

Системы и технологии интеллектуальной обработки данных

Лекция 5 Тема: Классификация

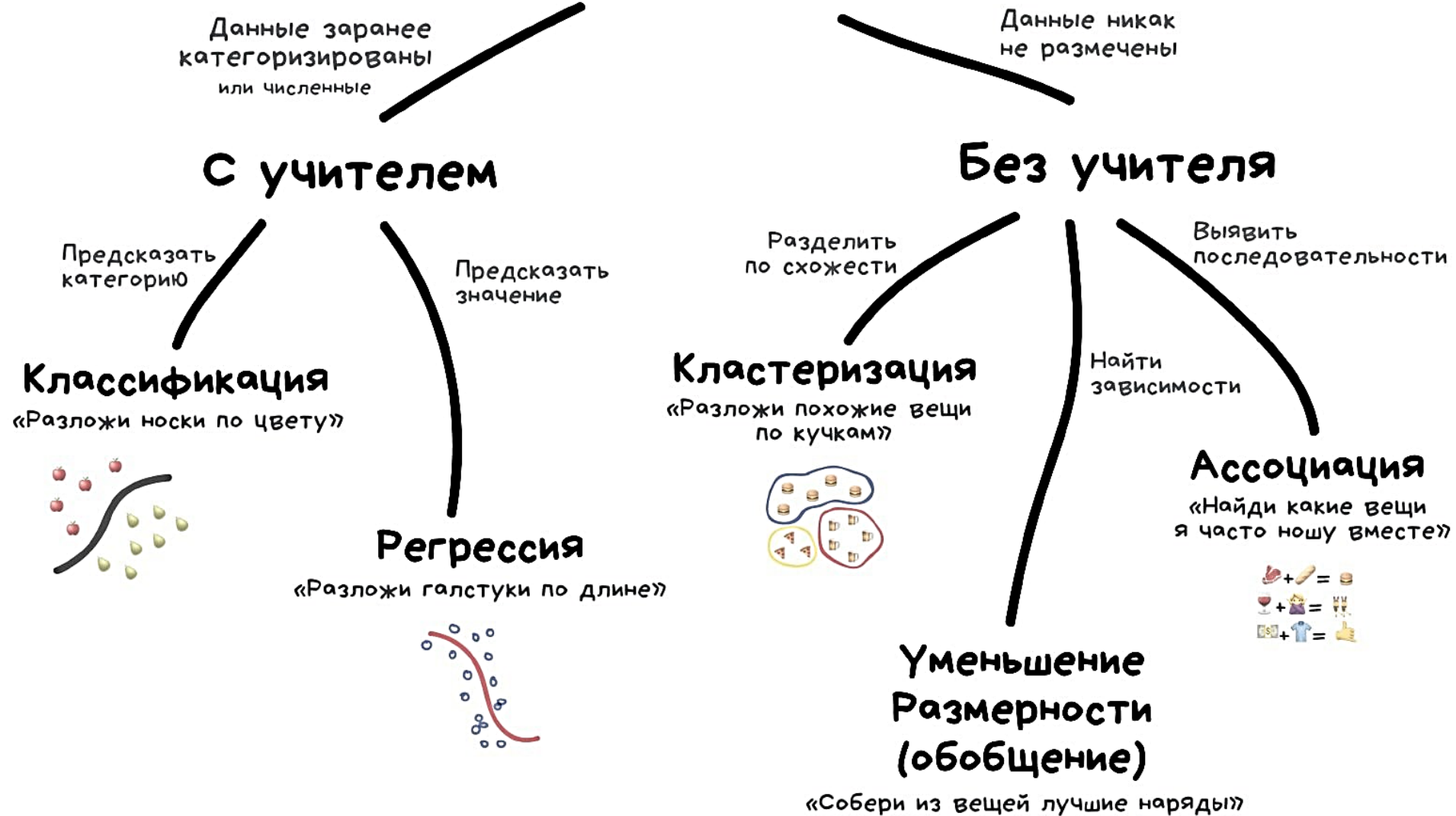
В лекции использованы материалы статей

<https://loginom.ru/blog/decision-tree-p1>

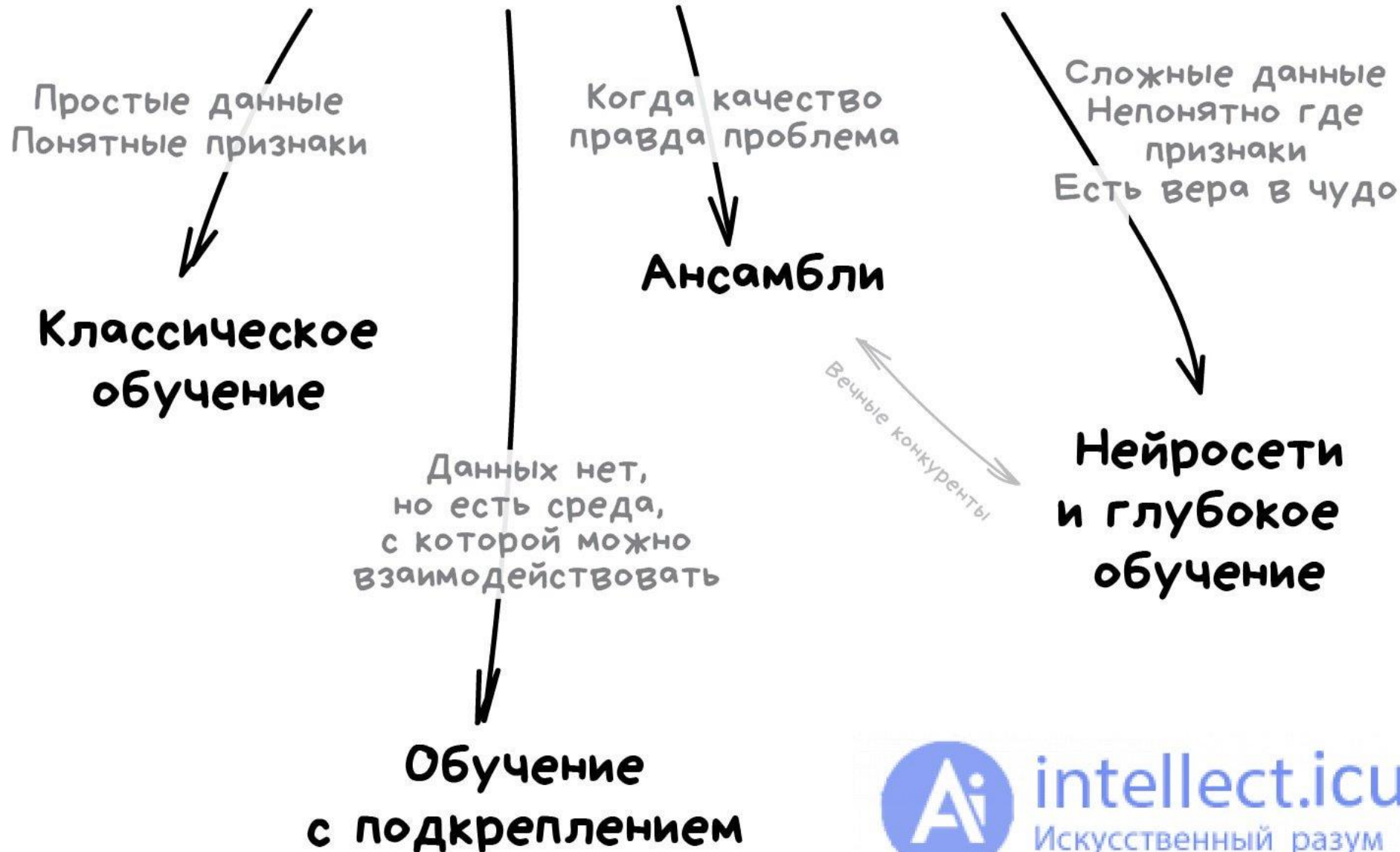
<https://habr.com/ru/company/ods/blog/322534/>

<https://www.bigdataschool.ru/blog/machine-learning-confusion-matrix.html>

Классическое Обучение



Основные виды машинного обучения



intellect.icu

Искусственный разум

Классификация — один из разделов машинного обучения, посвященный решению следующей задачи.

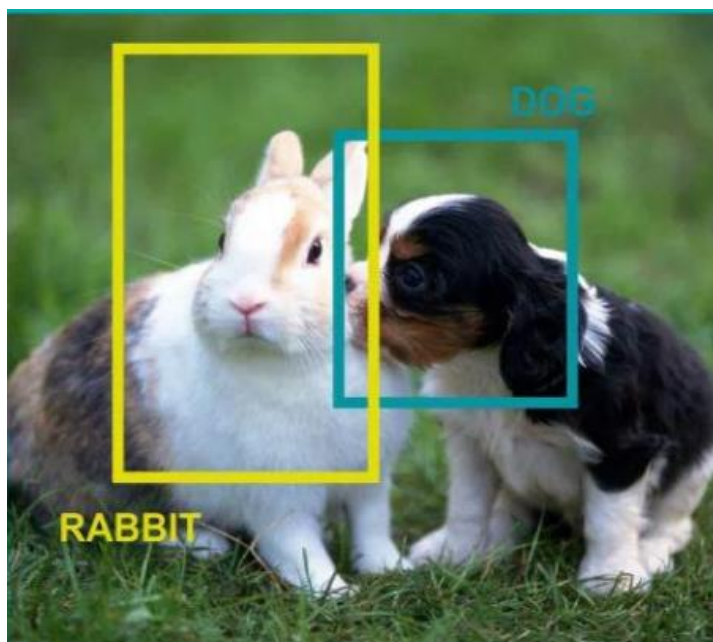
Имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из множества.

Классифицировать объект — значит, указать номер (или наименование класса), к которому относится данный объект.

Классификация иногда разделяется на **бинарную** классификацию (binary classification), которая является частным случаем разделения на два класса, и **мультиклассовую** классификацию (multiclass classification), когда в классификации участвует более двух классов.

В машинном обучении **задача классификации относится к разделу обучения с учителем.**

Обучение с учителем подразумевает, что данные в обучающей выборке должны быть размечены (в случае распознавания изображений, символов) или каждому набору признаков должны соответствовать метки (классы).

