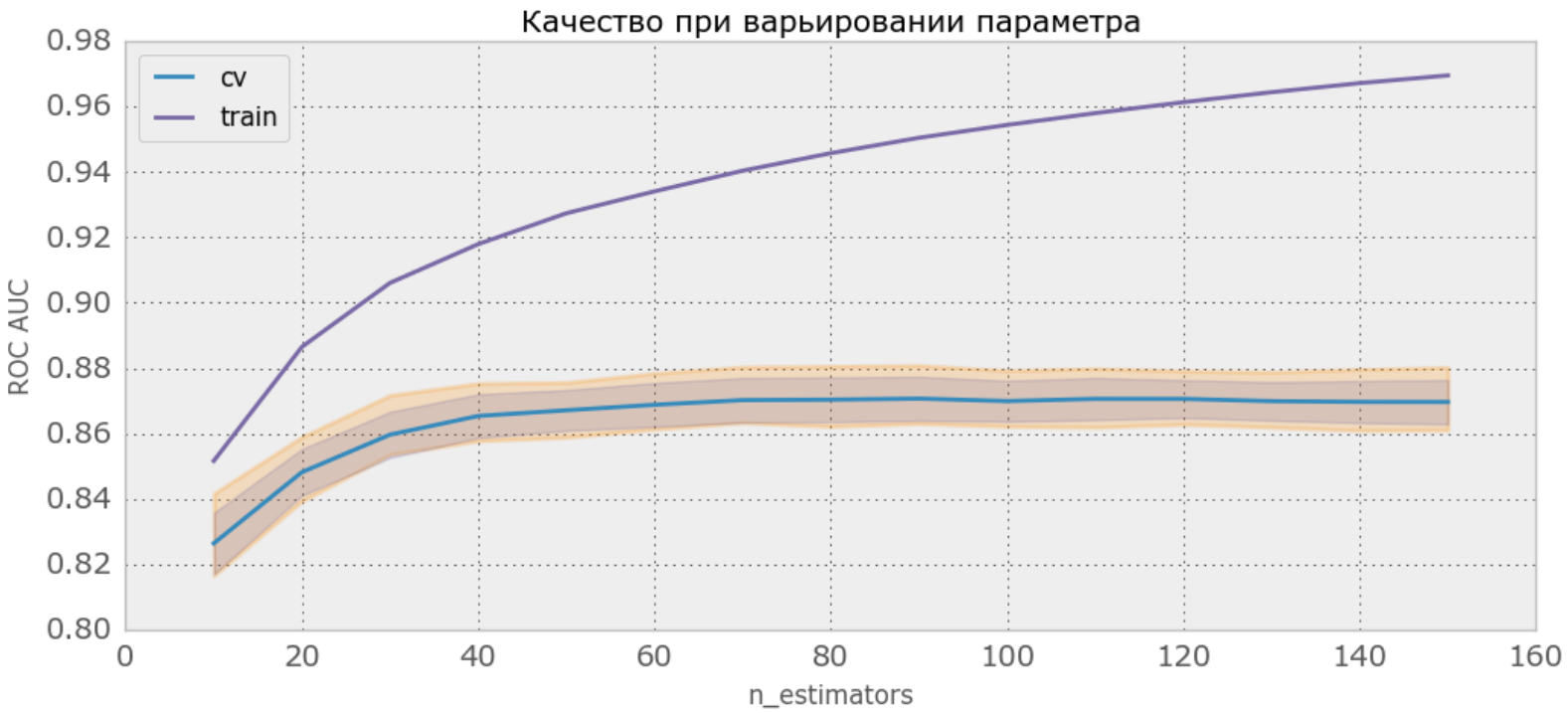
Наглядный пример работы случайного леса из 5 деревьев

