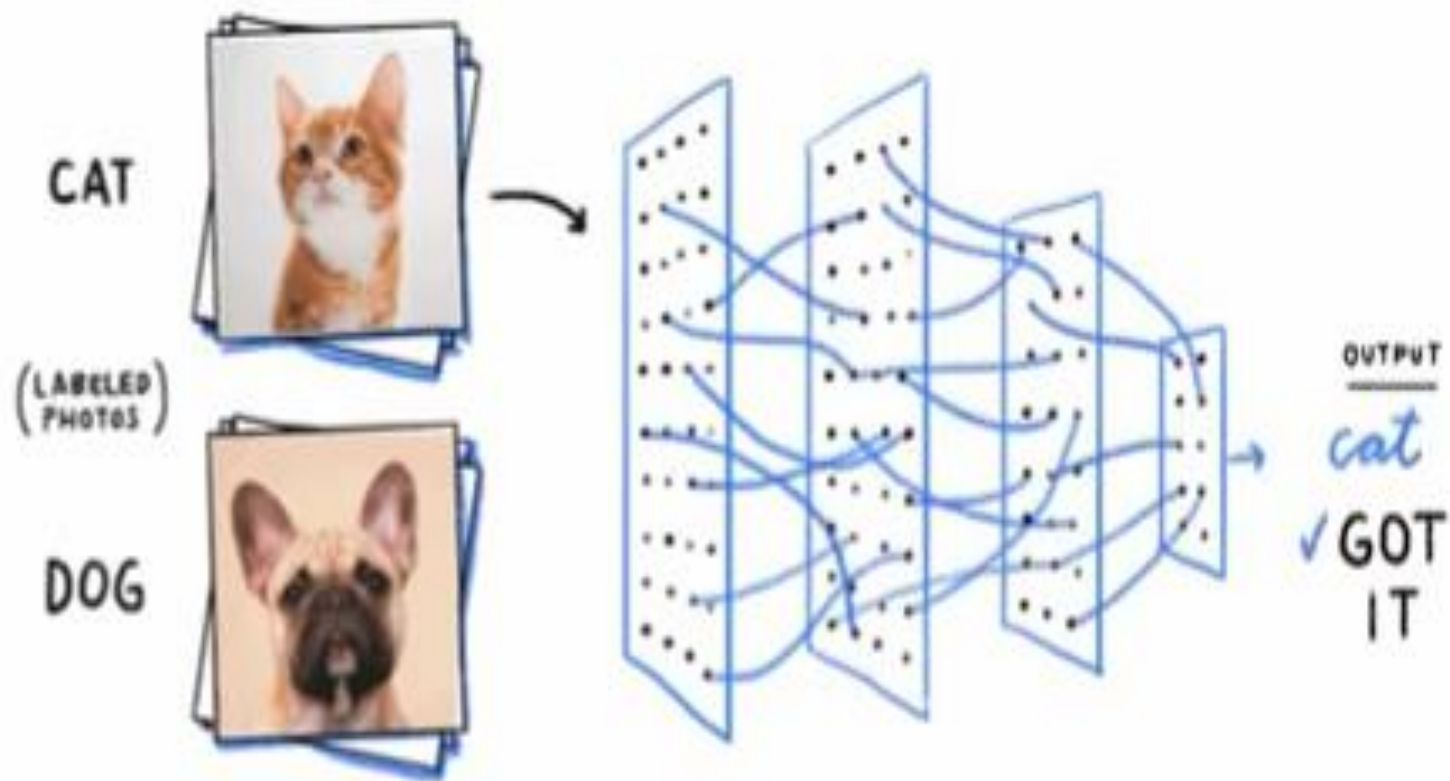


# Классификация

Семинар 8

# В чем суть классификации?





# Примеры задач классификации

- Задача классификации — это любая задача, где нужно определить тип объекта из двух и более существующих классов. Такие задачи могут быть разными: определение, кошка на изображении или собака, или определение качества вина на основе его кислотности и содержания алкоголя.
- В зависимости от задачи классификации вы будете использовать разные типы классификаторов. Например, если классификация содержит какую-то бинарную логику, то к ней лучше всего подойдёт логистическая регрессия.
- По мере накопления опыта вам будет проще выбирать подходящий тип классификатора. Однако хорошей практикой является реализация нескольких подходящих классификаторов и выбор наиболее оптимального и производительного.
- Классификация – обучение с учителем. Такой тип обучения подразумевает, что данные, подаваемые на входы системы, уже помечены, а важная часть признаков уже разделена на отдельные категории или классы. Поэтому сеть уже знает, какая часть входов важна, а какую часть можно самостоятельно проверить.

# Процесс машинного обучения

- Процесс содержит в себе следующие этапы: подготовка данных, создание обучающих наборов, создание классификатора, обучение классификатора, составление прогнозов, оценка производительности классификатора и настройка параметров.
- Во-первых, нужно подготовить набор данных для классификатора — преобразовать данные в корректную для классификации форму и обработать любые аномалии в этих данных. Отсутствие значений в данных либо любые другие отклонения — все их нужно обработать, иначе они могут негативно влиять на производительность классификатора. Этот этап называется предварительной обработкой данных (англ. *data preprocessing*).
- Следующим шагом будет разделение данных на обучающие и тестовые наборы.
- Как уже было сказано выше, классификатор должен быть создан и обучен на тренировочном наборе данных. После этих шагов модель уже может делать прогнозы. Сравнивая показания классификатора с фактически известными данными, можно делать вывод о точности классификатора.
- Вероятнее всего, вам нужно будет «корректировать» параметры классификатора, пока вы не достигните желаемой точности (т. к. маловероятно, что классификатор будет соответствовать всем вашим требованиям с первого же запуска).