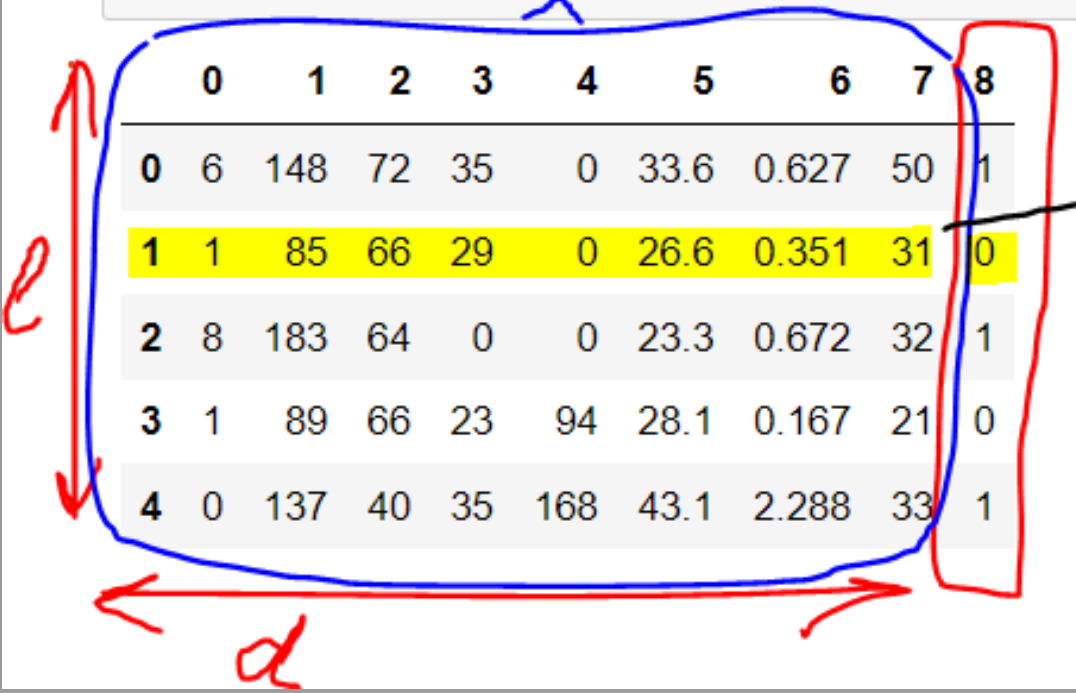


PassengerId	Survived	Pclass	Name	Sex	Age
1	0	3	Braund, Mr. Owen Harris	male	22.0
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
3	1	3	Heikkinen, Miss. Laina	female	26.0
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
5	0	3	Allen, Mr. William Henry	male	35.0

Существует также **обучение без учителя**, когда разделение объектов обучающей выборки на классы не задаётся, и требуется классифицировать объекты только на основе их сходства друг с другом. В этом случае принято говорить о задачах кластеризации.

6. Диабет родословной.
7. Возраст (годы).
8. Переменная класса (0 или 1) - выходная переменная

```
: data = pd.read_csv("data/indians-diabetes.csv", header=None)  
data.head()
```



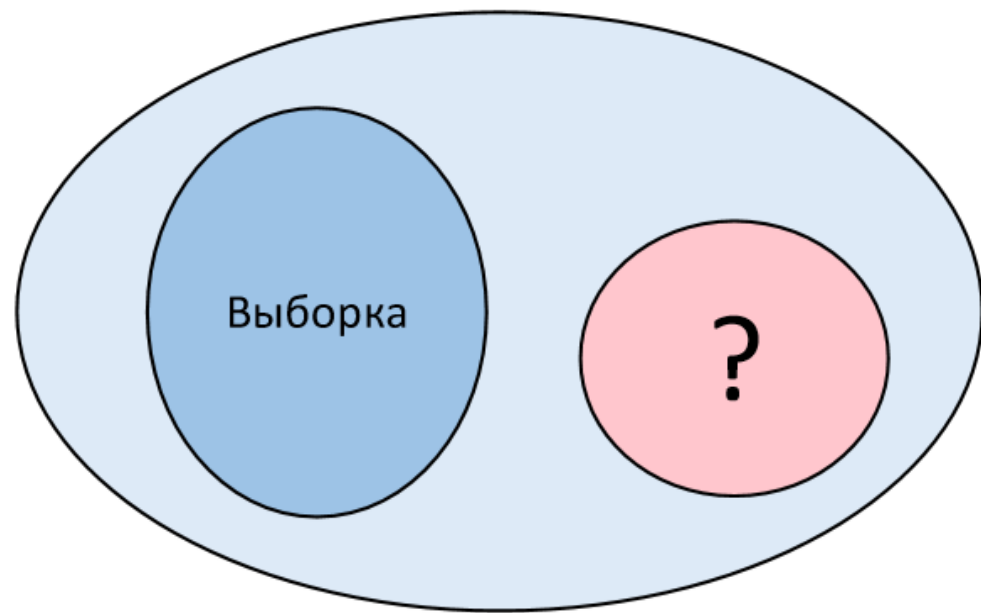
	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

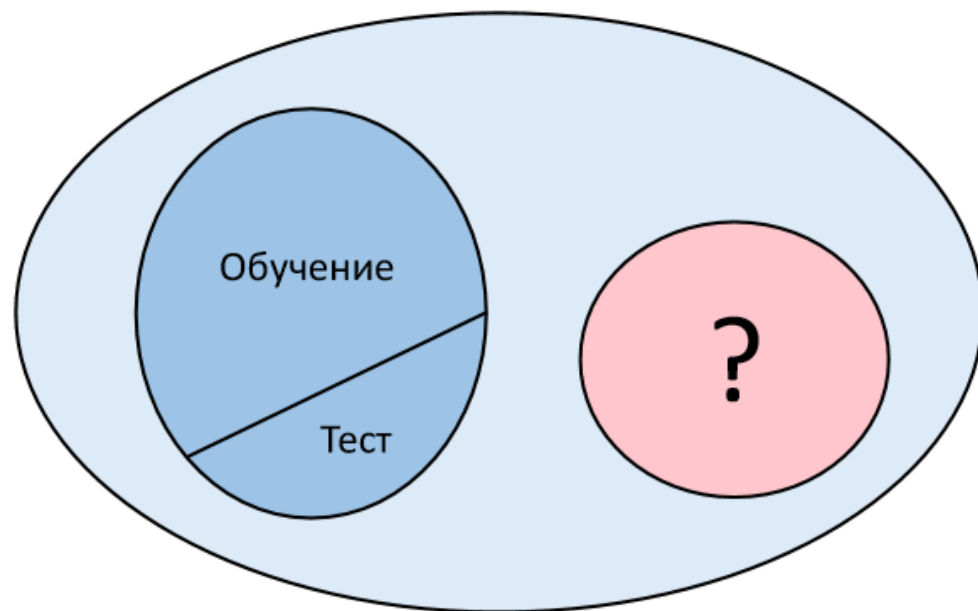
объект

Дано: численные или
категоризированные
данные

x- матрица объектов-
признаков,
y- вектор меток (дискретные
значения)

Нужно: получить модель
(алгоритм), ставящую в
соответствие каждому
новому объекту одну метку
(класс)



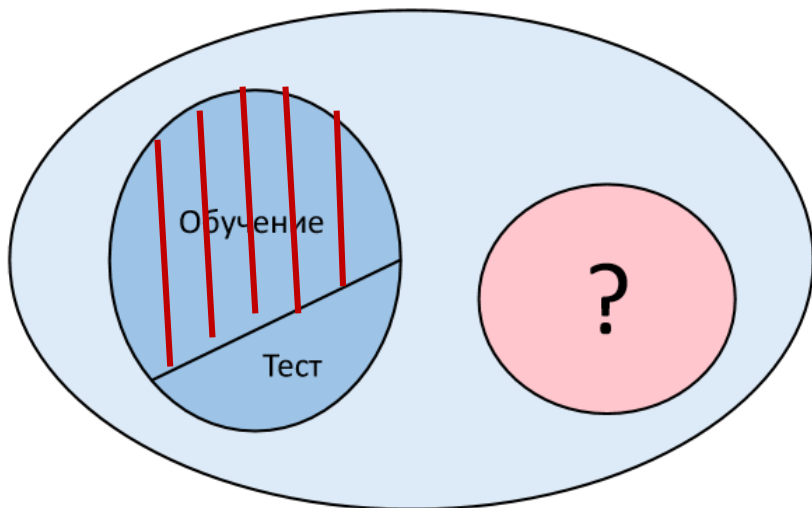


На практике все имеющиеся данные разбивают на обучающую и тестовую выборки (70% и 30%).