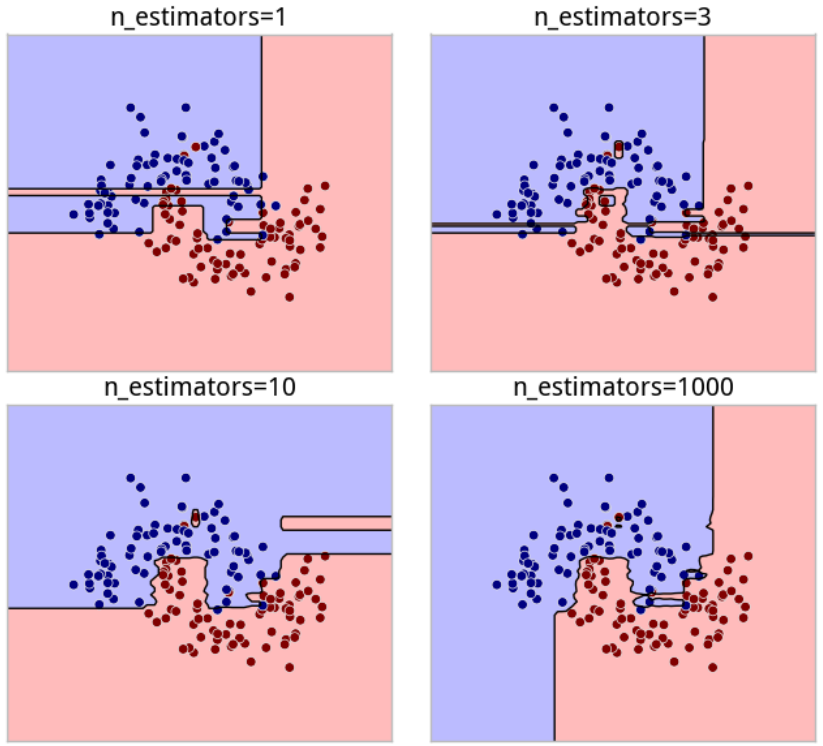
Random Forest

- Конечно же, немаловажно понимать и то, какие гиперпараметры есть у случайного леса, а также как их нужно настраивать и в каком приоритете.

Random Forest

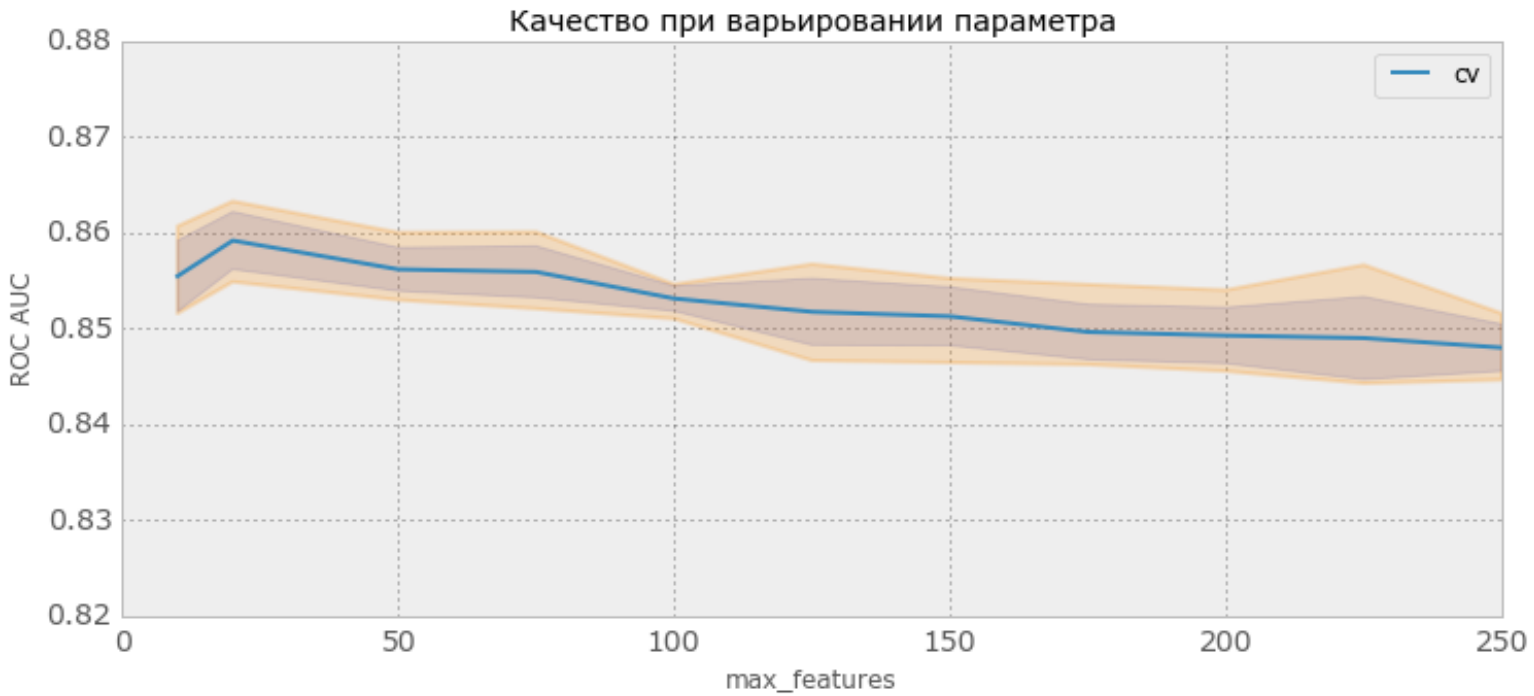
- N_estimators. Число деревьев в ансамбле.