# Алгоритм исследования данных

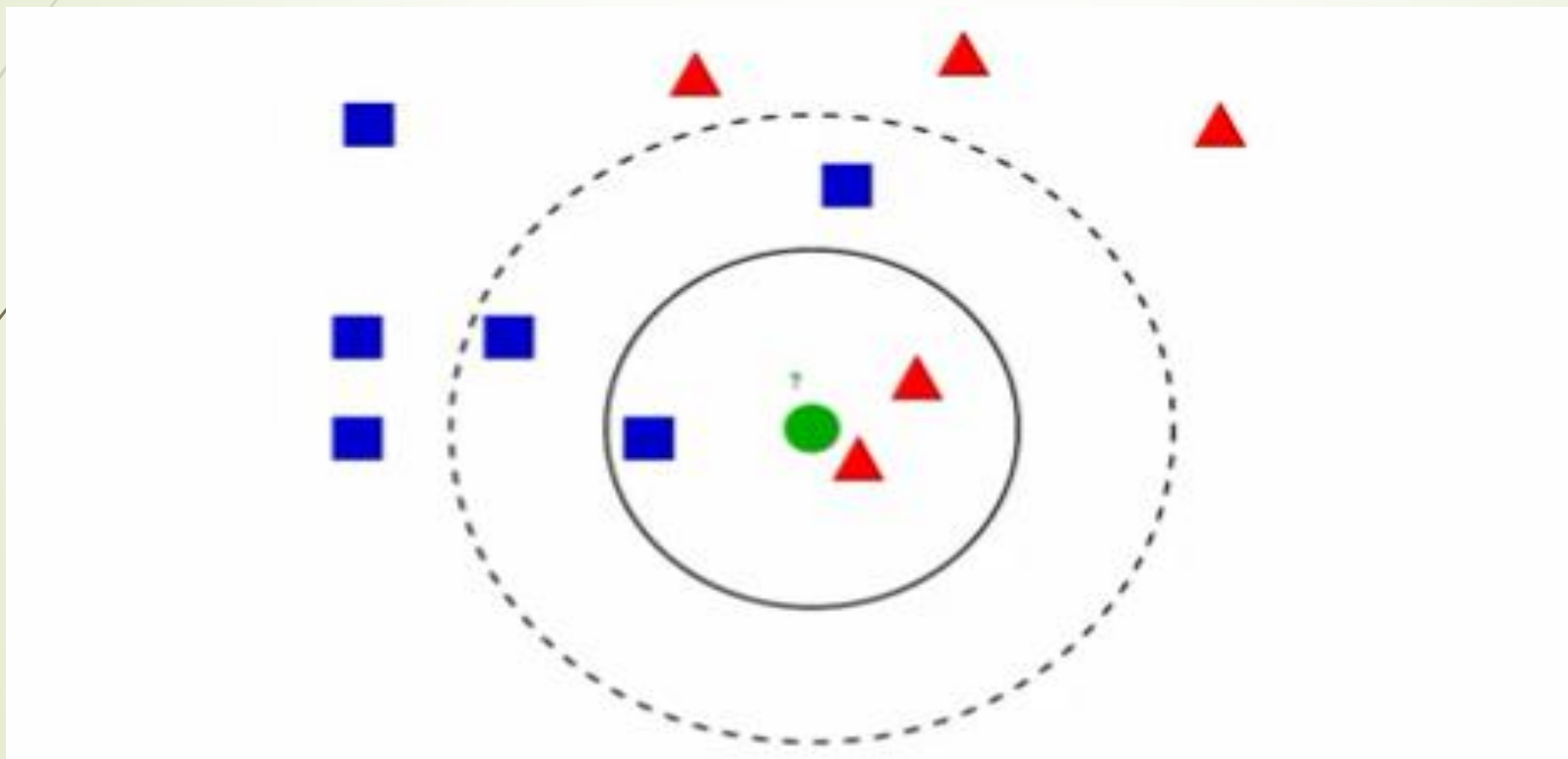
- Сбор и очистка данных. Как показывает практика, этот этап может занимать до 90% времени всего анализа данных;
- Визуальный анализ данных, их распределение, статистики;
- Анализ зависимости (корреляции) между переменными (признаками);
- Отбор и определение признаков, которые будут использоваться для построения моделей;
- Разделение на данные для обучения модели и тестовые;
- Построение моделей на данных для обучения / оценка результата на тестовых данных;
- Интерпретация полученной модели, визуализация результатов.



# Зеленый на чьей стороне?

Скажи мне кто твои друзья, а я скажу кто ты.

Предсказание делается на основе наблюдений наиболее близких соседей  $k=3$ ,  $K=5$





# Как это работает

- У нас есть тренировочная выборка объектов
- При поступлении нового объекта (тестового) мы ищем объекты из тренировочной выборки, больше всего на него похожих

# Алгоритм $k$ ближайших соседей

## Алгоритм классификации

1. Взять точку  $X$  для анализа
2. Упорядочить множества точек и классов по расстоянию
3. Выбрать первые  $k$  ближайших соседей
4. Создать массив голосов за каждый из классов
5. Присвоить точке  $X$  класс с максимальным числом голосов





# Характеристика метода knn

- Часто  $k$  выбирают нечетным
- Чем меньше  $k$  тем более гибкая модель, но тем больше нужно данных, чтобы не переобучиться
- С ростом размерности требуется больше данных (соседи стоят далеко)
- Если количество данных стремится к бесконечности, то метод стремится к наилучшей оценке



# Пример работы с knn алгоритмом

- ▶ <https://colab.research.google.com/drive/1-ejKd0lxPzyCCRDoFDPvOkHWHG9oGZbV?usp=sharing>

# Пример

## Кому дать кредит?

похожие объекты	№		Возраст	Ежемесячный доход (руб)	Наличие судимости
	1	Антон Петров	35	18000	1 (да)
	2	Лида Ермакова	45	50000	0 (нет)
	3	Олег Державин	33	20000	1 (да)
	...				

похожие ответы	№	Кредитный индекс
	1	0
	2	1
	3	0
	...	

# Регрессии

- Необходимы для предсказания
- Линейная регрессия – бинарный классификатор, разделяющий данные на «выше» и «ниже»