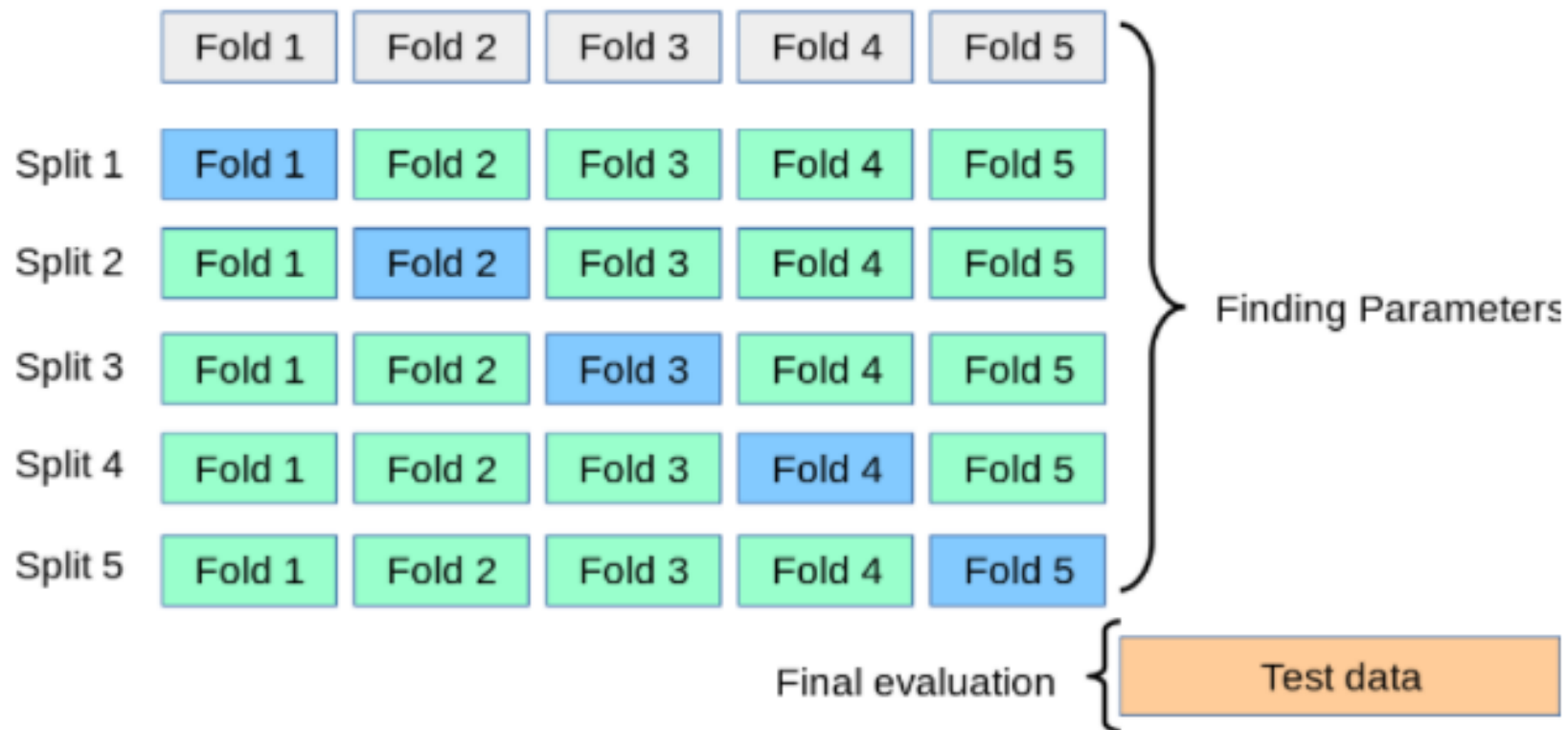
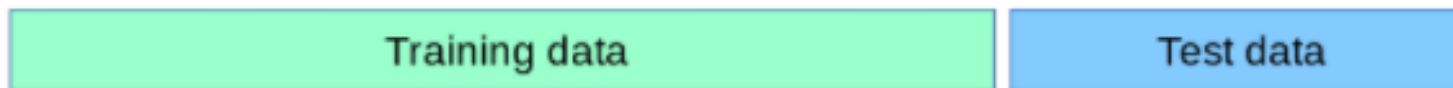
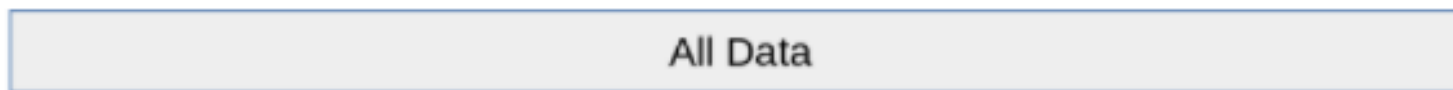
Обучение производится с использованием обучающей выборки, а оценка качества предсказания на основе данных тестовой выборки.

Какие могут быть ошибки при таком разбиении (например, выборка была отсортирована по какому либо признаку)

Оправдано, если очень много данных



На практике: выборку разбивают на обучающую и тестовую, а затем к обучающей применяют метод кросс валидации:



Переобучение и недообучение модели

Если модель может выдавать точные прогнозы на ранее не встречавшихся данных, мы говорим, что модель обладает **способностью обобщать** (generalize) результат на тестовые данные.

Необходимо построить модель, которая будет обладать максимальной обобщающей способностью.

Если обучающий и тестовый наборы имеют много общего между собой, можно ожидать, что модель будет точной и на тестовом наборе. Однако в некоторых случаях этого не происходит.

Чем сложнее модель, тем лучше она будет работать на обучающих данных. Однако, если наша модель становится слишком сложной, мы начинаем уделять слишком много внимания каждой отдельной точке данных в нашем обучающем наборе, и эта модель не будет хорошо обобщать результат на новые данные.

Существует оптимальная точка, которая позволяет получить наилучшую обобщающую способность.

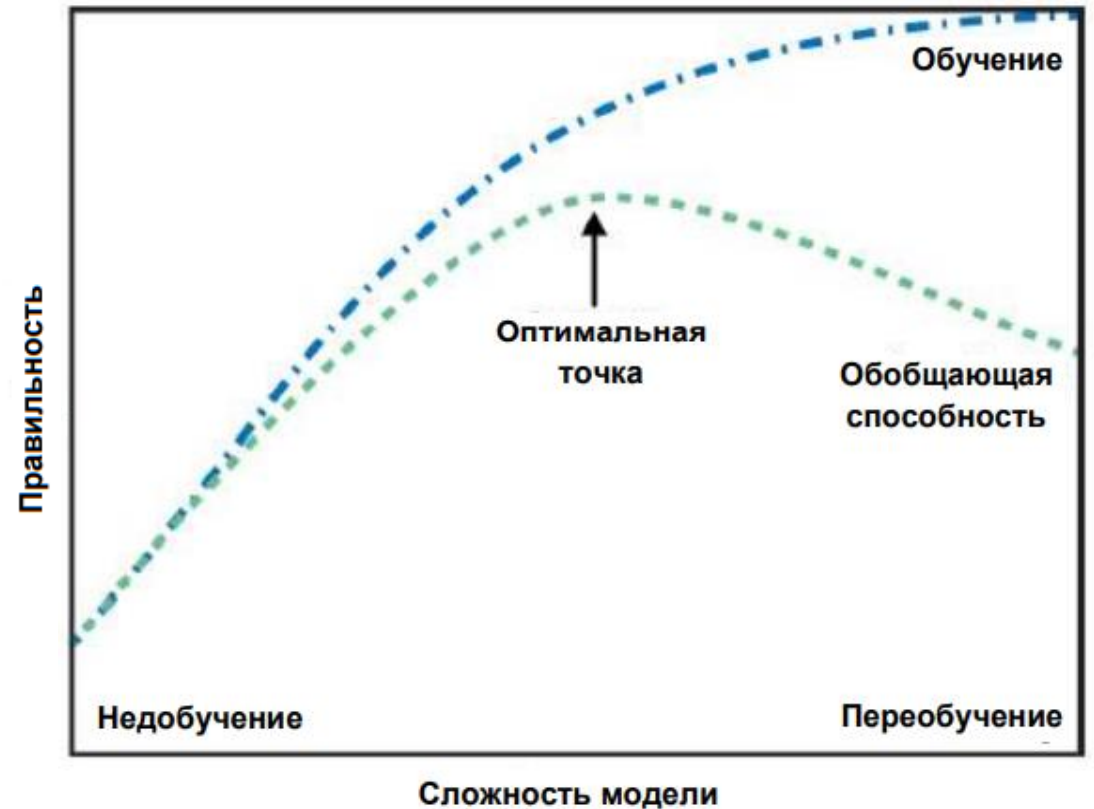


Рис. Компромисс между сложностью модели и правильностью на обучающей и тестовой выборке

Деревья решений (Decision Tree, DT)

Деревья решений являются одним из наиболее эффективных инструментов интеллектуального анализа данных и предсказательной аналитики, которые **позволяют решать задачи классификации и регрессии**.

Они представляют собой иерархические древовидные структуры, состоящие из решающих правил вида «Если ..., то ...». Правила автоматически генерируются в процессе обучения на обучающем множестве.

В обучающем множестве для примеров **должно быть задано целевое значение**, т.к. деревья решений являются моделями, строящимися на основе обучения с учителем. При этом, если **целевая переменная дискретная (метка класса)**, то модель называют **деревом классификации**, а если непрерывная, то деревом **регрессии**.

Метод деревьев решений для задачи классификации состоит в том, чтобы осуществлять процесс деления исходных данных на группы, пока не будут получены однородные (или почти однородные) их множества.

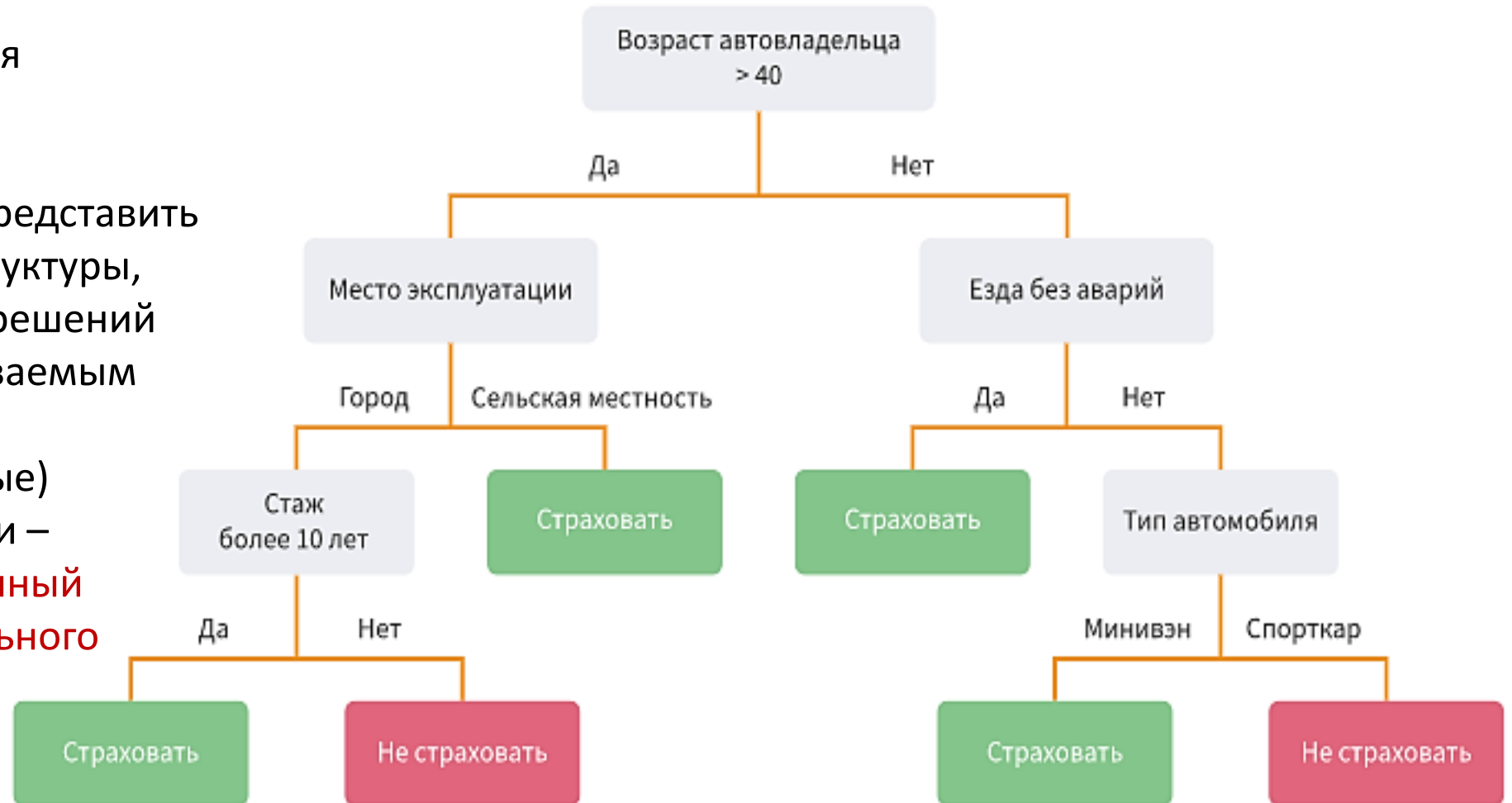
Совокупность правил, которые дают такое разбиение (англ.: partition), позволят затем делать прогноз (т.е. определять наиболее вероятный номер класса) для новых данных.