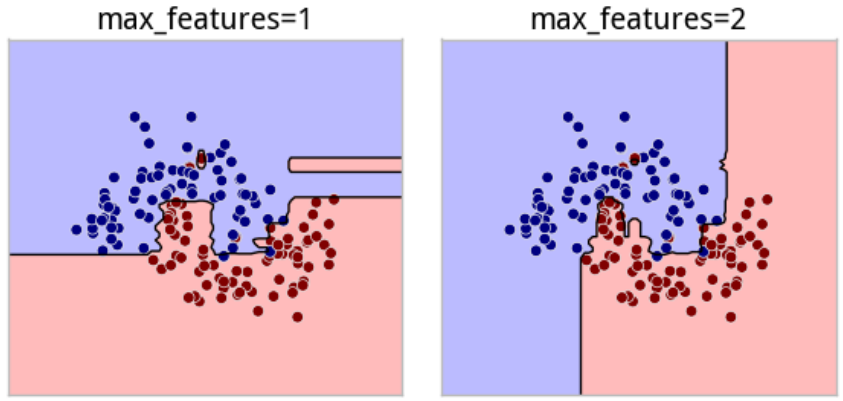




Random Forest

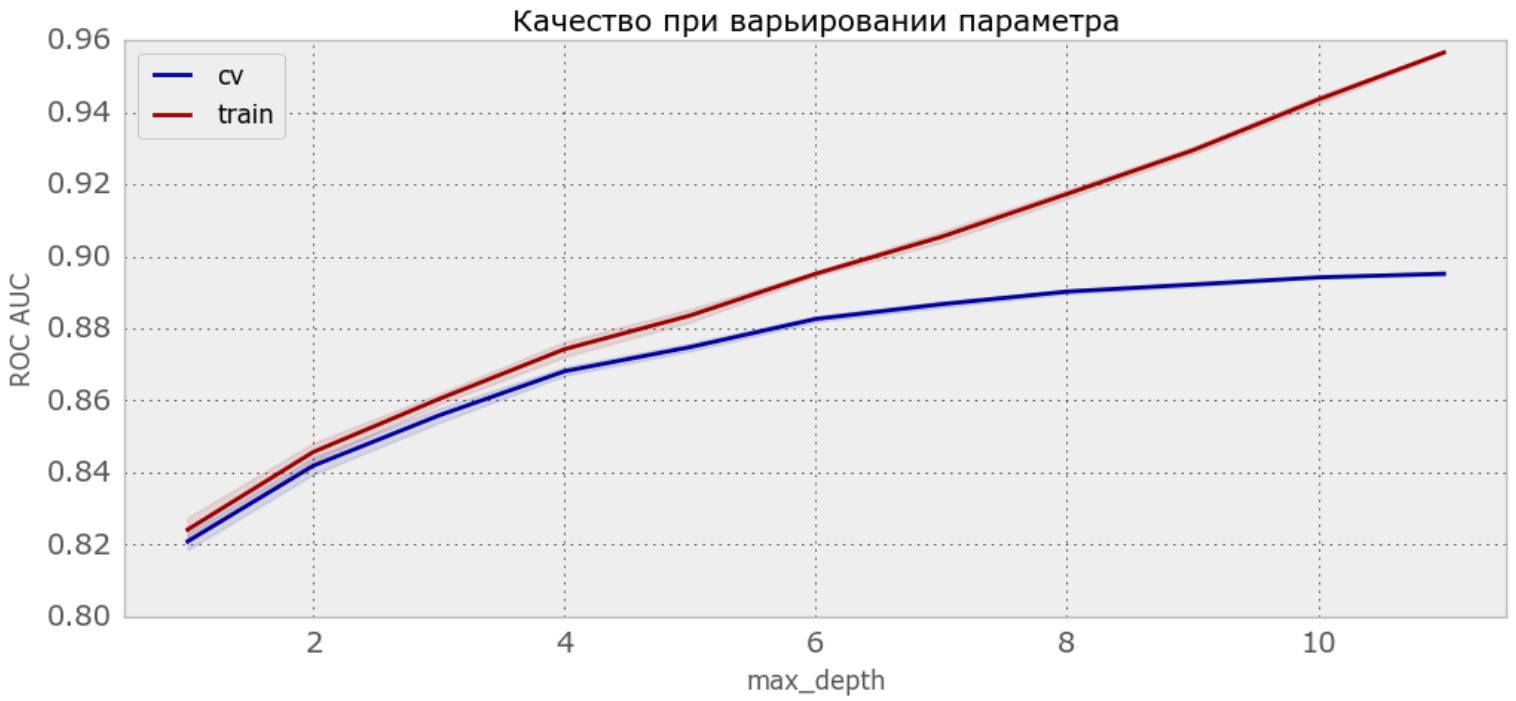
- Max_features. Число признаков для расщепления