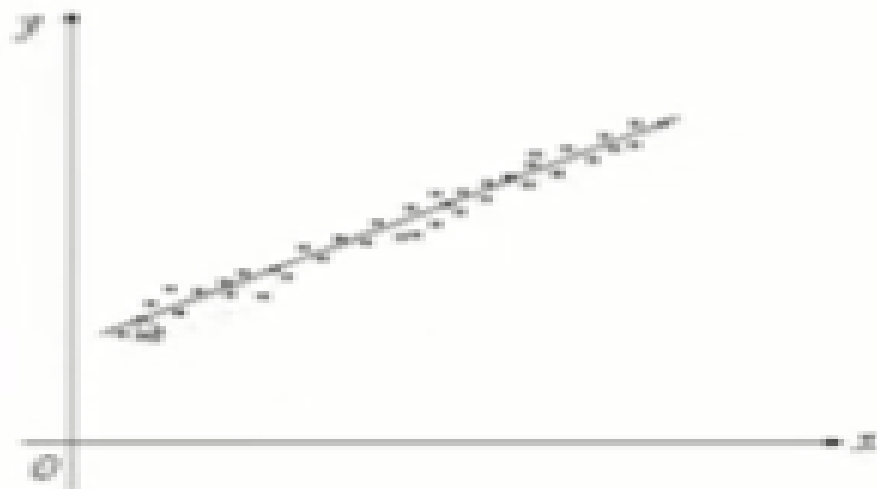


Рисунок 1 – Линейная регрессия

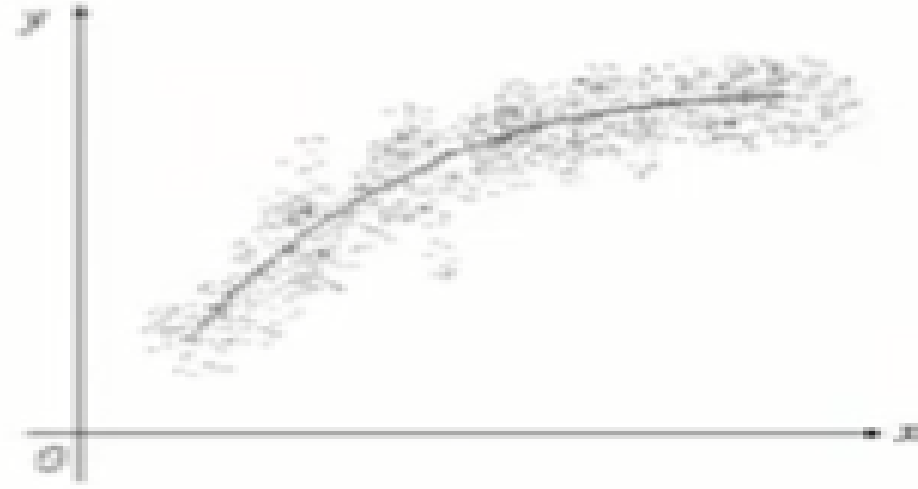
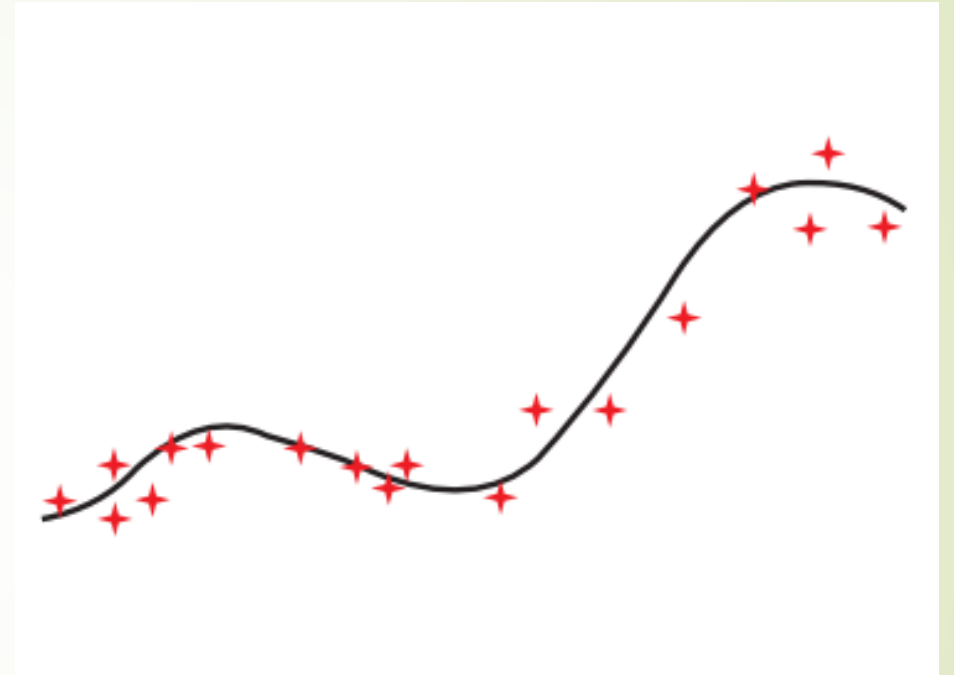


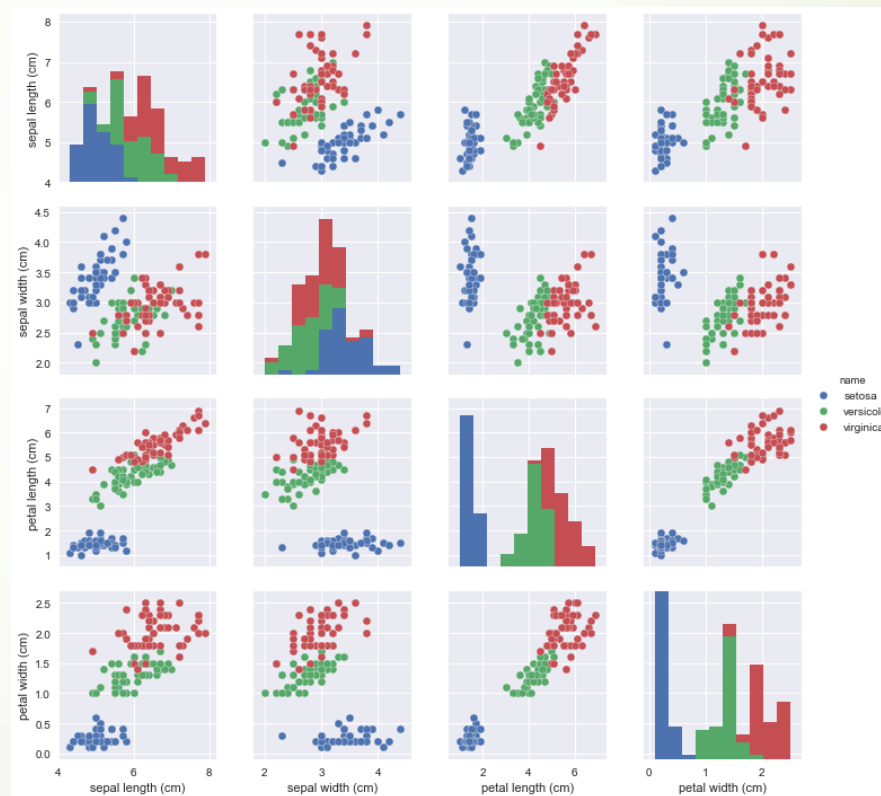
Рисунок 2 – Нелинейная регрессия

# Задача восстановления регрессии

- Оценка стоимости недвижимости: по характеристике района, экологической обстановке, транспортной связности оценить стоимость жилья
- Прогноз свойств соединений: по параметрам химических элементов спрогнозировать температуру плавления, электропроводность, теплоемкость получаемого соединения
- Медицина: по постоперационным показателям оценить время заживления органа
- Кредитный скоринг: по анкете заемщика оценить величину кредитного лимита
- Инженерное дело: по техническим характеристикам автомобиля и режиму езды спрогнозировать расход топлива

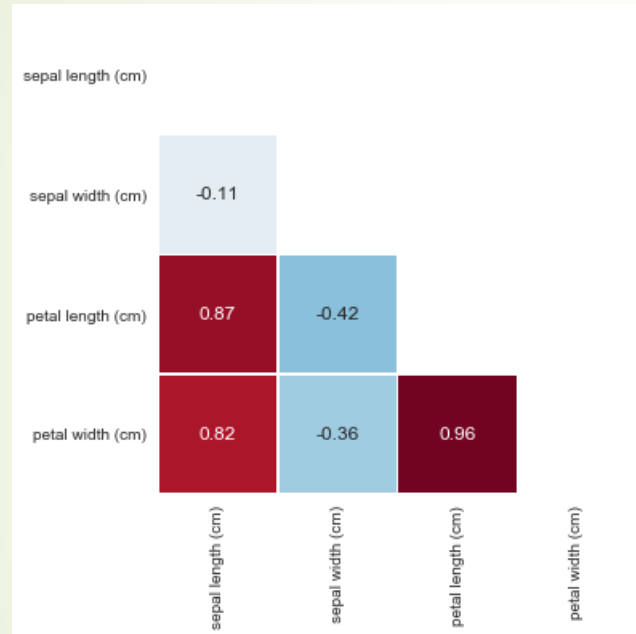


# Ирисы. Описательные статистики





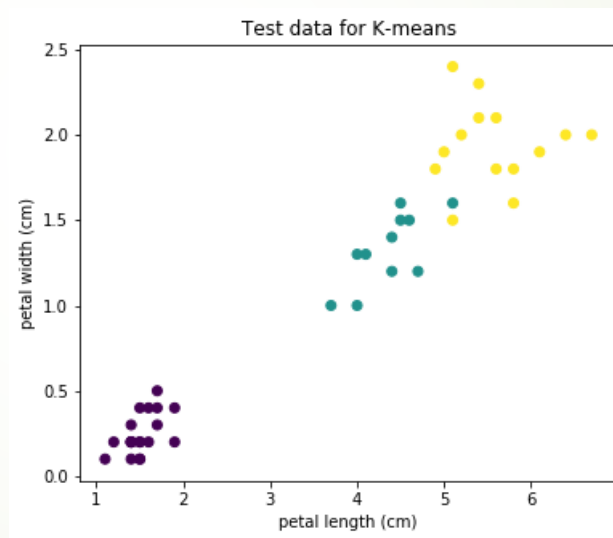
# Зависимость между переменными



- До 0,2 — очень слабая корреляция
- До 0,5 — слабая
- До 0,7 — средняя
- До 0,9 — высокая
- Больше 0,9 — очень высокая