Структура дерева решений

Дерево решений –это модель, представляющая собой совокупность правил для принятия решений.

Графически её можно представить в виде древовидной структуры, где моменты принятия решений соответствуют так называемым узлам.

Конечные (терминальные) узлы называют листьями – **каждый лист – это конечный результат последовательного принятия решений.**



Задача: разложить сценарии фильмов по трём ящикам:

- Популярные (англ.: «mainstream hits»);
- Не популярные у зрителей, но получившие высокую оценку критиков;
- Не имеющие успеха.

Есть тестовая выборка из 30 фильмов

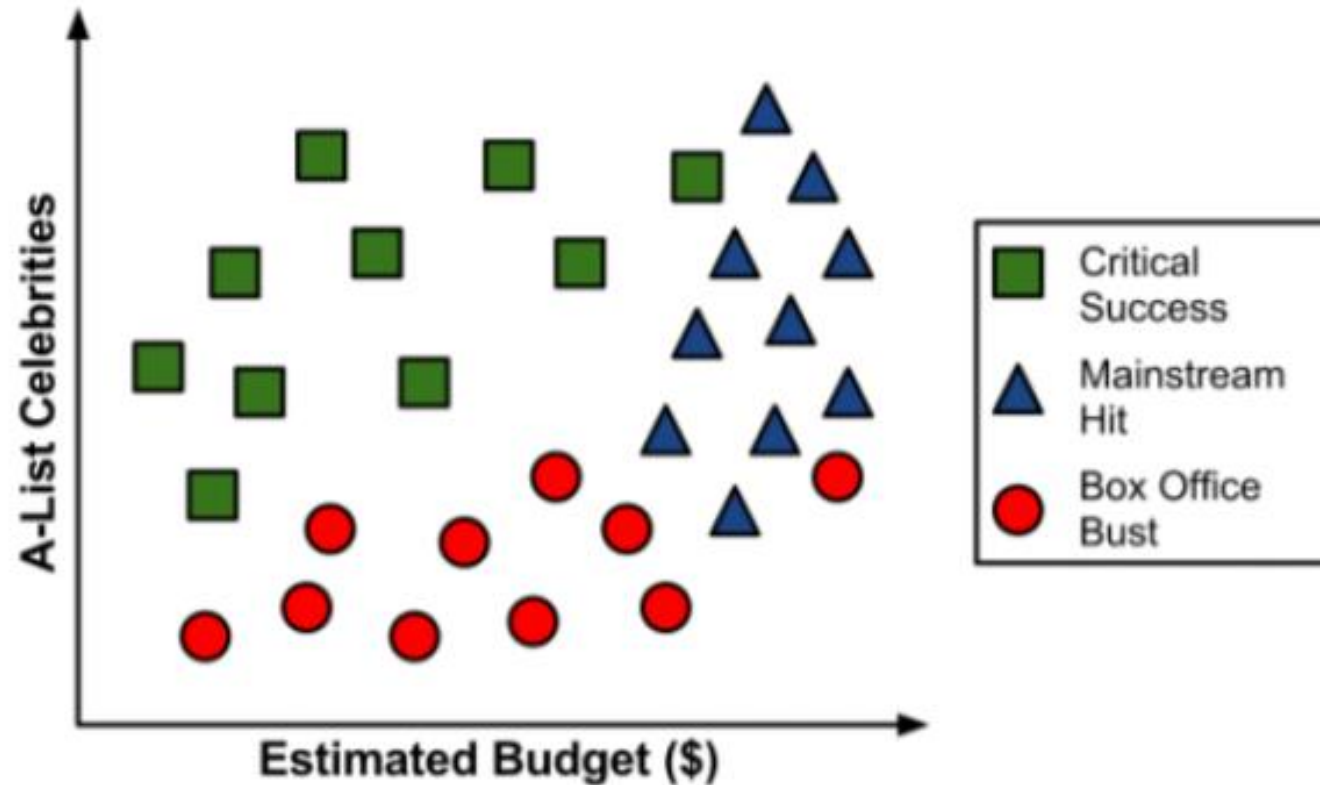


Рис.1 Зависимость количества звёзд, снимавшихся в фильме, от его бюджета

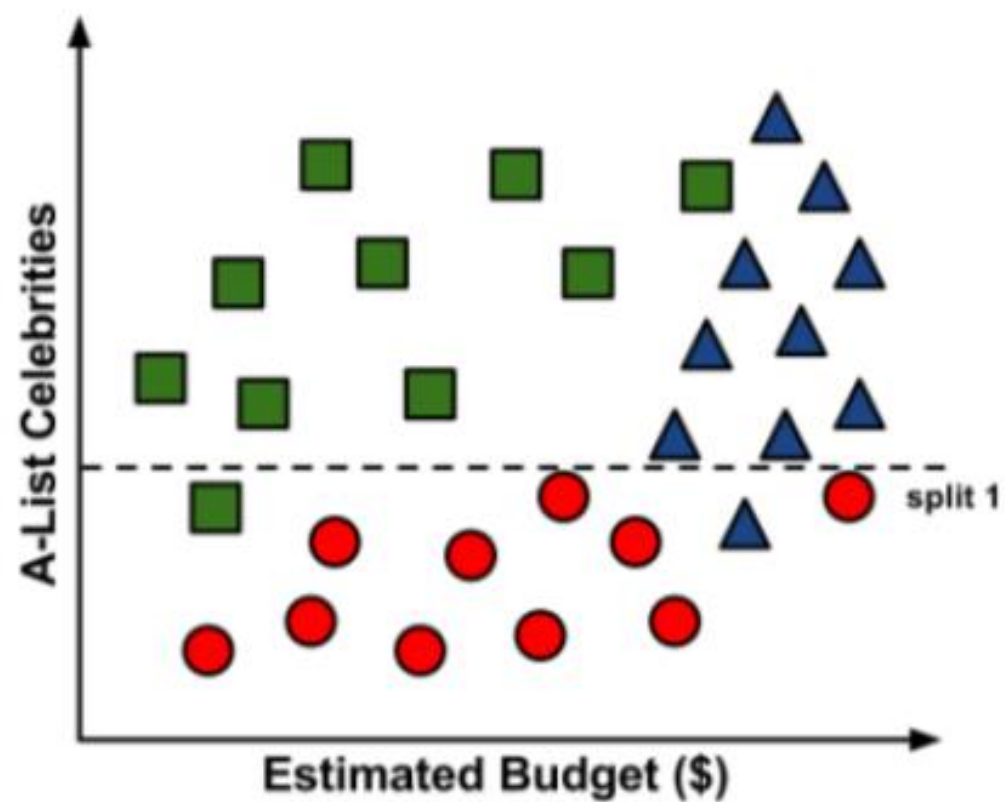


Рис.2 Разбиение множества киносценариев по признаку количества занятых звёзд

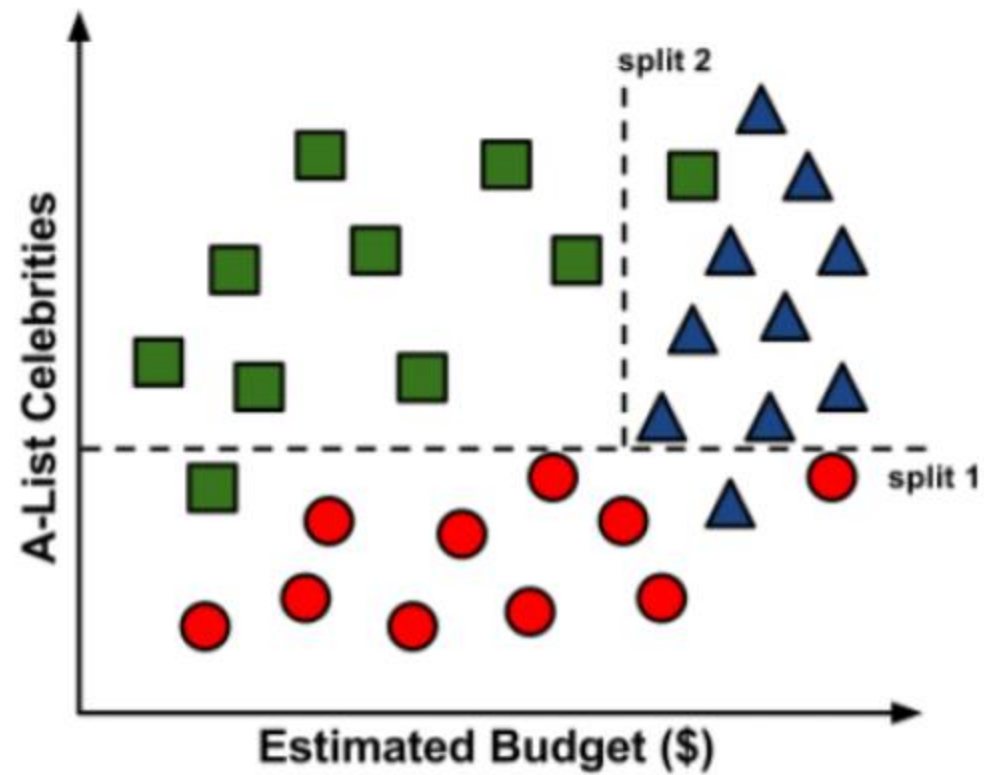
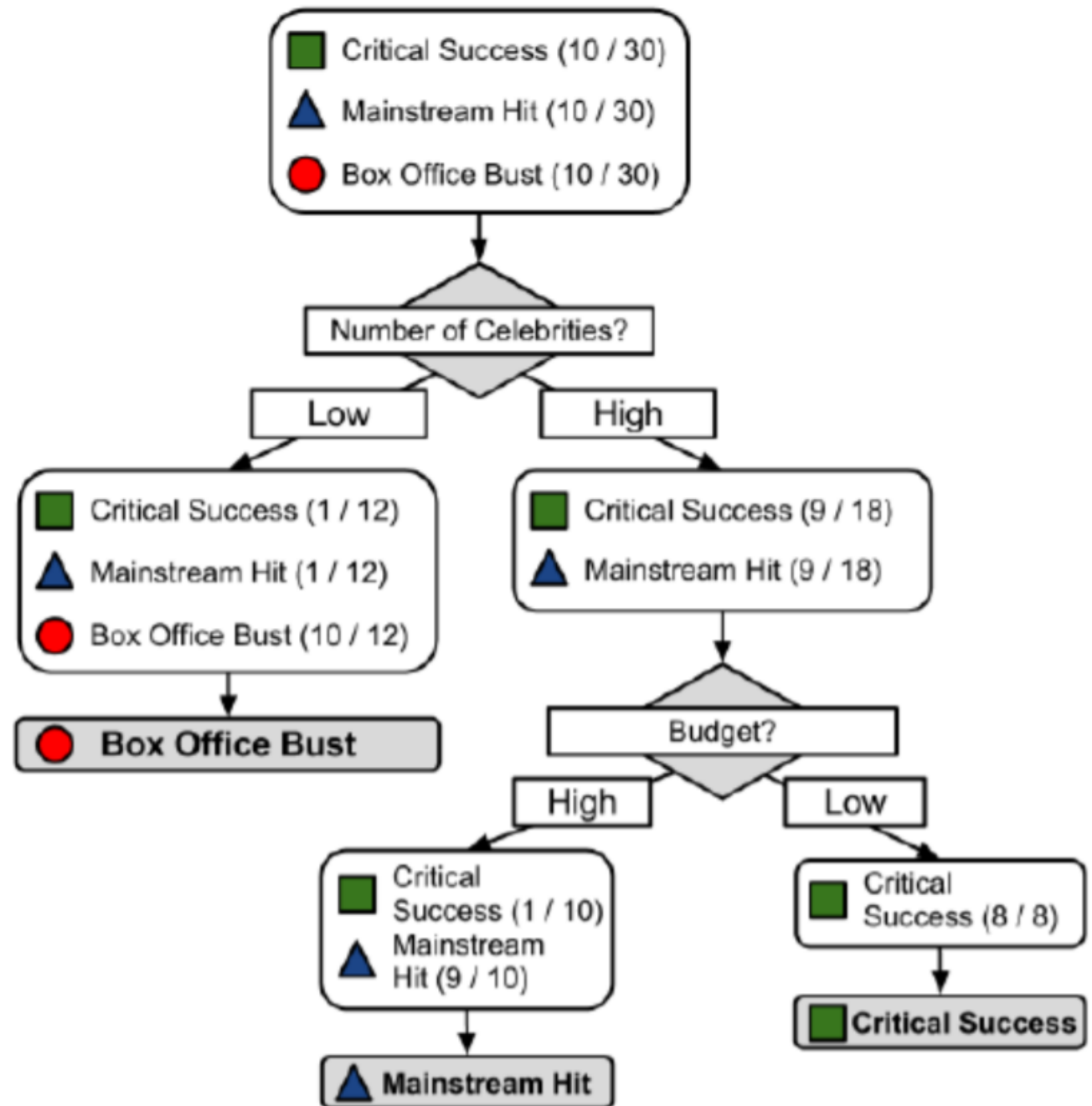


Рис.3 Разбиение группы киносценариев с большим количеством занятых звёзд по признаку размера бюджета

