Введение в машинное обучение

Сапунов Григорий
CTO / Intento (inten.to)

Что такое машинное обучение?

Машинное обучение (Machine Learning)

Подобласть искусственного интеллекта. Data-driven подход, полагающийся на выучивание закономерностей из имеющихся размеченных или неразмеченных данных.

Идея в том, чтобы не программировать алгоритм решения задачи вручную, а “выучить” его из данных.

Составные части

Машинное обучение (ML) содержательно включает в себя 3 компоненты:

1. Представление (Representation)
2. Оценка (Evaluation)
3. Оптимизация (Optimization)

1. Представление (Representation)

Что собой представляет модель, какие классы задач она способна (и не способна) решать.

Примеры: разделяющая гиперплоскость, деревья решений, нейросети.

1. Представление (Representation)

Отдельный вопрос здесь же, это как именно представляются данные (включая этап выделения нужных признаков, feature extraction/engineering).

2. Оценка (Evaluation)

Как оценивать качество модели в контексте решения задачи, как выбирать лучшую модель из нескольких.

Примеры: RMSE, Accuracy/Precision/Recall, Logistic Loss, ...

3. Оптимизация (Optimization)

Как именно проводить обучение модели, каким именно образом осуществлять перебор пространства возможных моделей, чтобы найти лучшую.

Примеры: стохастический градиентный спуск, генетические алгоритмы, grid search, …

Процесс решения задачи с помощью машинного обучения

Процесс решения задачи

0. Определиться со способом решения вашей задачи средствами ML.

1. Выбрать метрику качества.

2. Получить данные и выделить признаки.

3. Определиться с классами моделей.

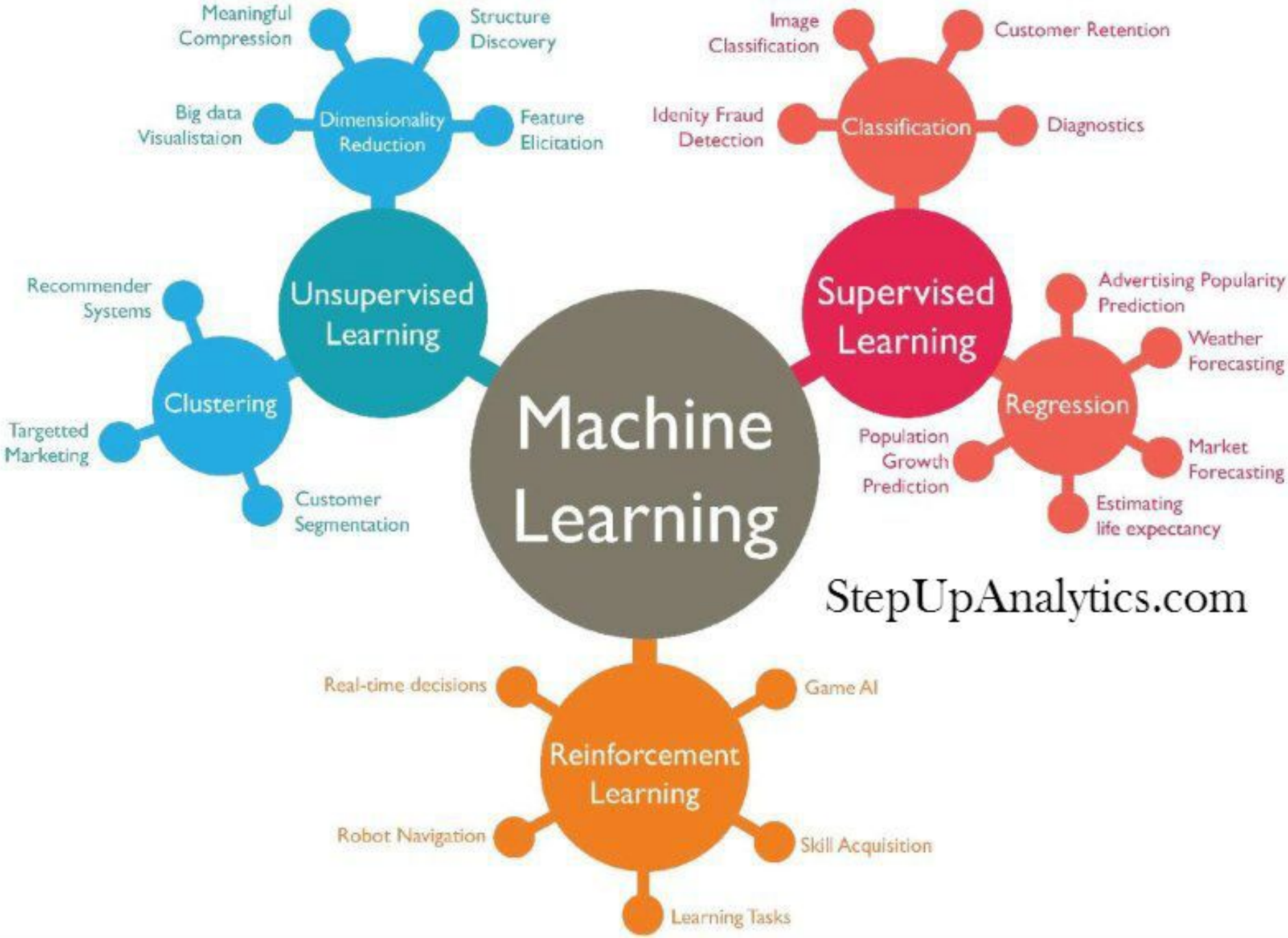
4. Подготовить данные для обучения и оценки

5. Провести обучение моделей, оценить результат, повторить шаги, если требуется.