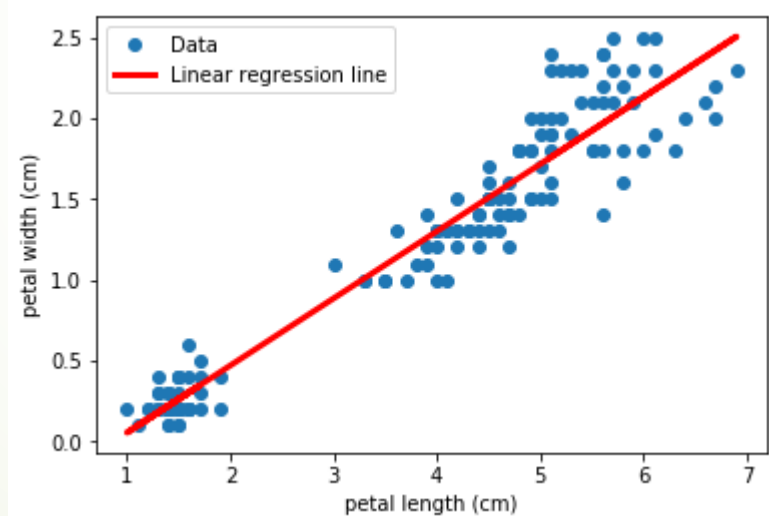
Действительно видим, что между переменными «petal length (cm)» и «petal width (cm)» выявлена очень сильная зависимость 0.96.

# Кластеризация ирисов



# Линейная регрессия для оценки результата

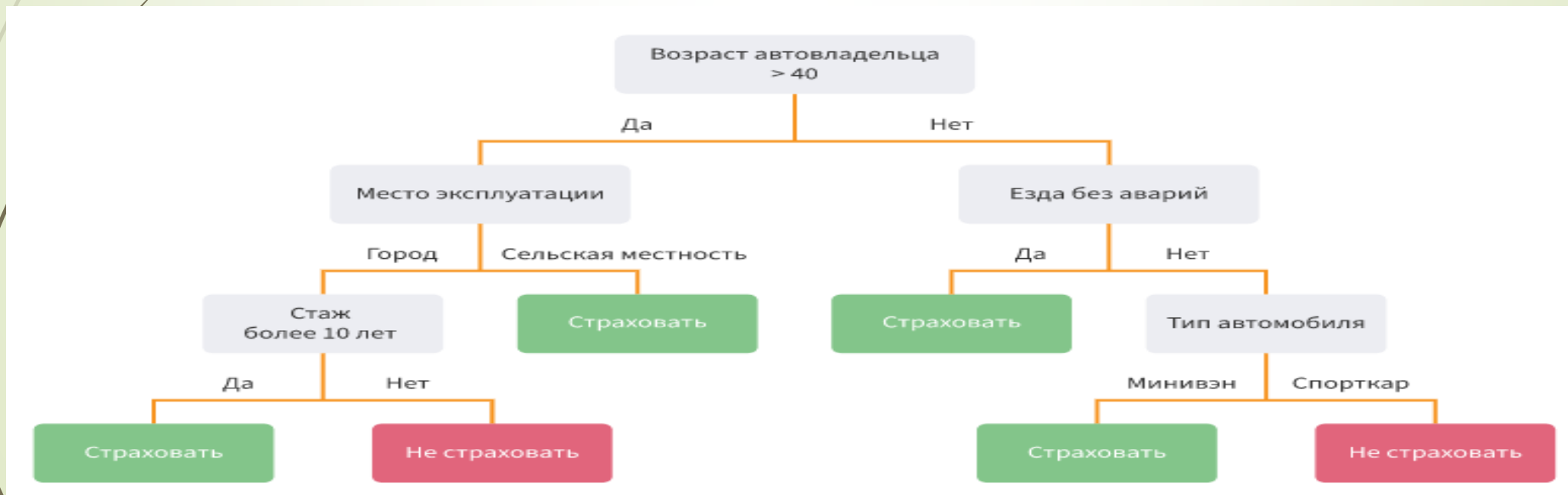
- Будем воспринимать линейную регрессию как абстрактный алгоритм нахождения линии, которая наиболее точно повторяет направление распределения объектов. Строим модель, используя переменные, которые, как мы поняли ранее, имеют сильную зависимость — это «petal length (cm)» и «petal width (cm)».



- Видим, что, действительно, найденная линия регрессии хорошо повторяет направление распределения точек. Теперь, если у нас будет в наличии, например, длина листочка `petal`, мы сможем с большой точностью определить, какая у него ширина!

# Методы классификации

- Дерево принятия решений. Цель – однозначно определиться с отнесением данным к тому или иному классу. Берем множество, выбираем параметр разделения, делим по нему на подмножества





# Деревья решений

- Преимущества

Интерпретируемость и наглядность

Возможность работать как с категориями так и с количественными значениями

Высокая производительность при классификации

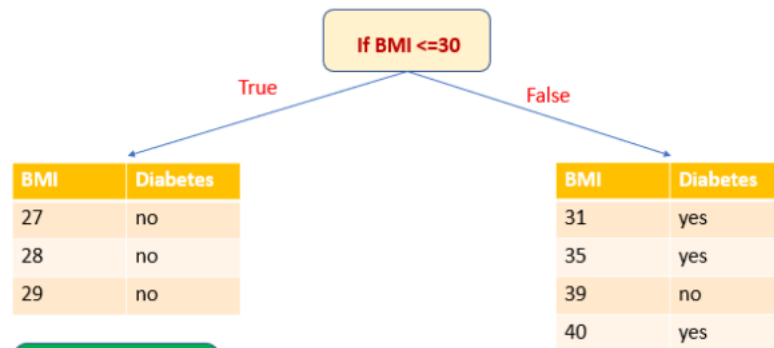
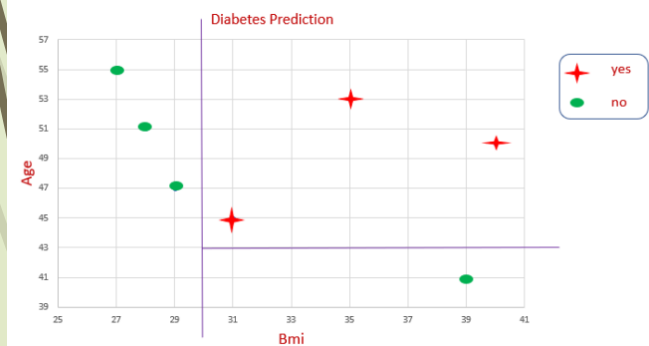
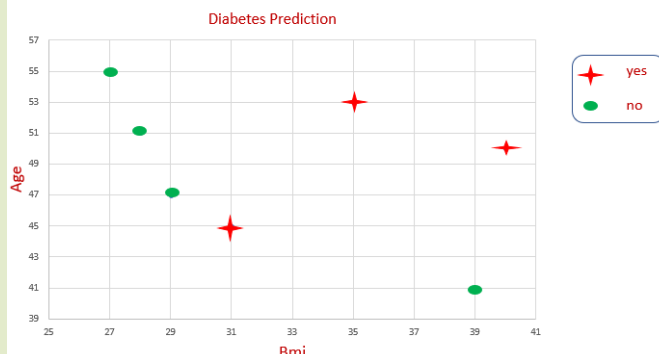
- Недостатки

Нестабильность структуры

Сложность контроля размера дерева

# Деревья и энтропия

Bmi	Age	Diabetes
31	45	yes
29	47	no
27	55	no
35	53	yes
28	51	no
40	50	yes
39	41	no