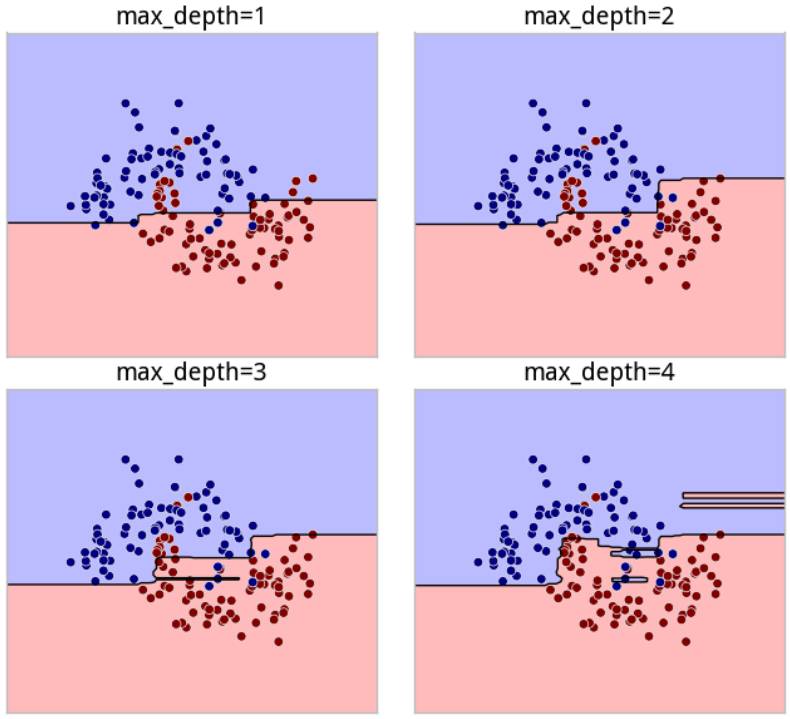




Random Forest

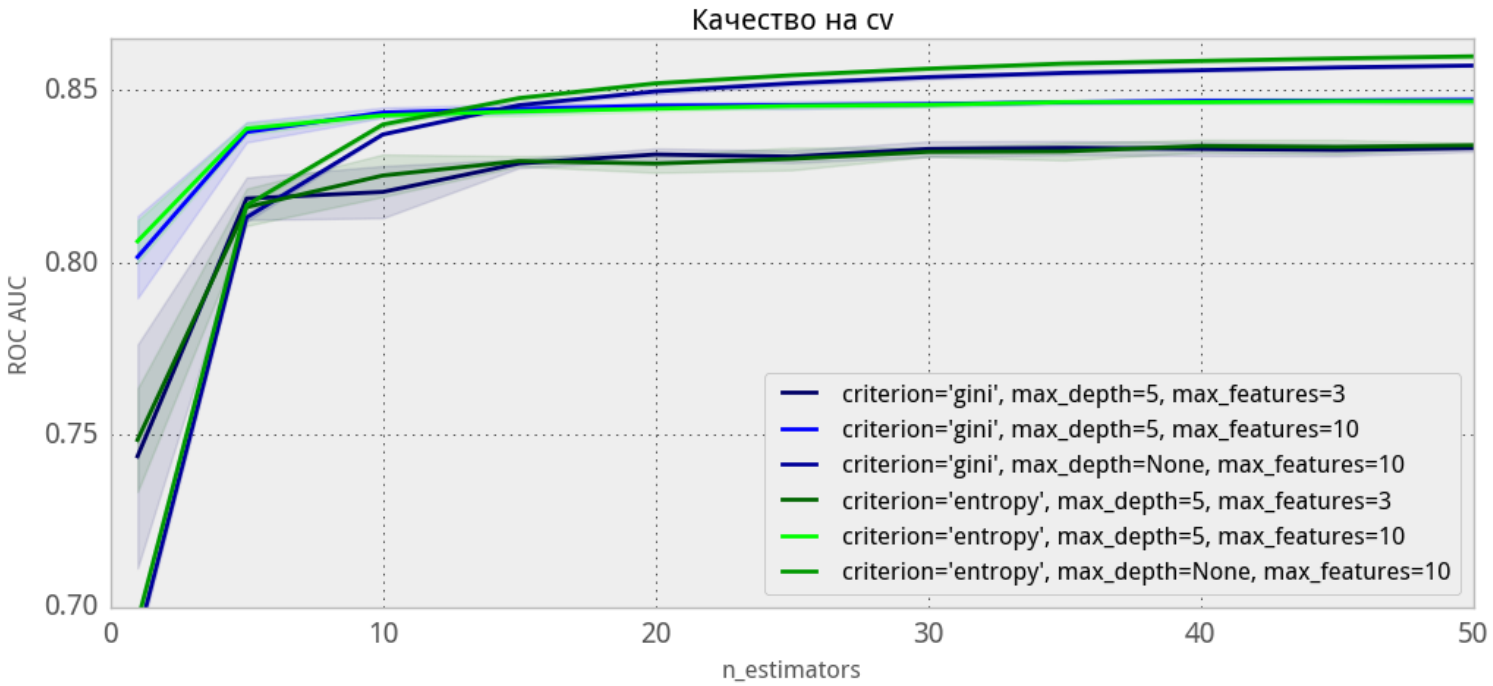
- Max_depth. Максимальная глубина деревьев