В нашем примере мы условно делим число задействованных в фильме звёзд по принципу «много» – «мало», и аналогично различаем малобюджетные и высокобюджетные. Соответствующее дерево решений показано на рис.

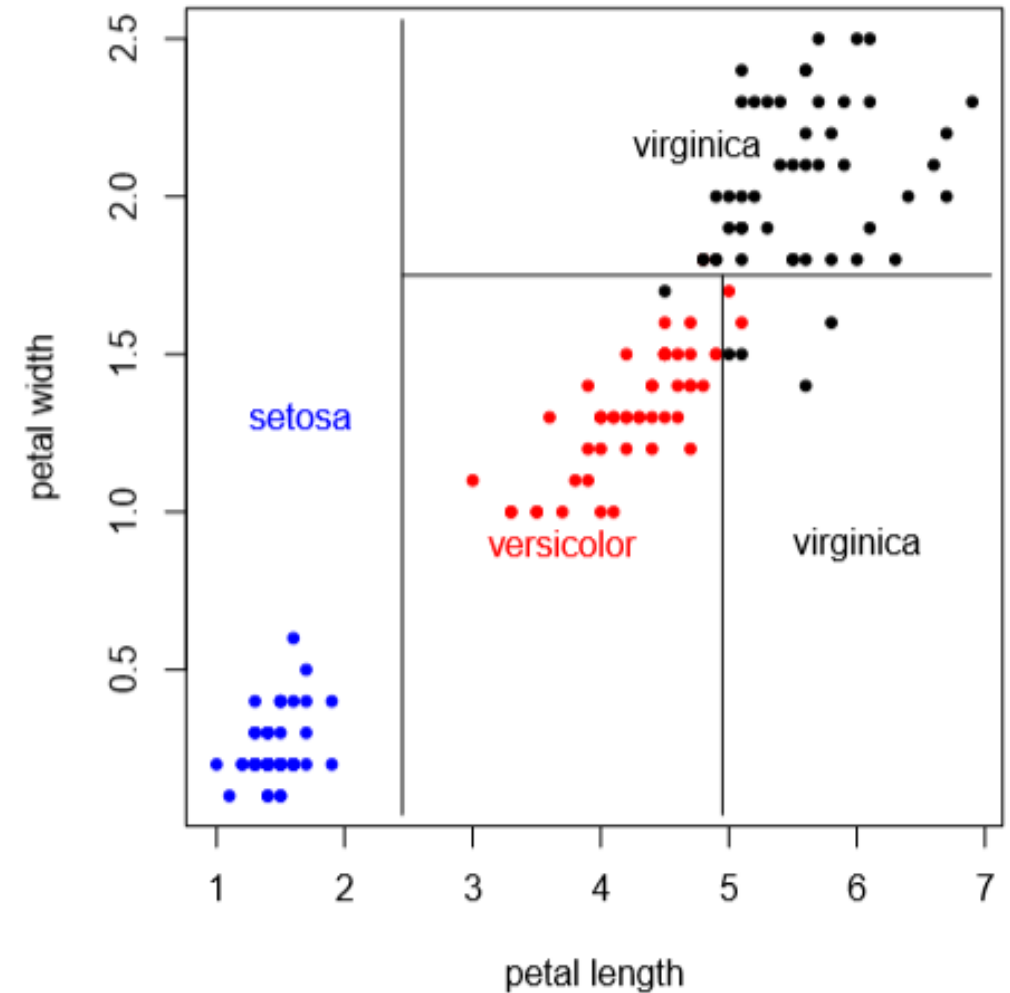
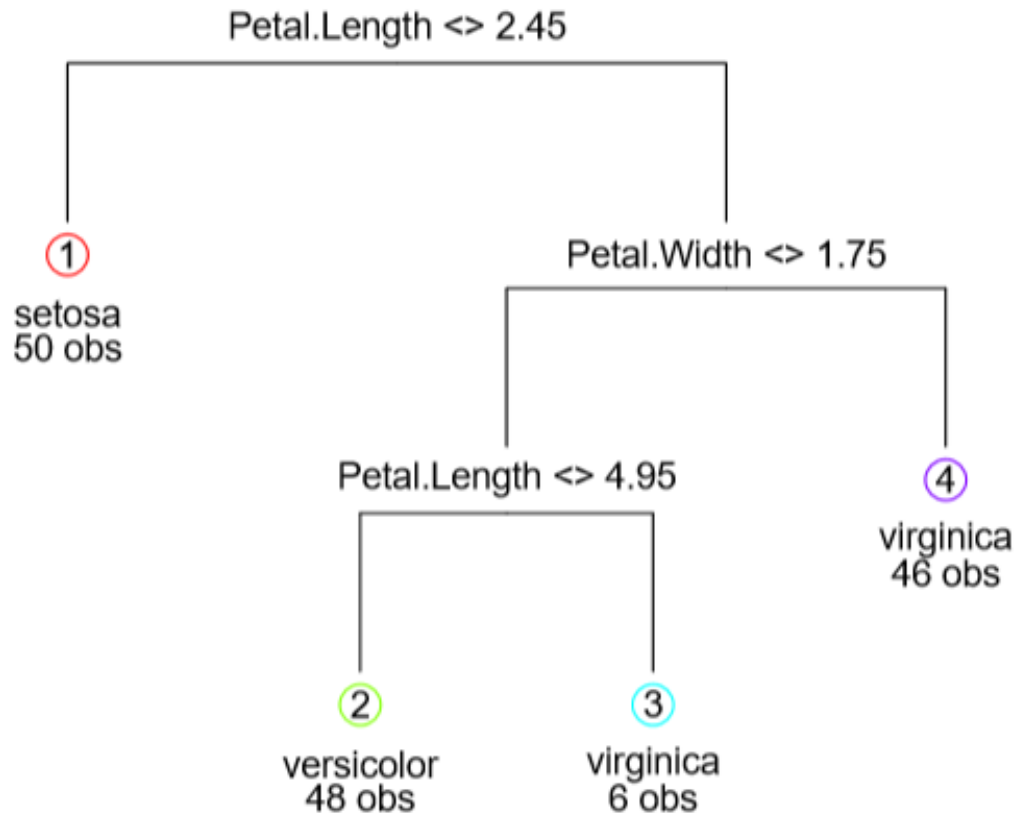
Ограничим ветвление дерева – например, когда каждая группа хотя бы на **80% будет состоять из элементов одного и того же класса**. (В нашем примере в первой группе все зелёные квадраты; во второй группе из 10 элементов 1 квадрат и 9 треугольников (90%); в третьей группе 12 элементов, из них 10 одинаковых – красных кружков, – т.е. показатель однородности = $10/12 = 83,33\ldots\%$).



Пример: деревья решений в задаче классификации цветов ириса

Задача классификации цветов ириса (Fisher, 1936).

x_1, x_2 — длина и ширина чашелистика.



Процесс построения дерева решений

Процесс построения деревьев решений заключается в последовательном, рекурсивном разбиении обучающего множества на подмножества с применением решающих правил в узлах. Эта стратегия рекурсивного разбиения также называется «Разделяй и властвуй».