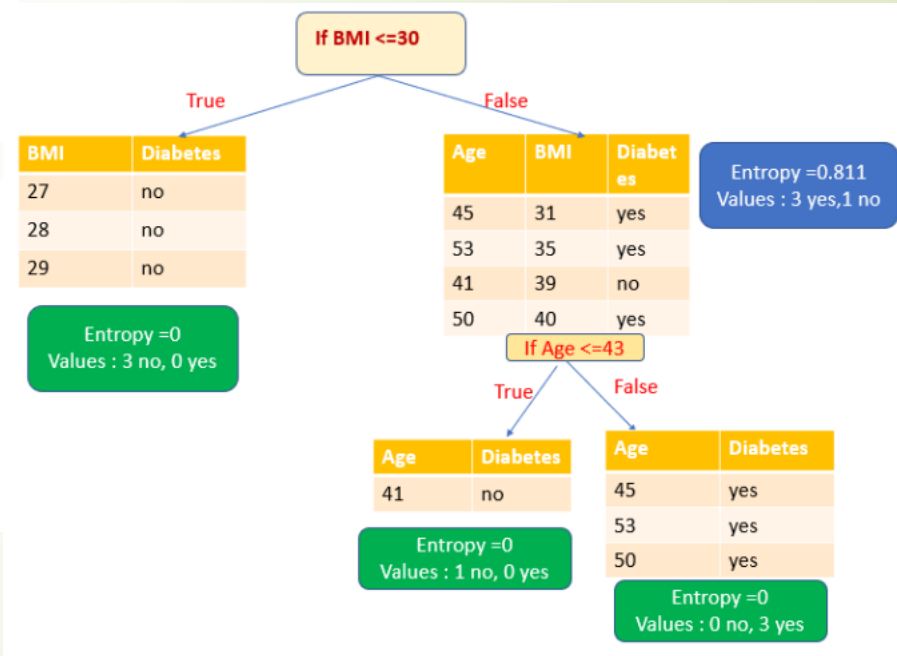
Entropy = 0  
Values : 3 no, 0 yes

Entropy = 0.811  
Values : 3 yes, 1 no

$P(0)=1/4$  [одна из четырех записей]  
 $P(1)=3/4$  [три из четырех записей]

$$\begin{aligned} \text{Entropy} &= -(p(0) * \log(P(0)) + p(1) * \log(P(1))) \\ &= -(1/4 * \log_2(1/4) + 3/4 * \log_2(3/4)) \\ &= -(0.25 * (-2) + 0.75 * (-0.415)) \\ &= -((-0.5) + (-0.311)) \\ &= -(-0.811) \end{aligned}$$

Entropy=0.811




Entropy = 0  
Values : 3 no, 0 yes

Entropy = 0.811  
Values : 3 yes, 1 no

Entropy = 0  
Values : 1 no, 0 yes

Entropy = 0  
Values : 0 no, 3 yes





# Определение точности работы классификатора

- Accuracy = точность ответов = правильные ответы/все ответы
- Recall = насколько правильно система определяет искомый класс, сколько объектов из предъявленных было правильно классифицировано
- Precision[i] = точность по i классу, насколько можно верить ответам
- F мера =  $P * R / (P + R)$
- Устойчива модель тогда, когда ошибка приемлема
- Куда денем score?

# Метрики

- accuracy — это главная метрика, которая показывает долю правильных ответов модели. Ее значение равно отношению числа правильных ответов, которые дала модель, к числу всех объектов. Но она не полностью отражает качество модели. Поэтому вводятся precision и recall.
- precision (точность) — эта метрика показывает, насколько мы можем доверять модели, другими словами, какое у нас количество «ложных срабатываний». Значение метрики равно отношению числа ответов, которые модель считает правильными, и они действительно были правильными (это число обозначается «true positives») к сумме «true positives» и числа объектов которые модель посчитала правильными, а на самом деле они были неправильные (это число обозначается «false positives»). В виде формулы:  $\text{precision} = \frac{\text{«true positives»}}{\text{«true positives»} + \text{«false positives»}}$
- recall (полнота) — эта метрика показывает насколько модель может вообще обнаруживать правильные ответы, другими словами, какое у нас количество «ложных пропусков». Ее численное значение равно отношению ответов, которые модель считает правильными, и они действительно были правильными к числу всех правильных ответов в выборке. В виде формулы:  $\text{recall} = \frac{\text{«true positives»}}{\text{«all positives»}}$
- f1-score (f-мера) — это объединение precision и recall



# Пример с цифрами

- ▶ [https://colab.research.google.com/drive/1-WGbK6KqUE3LVrR3X3CS6t\\_kZPSYHD2V?usp=sharing](https://colab.research.google.com/drive/1-WGbK6KqUE3LVrR3X3CS6t_kZPSYHD2V?usp=sharing)



# Задание

- Классифицировать новый объект в интересной для Вас выборке
- 



# Ансамбли в классификации



# Литература

- [neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/](https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/)



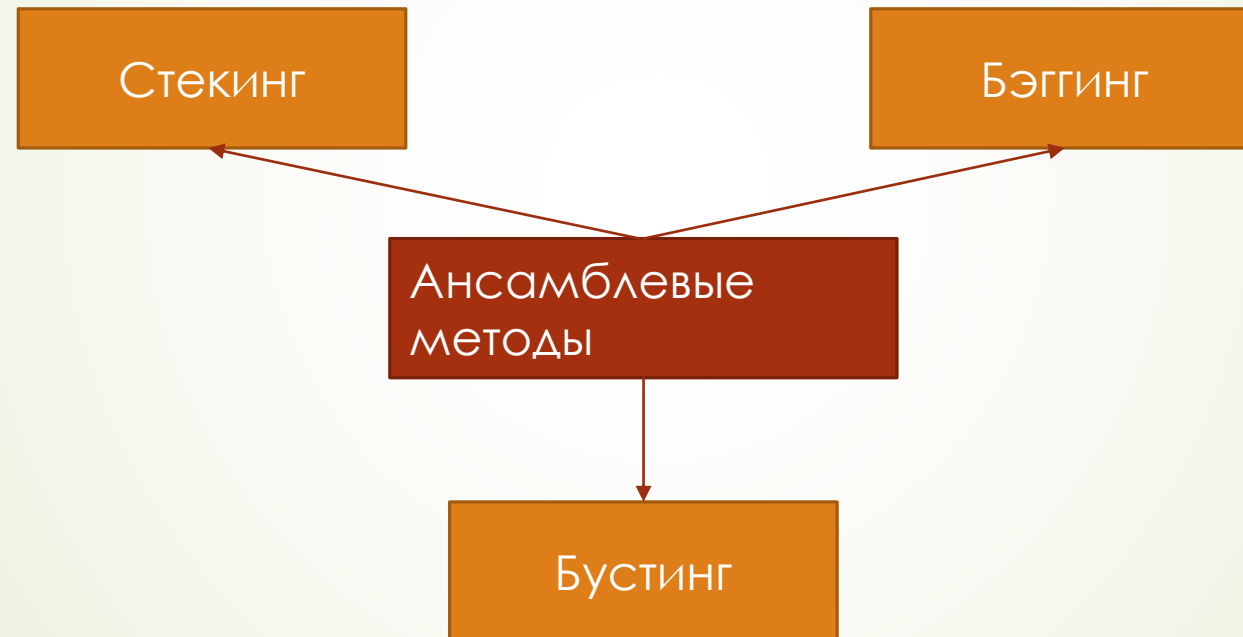


# Ансамбли



- Ни один из методов не является идеальным и именно поэтому их появилось так много; исследователи и энтузиасты направления не один год ломают голову над поиском компромисса между точностью, простотой и интерпретируемостью каждой отдельной модели. Однако хочется отметить, что большинство экспертов отдает предпочтение точности – в самом деле, если подумать, именно это качество и делает модель полезной (хотя мы не станем отрицать, что такое мнение было и остается несколько субъективным).
- Итак, как же повысить точность модели, желательно, не изменяя ее сути? Одним из способов повышения точности моделей является создание и обучение ансамблей моделей – то есть наборов моделей, используемых для решения одной и той же задачи. Под обучением ансамбля понимается обучение конечного набора базовых классификаторов с последующим объединением результатов их прогнозирования в единый прогноз агрегированного классификатора. Понятно, что объединенный (агрегированный) классификатор даст более точный результат.