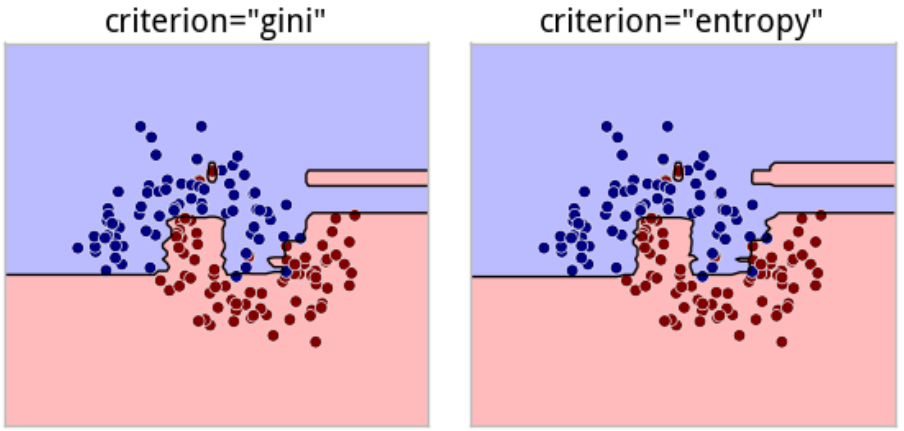




Random Forest

- Criterion. Критерий расщепления





Random Forest

- Совет по обучению: По умолчанию в sklearn-овских методах n_jobs=1, т.е. случайный лес строится на одном процессоре. Если вы хотите существенно ускорить построение, используйте n_jobs=-1 (строить на максимально возможном числе процессоров).
- Для построения воспроизводимых экспериментов используйте предустановку генератора псевдослучайных чисел: random_state.

Random Forest

- Совет по обучению: По умолчанию в sklearn-овских методах n_jobs=1, т.е. случайный лес строится на одном процессоре. Если вы хотите существенно ускорить построение, используйте n_jobs=-1 (строить на максимально возможном числе процессоров).
- Для построения воспроизводимых экспериментов используйте предустановку генератора псевдослучайных чисел: random_state.

Вопрос: а почему мультипроцессорность вообще работает в рамках этой модели? Так будет со всеми моделями, или RF чем-то особенен в этом контексте?

Random Forest. Плюсы

- Обычно достаточно хорошо работает “из коробки”.
  - Можно даже выбросы не удалять — их всё равно мало и шанс их попадания в бутстрепную выборку невелик.
- Может обрабатывать и категориальные, и числовые признаки.
  - Модели, которые способны обрабатывать категориальные столбцы без кодирования и дополнительной предобработки, — можно буквально по пальцам пересчитать.
- Тривиально обобщается на случай многомерной целевой переменной.
  - Будь то многоклассовая классификация или векторная регрессия.
- Обучение прекрасно параллелится.
  - Все деревья можно обучать независимо. Можно использовать MapReduce.

Random Forest. Минусы

- Интерпретируемость алгоритма под вопросом.
  - По единичному объекту понять, почему лес выдал именно такое предсказание, очень трудно. Особенно если деревьев много.
  - Тем не менее, есть подходы к извлечению важности признаков из случайного леса, так что не всё потеряно.
- Если у данных низкая дисперсия, а признаки коррелируют, то трудно добиться некоррелированности деревьев.
  - А без этого ничего работать не будет.