Процесс разбиения продолжается до тех пор, пока все узлы в конце всех ветвей не будут объявлены листьями. **Объявление узла листом может произойти** естественным образом (**когда он будет содержать единственный объект, или объекты только одного класса**), или **по достижении некоторого условия остановки**, задаваемого пользователем (например, достижение заданного показателя однородности в листьях или максимальная глубина дерева).

Основные этапы построения

- ✓ Выбор признака, по которому будет производиться разбиение в данном узле.
- ✓ Выбор критерия остановки обучения.
- ✓ Оценка точности построенного дерева.

Выбор признака

Какой признак выбрать первым?

Здесь можно вспомнить игру "20 вопросов", которая часто упоминается во введении в деревья решений.

Один человек загадывает знаменитость, а второй пытается отгадать, задавая только вопросы, на которые можно ответить "Да" или "Нет". Какой вопрос отгадывающий задаст первым делом? Конечно, такой, который сильнее всего уменьшит количество оставшихся вариантов. К примеру, вопрос "Это женщина?" отсечет уже около половины знаменитостей. То есть, признак "пол" намного лучше разделяет выборку людей, чем признак "это Джонни Депп", "национальность-испанец" или "любит футбол".

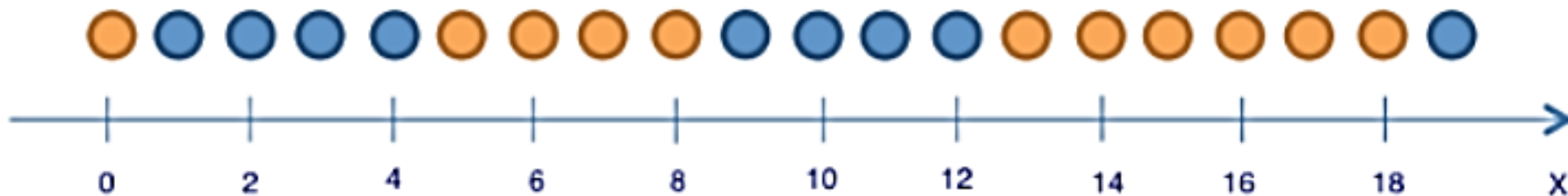
Энтропия определяется для системы с N возможными состояниями следующим образом:

$$S = - \sum_{i=1}^N p_i \log_2 p_i,$$

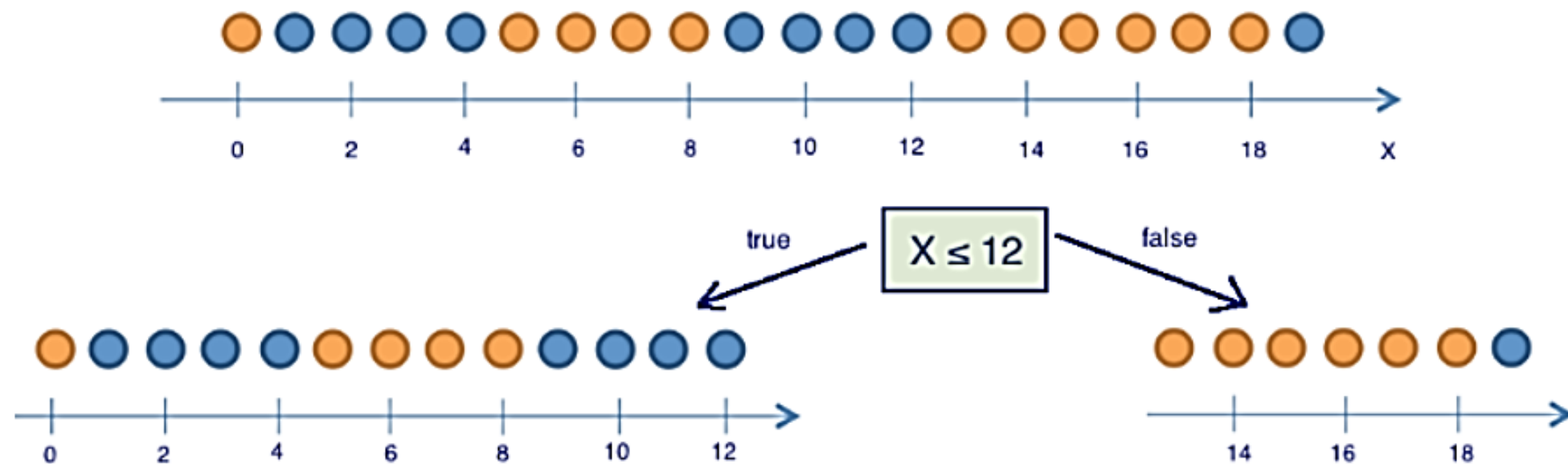
где p_i – вероятности нахождения системы в i -ом состоянии.

Энтропия соответствует степени хаоса в системе. Чем выше энтропия, тем менее упорядочена система и наоборот. Это понятие поможет формализовать "эффективное разделение выборки".

Для иллюстрации того, как энтропия поможет определить хорошие признаки для построения дерева, приведем игрушечный пример. Будем предсказывать цвет шарика по его координате.

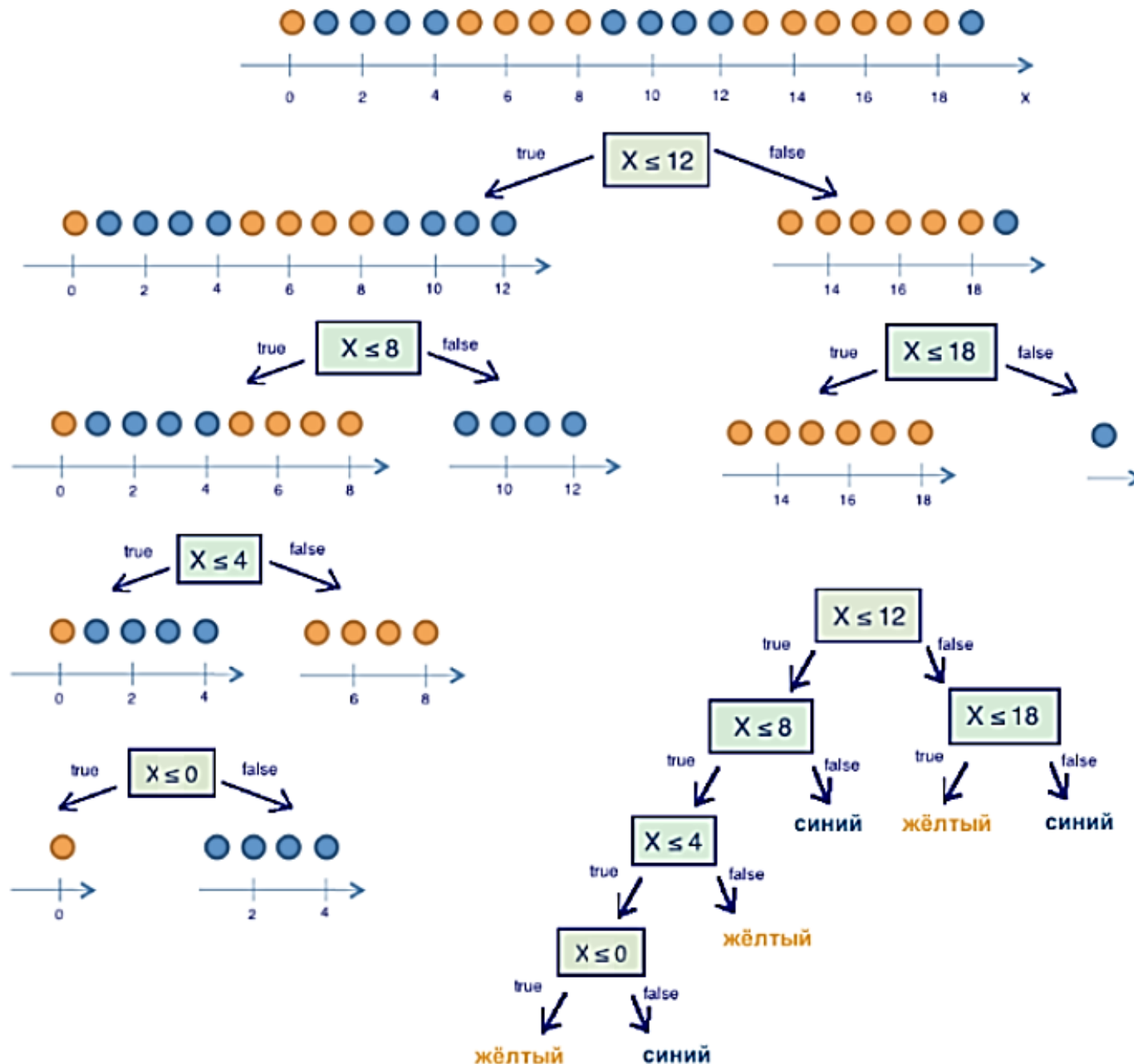


Здесь 9 синих шариков и 11 желтых. Если мы наудачу вытащили шарик, то он с вероятностью $p_1 = \frac{9}{20}$ будет синим и с вероятностью $p_2 = \frac{11}{20}$ – желтым. Значит, энтропия состояния $S_0 = -\frac{9}{20} \log_2 \frac{9}{20} - \frac{11}{20} \log_2 \frac{11}{20} \approx 1$. Само это значение пока ни о чем нам не говорит. Теперь посмотрим, как изменится энтропия, если разбить шарики на две группы – с координатой меньше либо равной 12 и больше 12.



В левой группе оказалось 13 шаров, из которых 8 синих и 5 желтых. Энтропия этой группы равна $S_1 = -\frac{5}{13} \log_2 \frac{5}{13} - \frac{8}{13} \log_2 \frac{8}{13} \approx 0.96$. В правой группе оказалось 7 шаров, из которых 1 синий и 6 желтых. Энтропия правой группы равна $S_2 = -\frac{1}{7} \log_2 \frac{1}{7} - \frac{6}{7} \log_2 \frac{6}{7} \approx 0.6$. Как видим, энтропия уменьшилась в обеих группах по сравнению с начальным состоянием, хоть в левой и не сильно.

Получается, разделив шарики на две группы по признаку "координата меньше либо равна 12", мы уже получили более упорядоченную систему, чем в начале. Продолжим деление шариков на группы до тех пор, пока в каждой группе шарики не будут одного цвета.



Вверху дерева – общие правила
Внизу - частные

Очевидно, энтропия группы с шариками одного цвета равна 0 ($\log_2 1 = 0$), что соответствует представлению, что группа шариков одного цвета – упорядоченная.

Переобученное дерево может иметь по одному объекту в листьях.

Кроме этого, **переобученные деревья** имеют очень сложную структуру, и поэтому их сложно интерпретировать.