# Куча алгоритмов учатся исправлять ошибки друг друга

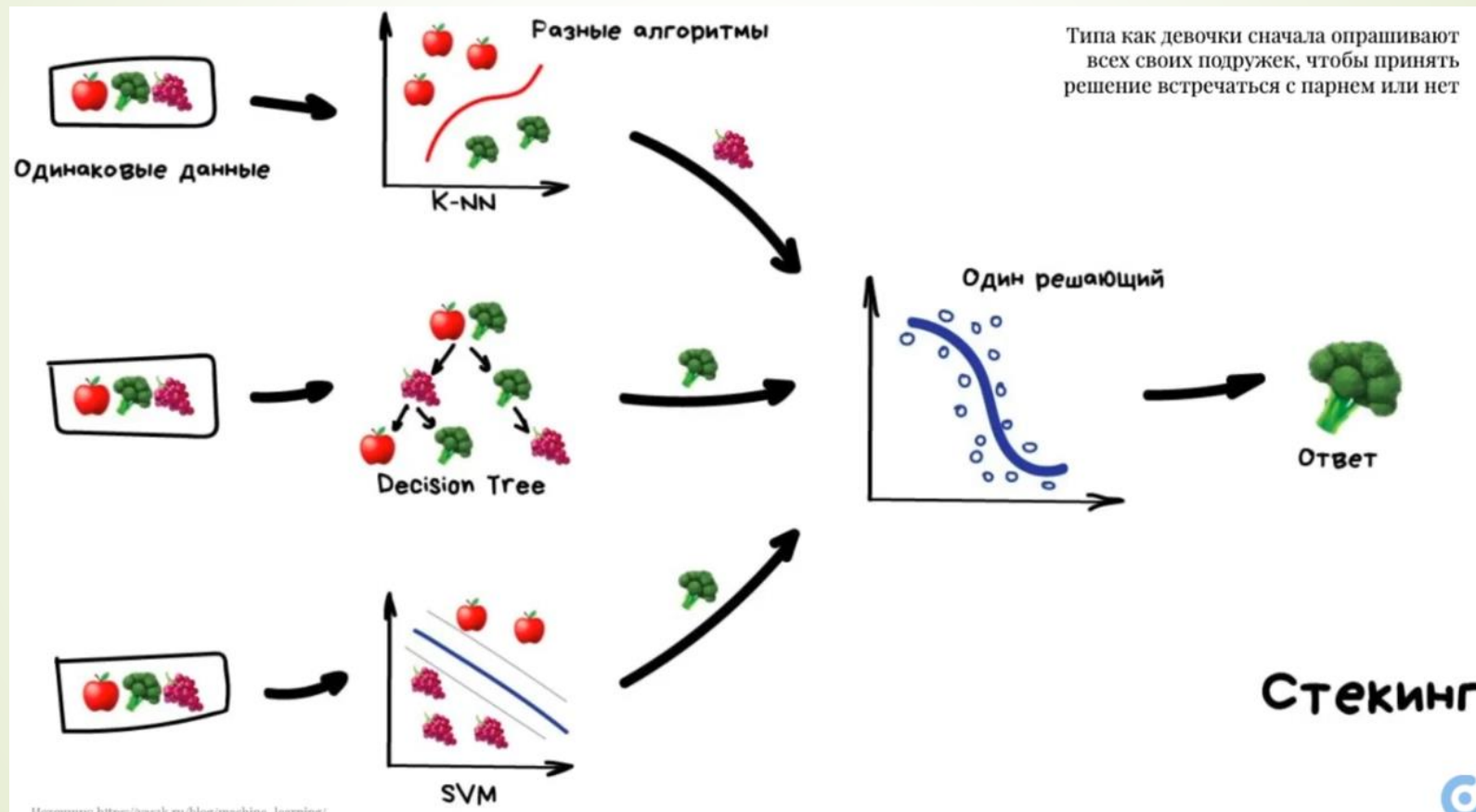




# Стекинг

- Суть: начальник 1 уровня принимает решение на основании мнений подчиненных. Начальник 2 уровня принимает решение на основании мнений начальников 1 уровня и т.д.
- Применение: распознавание сложного изображений. Один видит линию, другой - дом, третий – ландшафт.