Процесс машинного обучения: #0. Постановка задачи

Постановка задачи

Надо понять, как именно ваша конкретная задача может быть сведена к одной из типовых задач машинного обучения.


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
StepUpAnalytics.com
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Виды задач в ML

Машинное обучение разделяется на несколько основных подходов:

- **Обучение с учителем** (supervised learning)
  - Классификация (classification)
  - Регрессия (regression)
  - Ранжирование (learning to rank)
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация (clustering)
  - Уменьшение размерности (dimensionality reduction)
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Процесс машинного обучения: #1. Данные

Почему данные – это важно?

Замечание #1:

80% времени будет потрачено на работу с данными, а не на создание модели.

Интеграционная часть

Определение релевантных задаче данных, инвентаризация имеющихся данных, заказ недостающих данных, интеграция с информационными системами, выгрузка данных, конвертация в нужные форматы, …

Качество данных

Замечание #2:

Garbage In — Garbage Out

Качество данных — многоаспектная вещь. Данные должны быть: доступными, точными, когерентными, полными, консистентными, определёнными, релевантными, актуальными, …

Качество данных

Провал в каком-то из аспектов качества может сделать данные малопригодными или бесполезными.

Или, что ещё хуже, внешне годными, но ведущими к неправильным заключениям.

Данные vs. Алгоритмы

Замечание #3:

Похоже на то, что лимитирующий ресурс во многих случаях — это данные, а не алгоритмы.

Year	Breakthroughs in AI	Datasets (First Available)	Algorithms (First Proposed)
1994	Human-level spontaneous speech recognition	Spoken Wall Street Journal articles and other texts (1991)	Hidden Markov Model (1984)
1997	IBM Deep Blue defeated Garry Kasparov	700,000 Grandmaster chess games, aka "The Extended Book" (1991)	Negascout planning algorithm (1983)
2005	Google's Arabic- and Chinese-to-English translation	1.8 trillion tokens from Google Web and News pages (collected in 2005)	Statistical machine translation algorithm (1988)
2011	IBM Watson became the world Jeopardy! champion	8.6 million documents from Wikipedia, Wiktionary, Wikiquote, and Project Gutenberg (updated in 2010)	Mixture-of-Experts algorithm (1991)
2014	Google's GoogLeNet object classification at near-human performance	ImageNet corpus of 1.5 million labeled images and 1,000 object categories (2010)	Convolution neural network algorithm (1989)
2015	Google's Deepmind achieved human parity in playing 29 Atari games by learning general control from video	Arcade Learning Environment dataset of over 50 Atari games (2013)	Q-learning algorithm (1992)
Average No. of Years to Breakthrough:		**3 years**	**18 years**

http://www.spacemachine.net/views/2016/3/datasets-over-algorithms

Объекты и признаки

Каждый объект X характеризуется набором признаков $x_1, x_2, \ldots, x_m$

Пример:
Объект: *сообщение электронной почты*
Признаки: *набор слов, длина, дата, отправитель, получатель, язык, частота сообщения от данного адресата …*

Извлечение признаков

Признаки (features) — необходимое “топливо” для работы алгоритмов машинного обучения.

Признаки могут быть извлечены из данных “как есть”: возраст, вес, уровень экспрессии гена CTCF, открытость хроматина, число покупок, язык сообщения, …

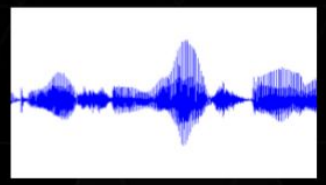
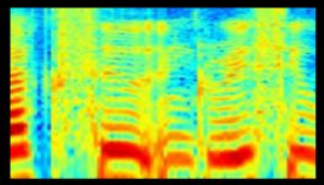
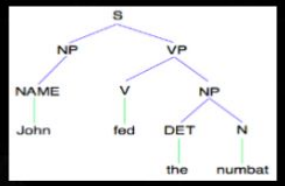
Извлечение признаков

Признаки могут быть “придуманы” исходя из опыта и интуиции:

- “Уход клиента может быть связан с количеством обращений в техподдержку, пусть будет признак <число обращений в техподдержку>”
- Число разрезов ДНКазой на участке длиной 100 нуклеотидов

Извлечение признаков

Для сложных объектов (изображения, звук) есть уже придуманные учёными признаки (SIFT, HoG, MFCC, …)

Raw data		Feature extraction		Classifier/ detector		Result
	⇨		⇨	SVM, shallow neural net, …	⇨	
	⇨		⇨	HMM, shallow neural net, …	⇨	Speaker ID, speech transcription, …
	⇨		⇨	Clustering, HMM, LDA, LSA …	⇨	Topic classification, machine translation, sentiment analysis…

Извлечение признаков

Deep Learning позволяет “учить” признаки наравне с моделью. И часто получается лучше разработанных вручную признаков.