Очевидным решением проблемы является принудительная остановка построения дерева, пока оно не стало переобученным. Для этого разработаны следующие подходы.

- ✓ **Ранняя остановка** — алгоритм будет остановлен, как только будет достигнуто заданное значение некоторого критерия, например процентной доли правильно распознанных примеров. Единственным преимуществом подхода является снижение времени обучения. Главным недостатком является то, что ранняя остановка всегда делается в 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.
- ✓ **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.

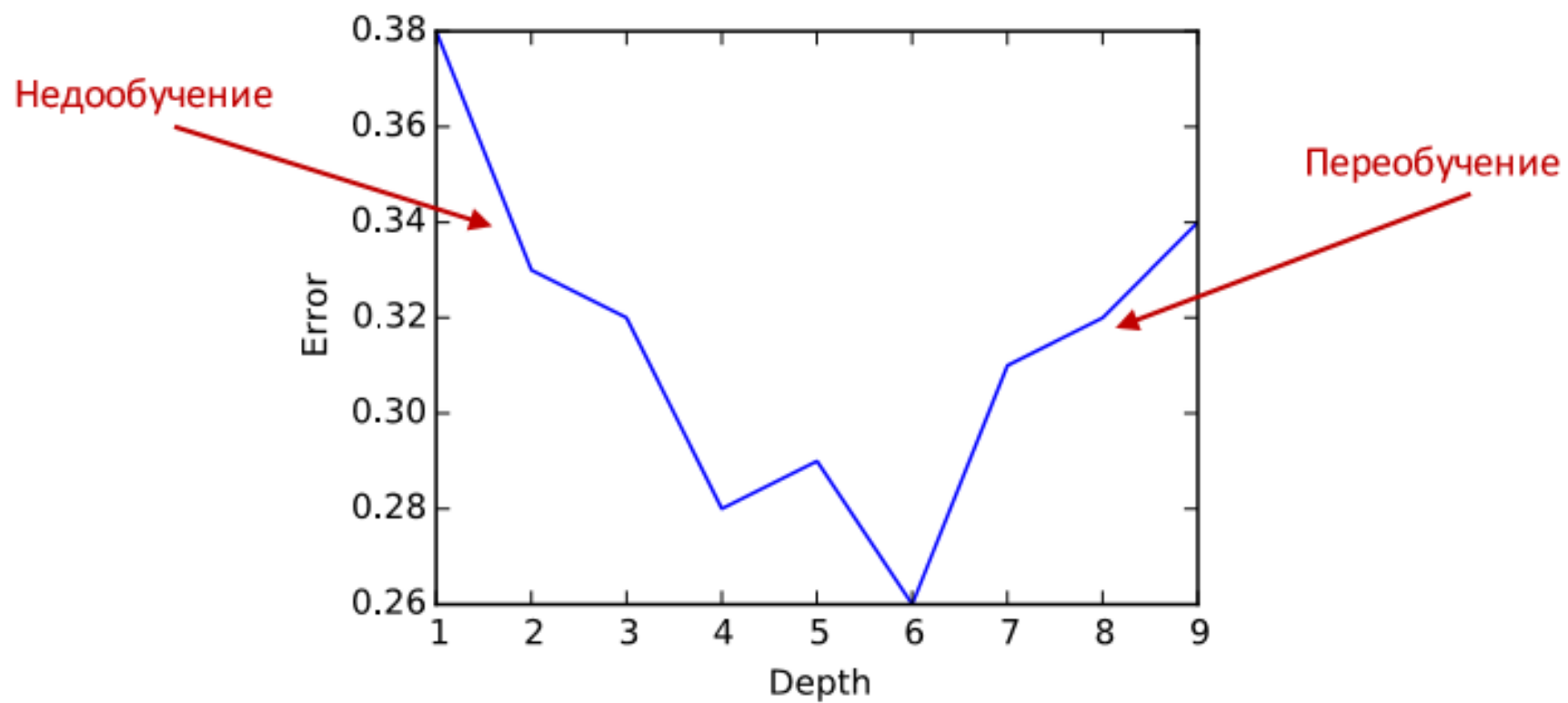
Как найти оптимальную глубину?

На практике можно сравнить результаты деревьев разных глубин.

Например

1- 72% 2-84% **3-86%** 4-76%

Зависимость качества от глубины дерева



Деревья решений являются одним из наиболее наглядных и универсальных алгоритмов обучения.

К достоинствам деревьев решений следует отнести:

- Возможность производить обучение на исходных данных без их дополнительной предобработки (нормализация и т. п.);
- Нечувствительность к монотонным преобразованиям данных;
- Устойчивость к выбросам;
- Поддержка работы с большими выборками;
- Поддержка работы с входными переменными разных типов;
- Возможность интерпретации построенного дерева решений.

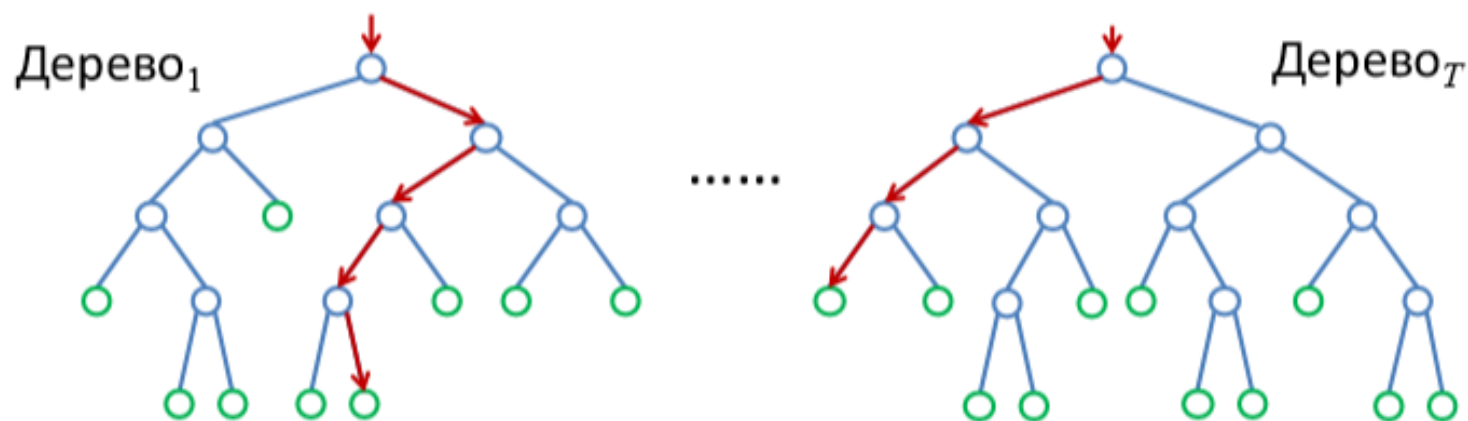
Недостаток:

- Деревья решений не устойчивы даже к небольшим изменениям.
- Проблематичность построения оптимального дерева решений. Построение и поиск такого дерева решений являются NP- полной задачей, сложно разрешимой на практике. Поэтому практическое построение деревьев решений связано с применением эвристических «жадных» алгоритмов, оптимальных только в каждом узле дерева, но не оптимальных для дерева в целом.

Применение деревьев решений

В чистом виде применяются редко. Как правило применяются множества деревьев решений: случайный лес и бустинг.

Случайный лес, а точнее – случайные леса (random forests), является одним из наиболее универсальных и эффективных алгоритмов обучения с учителем, применимым как для задач классификации, так и для задач восстановления регрессии. Идея метода заключается в использовании ансамбля из деревьев решений, которые обучаются независимо друг от друга.



Бустинг — комплекс методов, способствующих повышению точности аналитических моделей.

Матрица ошибок (confusion matrix) используется с целью сопоставления предсказаний и реальности в Data Science. Это таблица с 4 различными комбинациями прогнозируемых и фактических значений. Прогнозируемые значения описываются как положительные и отрицательные, а фактические – как истинные и ложные

Прогноз	Реальность	
	+	-
+	True Positive (истинно-положительное решение): прогноз совпал с реальностью, результат положительный произошел, как и было предсказано ML-моделью	False Positive (ложноположительное решение): ошибка 1-го рода, ML-модель предсказала положительный результат, а на самом деле он отрицательный
-	False Negative (ложноотрицательное решение): ошибка 2-го рода – ML-модель предсказала отрицательный результат, но на самом деле он положительный	True Negative (истинно-отрицательное решение): результат отрицательный, ML-прогноз совпал с реальностью

П р о г н о з	Реальность	
	Правильно	Неправильно
	Правильно	Неправильно
	Правильно	Неправильно
	TP	FP
	FN	TN

Матрица ошибок (confusion matrix)

С математической точки зрения оценить точность ML-модели можно с помощью следующих метрик:

Точность – сколько всего результатов было предсказано верно;

Доля ошибок;

Полнота – сколько истинных результатов было предсказано верно;

F-мера, которая позволяет сравнить 2 модели, одновременно оценив полноту и точность. Здесь используется среднее гармоническое вместо среднего арифметического, сглаживая расчеты за счет исключения экстремальных значений.

В количественном выражении это будет выглядеть так:

P – число истинных результатов, $P = TP + FN$;

N – число ложных результатов, $N = TN + FP$.

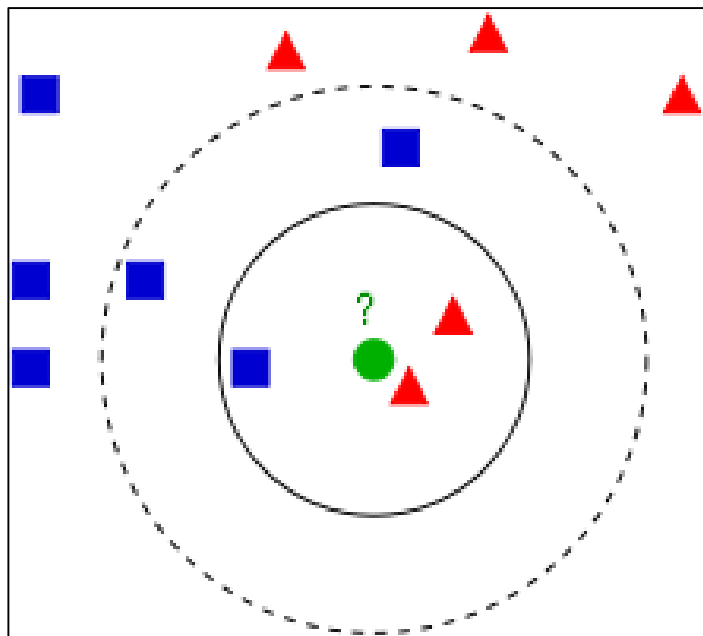
Точность (accuracy):	$\frac{TP+TN}{P+N}$	Доля ошибок (error rate):	$1 - \text{accuracy} = \frac{FP+FN}{P+N}$
FPR (ложная тревога):	$\frac{FP}{N}$	TPR (вероятность обнаружения):	$\frac{TP}{P}$
Точность (precision):	$\frac{TP}{TP+FP}$	Полнота (recall):	$\frac{TP}{P}$
F-мера:	$\frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$	взвешенная F-мера:	$\frac{1}{\frac{\beta^2}{1+\beta^2} \frac{1}{Precision} + \frac{1}{1+\beta^2} \frac{1}{Recall}}$

Алгоритм kNN (k Nearest Neighbor или k Ближайших Соседей) – метрический алгоритм для классификации объектов или регрессии.