# Разные алгоритмы на одних данных



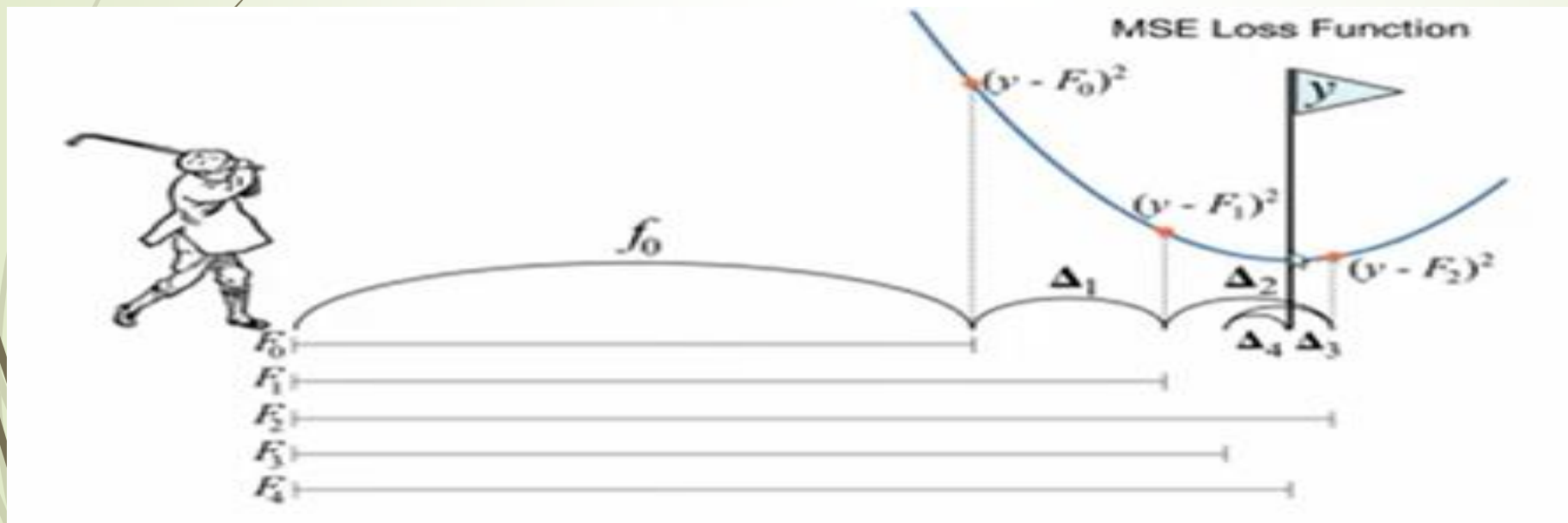


# БУСТИНГ

- Прежде чем говорить о бустинге, упомянем о двух терминах data mining – сильной и слабой моделях. Сильной моделью называется та модель, которая допускает минимальное количество ошибок классификации. Слабая же модель, напротив, допускает множество ошибок – то есть не является точной (либо теряет в надежности). Так вот, бустингом (от англ. boosting – усиление) называется метод, направленный на превращение слабых моделей в сильные путем построения ансамбля классификаторов.
- При бустинге происходит последовательное обучение классификаторов. Таким образом, обучающий набор данных на каждом последующем шаге зависит от точности прогнозирования предыдущего базового классификатора. Первый алгоритм Boost1, например, применял три базовых классификатора. При этом первый классификатор обучался на всем наборе данных, второй на выборке примеров, а третий – на наборе тех данных, где результаты прогнозирования первых двух классификаторов разошлись. Современная модификация первого алгоритма подразумевает использование неограниченного количества классификаторов, каждый из которых обучается на одном наборе примеров, поочередно применяя их на различных шагах.

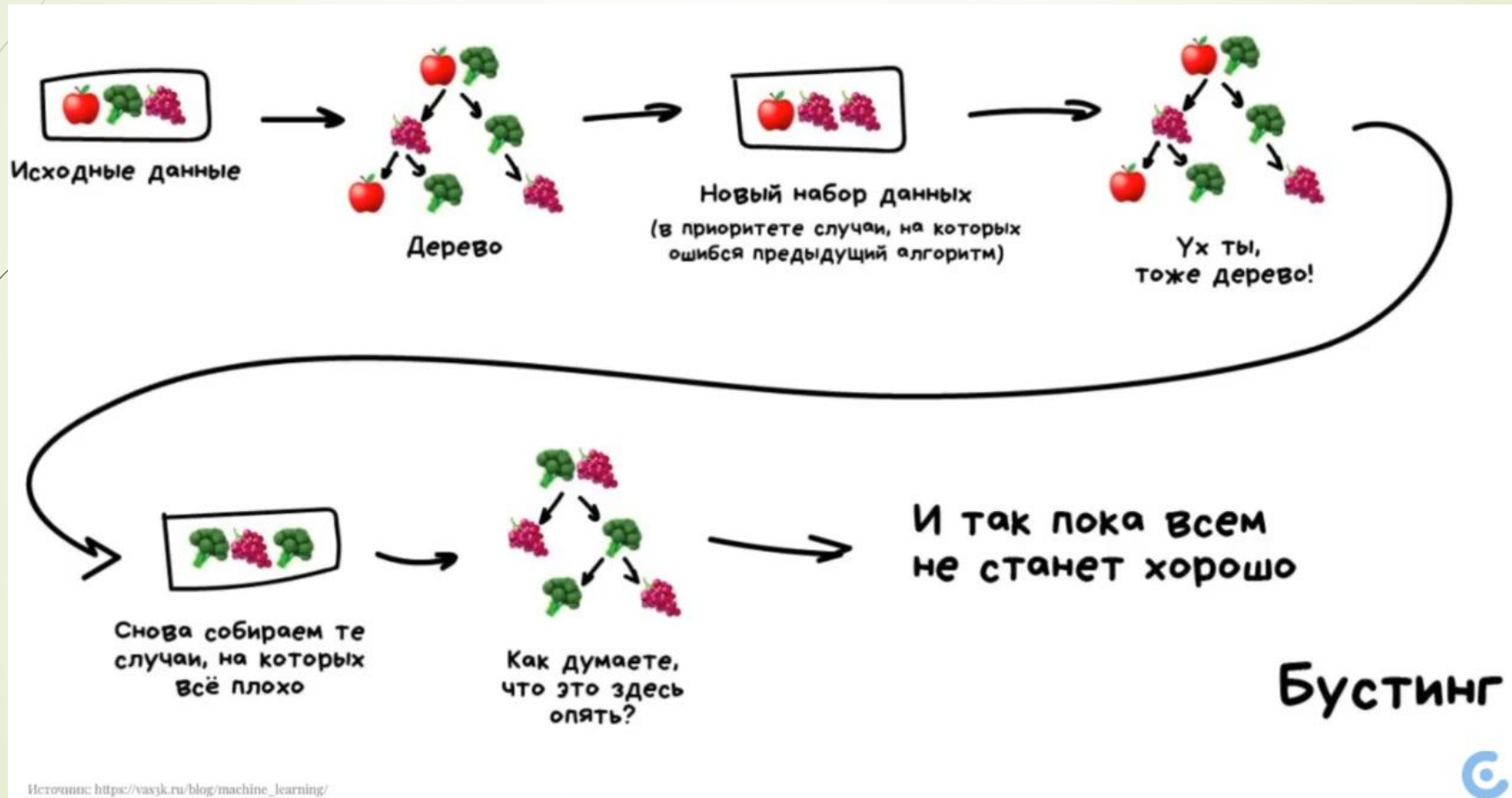
# Boosting

- Последовательное приближение модели.
- Построение уточняющих классификаторов.
- Градиентный бустинг





# Последовательное обучение классификаторов

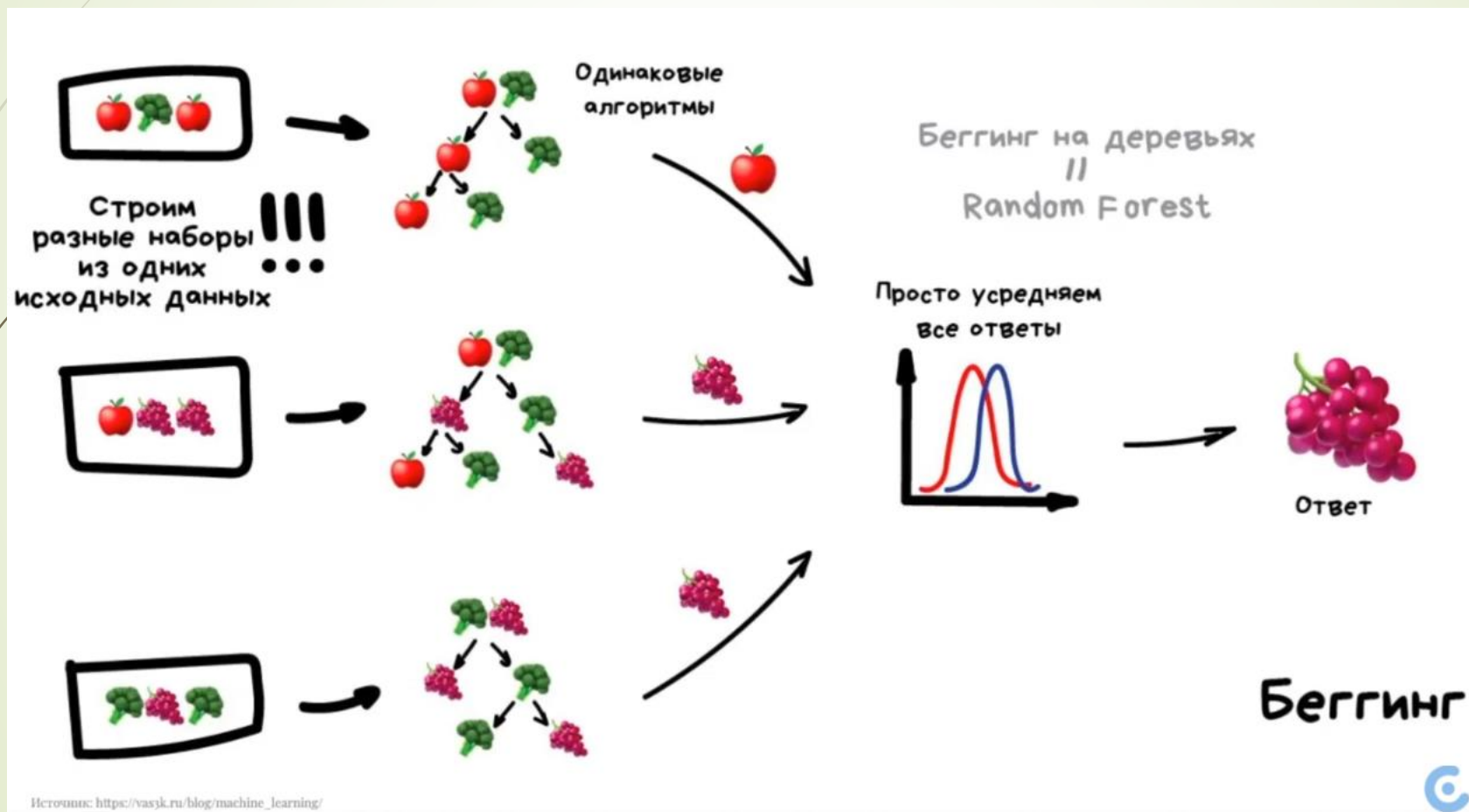




# БЭГГИНГ

- В отличие от предыдущего метода, бэггинг (bootstrap aggregating) использует параллельное обучение базовых классификаторов. В ходе бэггинга происходит следующее:
  1. Из множества исходных данных случайным образом отбирается несколько подмножеств, содержащих количество примеров, соответствующее количеству примеров исходного множества.
  2. Поскольку отбор осуществляется случайным образом, то набор примеров всегда будет разным: некоторые примеры попадут в несколько подмножеств, а некоторые не попадут ни в одно.
  3. На основе каждой выборки строится классификатор.
  4. Выводы классификаторов агрегируются (путем голосования или усреднения).
- Как и при бустинге, ожидается, что результат прогноза агрегированного классификатора будет намного точнее результата прогноза одиночной модели на том же наборе данных.

# Параллельное обучение классификаторов

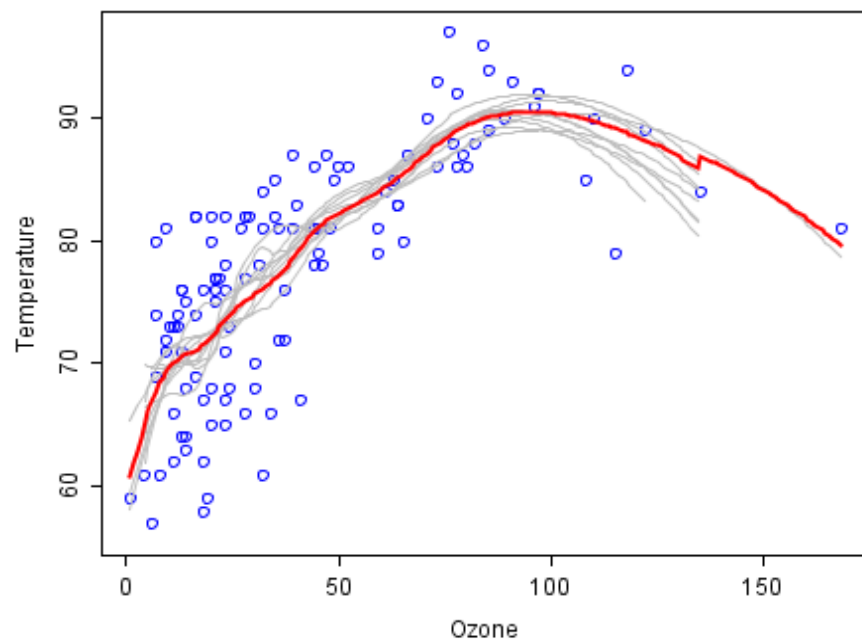


# БЭГГИНГ

Для иллюстрации основных принципов бэггинга ниже приведён анализ связи между [озоном](#) и температурой (данные взяты из книги [Руссёва](#) <sup>русск.</sup><sub>(англ.)</sub> и Леруа <sup>[6]</sup>. Анализ осуществлён на языке программирования R).

Связь между температурой и озоном в этом наборе данных, очевидно, нелинейна. Чтобы описать эту связь, использовались сглаживатели [LOESS](#) <sup>русск.</sup><sub>(англ.)</sub> (с полосой пропускания 0,5). Вместо построения единого сглаживателя из всего набора данных извлечено 100 выборок [бутстрэпов](#) данных. Каждая выборка отличается от исходного набора данных, но они, всё же, совпадают по распределению и дисперсии. Для каждой бутстрэп-выборки применялся сглаживатель LOESS. Затем сделано предсказание по данным на основе этих 100 сглаживаний. Первые 10 сглаживаний показаны серыми линиями на рисунке ниже. Линии, как видно, очень *волнисты* и страдают перепогонкой данных – результат полосы слишком мал.

Взяв среднее 100 сглаживателей, которые применялись к подмножествам оригинального набора данных, мы получаем сборный предсказатель (красная линия). Ясно, что среднее более устойчиво и не столь подвержено [переобучению](#).





# Random Forest

- Random Subspace Method для обработки не объектов, а признаков.
- Random Patches – обработка и объектов и признаков. Пример Random Forest




# Случайный лес

- Ансамбль классификаторов на основе деревьев принятия решений – многократное обучение на разных множествах, коллективное мнение всегда разумнее.
- Ансамбль – алгоритм, который состоит из нескольких разноресовых алгоритмов машинного обучения. Много классификаторов принимают лучшее решение.
- $F(x)=d(h_1(x),h_2(x),\dots,h_n(x))$



# Основа ансамблей





# Блендинг

- **Блендинг (блендинг)** - "простая" композиция алгоритмов. Обычно под этим понимаю обобщенное среднее ответов нескольких алгоритмов.
- *Примеры: среднее арифметическое, среднее геометрическое, среднее рангов и т.д.*



# Голосование



- Теперь рассмотрим второй путь, когда имеется несколько моделей, проводящих голосование.
- Попробуем решить несколькими методами следующую задачу. На вход системы поступают тексты, необходимо определять язык текста. Очевидно, задача является задачей многоклассовой классификации. У нас есть несколько путей ее решения.
- Построить один многоклассовый классификатор, обучив его на всей обучающей выборке.
- Использовать метод классификации, который само по себе ансамбль, но за счет интерфейса смотрится монолитно.
- Обучить несколько многоклассовых классификаторов на разных подвыборках, организовать голосование между ними:
  - выбирать класс, за который проголосовало большинство, возвращать его;
  - нормировать количество голосов, отданных за каждый из классов, возвращать вероятность ответа.



# Пример

➤ [colab.research.google.com/drive/1\\_XHTfz0OSeVwrX4h\\_PdJ8\\_7BaWp2L5np?usp=sharing#scrollTo=cVDxtCHuhRR3](https://colab.research.google.com/drive/1_XHTfz0OSeVwrX4h_PdJ8_7BaWp2L5np?usp=sharing#scrollTo=cVDxtCHuhRR3)



# Контрольная работа

- Классифицировать произвольные данные 5 различными классификаторами. Выбрать лучший метод. Свой выбор обосновать.
- 