Гипотеза компактности (схожие параметры ведут к схожим меткам)

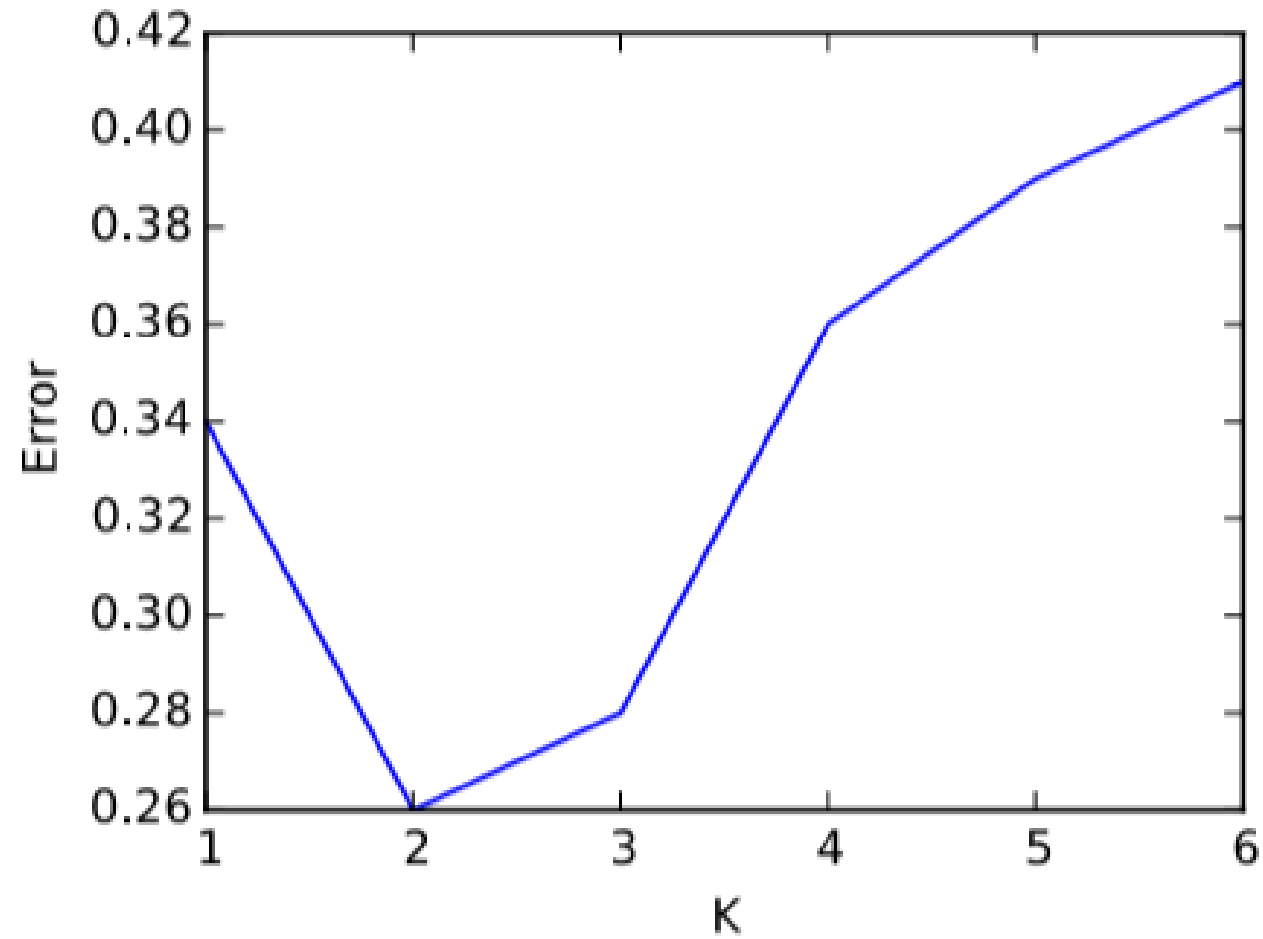
В случае использования метода для классификации объект присваивается тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Алгоритм может быть применен к выборкам с большим количеством атрибутов (многомерным). Для этого перед применением нужно определить функцию расстояния; классический вариант такой функции — евклидова метрика



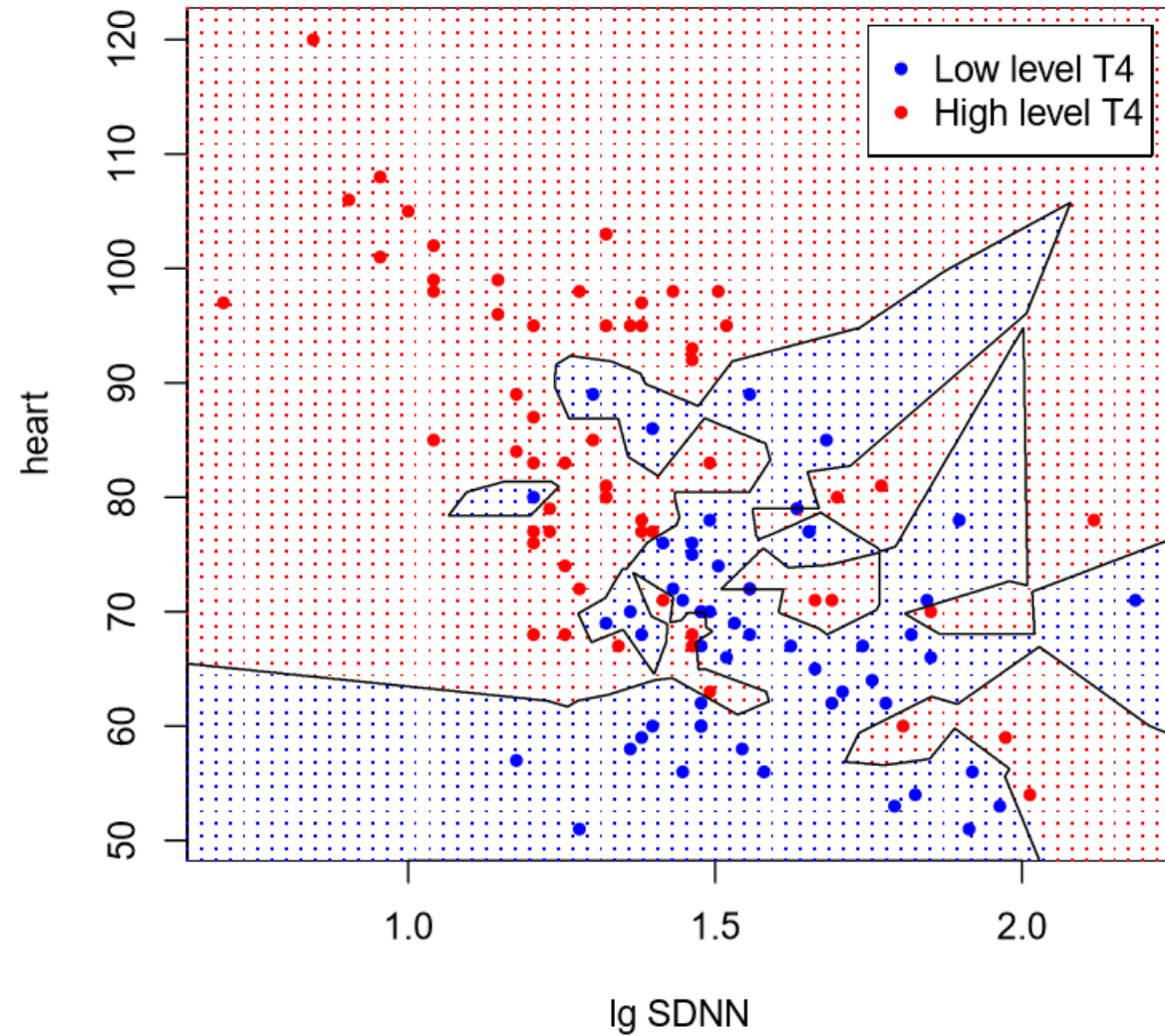
Пример классификации k-ближайших соседей. Тестовый образец (зелёный круг) должен быть классифицирован как синий квадрат (класс 1) или как красный треугольник (класс 2). Если $k = 3$, то он классифицируется как 2-й класс, потому что внутри меньшего круга 2 треугольника и только 1 квадрат. Если $k = 5$, то он будет классифицирован как 1-й класс (3 квадрата против 2 треугольников внутри большего круга)

Метод K-соседей. Доля ошибок при разных K



Метод k ближайших
соседей

Здесь явное
переобучение,
модель очень точно
описывает все
объекты выборки



Метод ближайшего соседа (с масштабированием)
Ошибка на обучающей выборке — 0%.

Достоинства и недостатки алгоритма

- устойчивость к выбросам и аномальным значениям, поскольку вероятность попадания содержащих их записей в число k -ближайших соседей мала;
- программная реализация алгоритма относительно проста;
- результаты работы алгоритма легко поддаются интерпретации.

К недостаткам алгоритм KNN можно отнести:

- данный метод не создает каких-либо моделей, обобщающих предыдущий опыт, а интерес могут представлять и сами правила классификации;
- при классификации объекта используются все доступные данные, поэтому метод KNN является достаточно затратным в вычислительном плане, особенно в случае больших объёмов данных;
- высокая трудоёмкость из-за необходимости вычисления расстояний до всех примеров;
- повышенные требования к репрезентативности исходных данных.

Ещё одной проблемой алгоритма KNN, характерной, впрочем, и для большинства методов классификации, является различная значимость признаков с точки зрения определения класса объектов. Учет фактора значимости признаков в алгоритме может позволить повысить точность классификации.