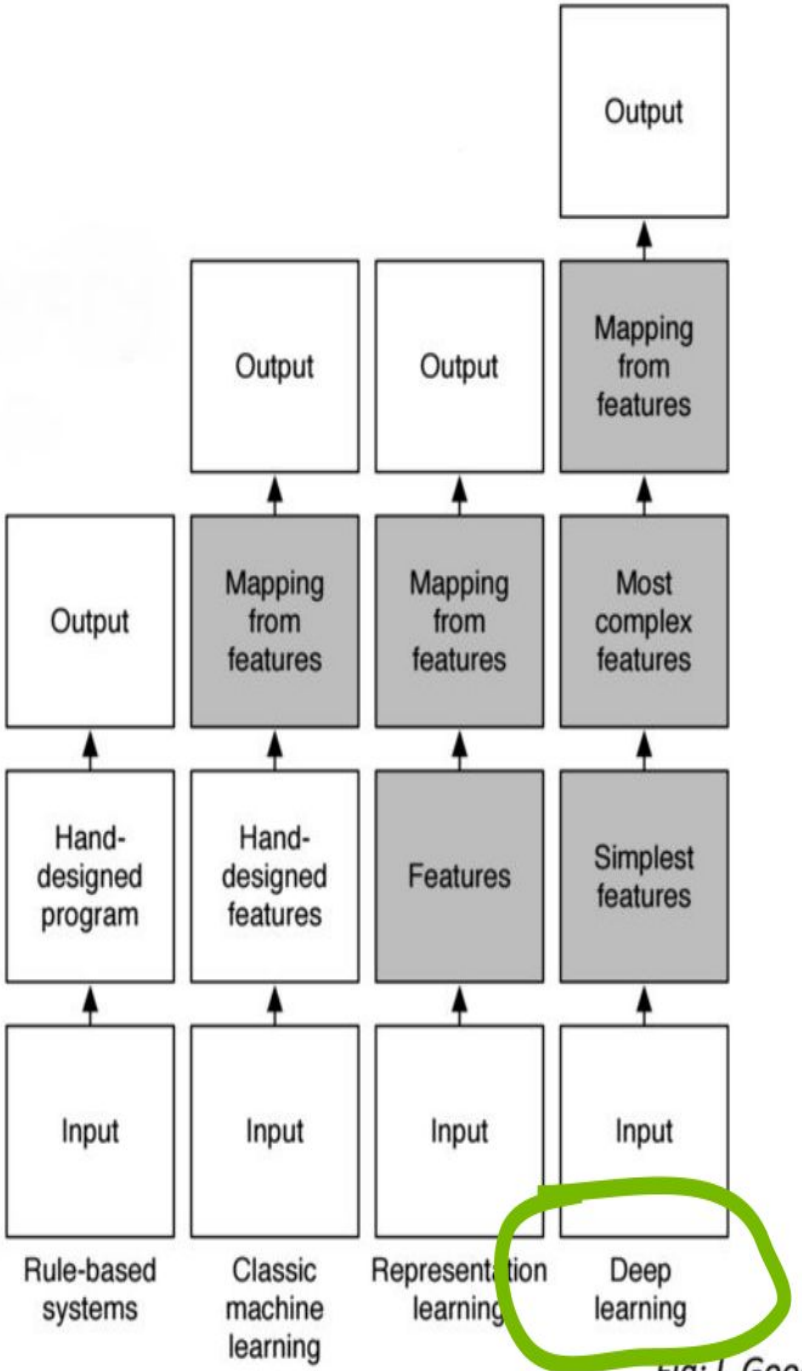


Fig: I. Goodfellow

Процесс машинного обучения: #2. Классы моделей

Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Обучение с учителем: постановка задачи

Каждый объект описывается парой <x,y>

- x – данные (многомерный вектор)
- y – целевое значение, метка

Надо найти функцию f(x)=y

Обучение с учителем: Типичные задачи

Классификация

- Значения y дискретны (принимают несколько заранее определённых значений, классов)
- Альтернативная постановка: предсказать вероятности определённых классов
- Пример: предсказание состояния хроматина, сайта связывания TF, растворимости химического вещества, пола, спам/не спам, вероятность ухода сотрудника/клиента

Регрессия

- Значения y непрерывны (принимают любое значение из диапазона)
- Пример: прогнозирование уровня экспрессии гена, цены жилья по его описанию, предсказание объёма продаж

Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

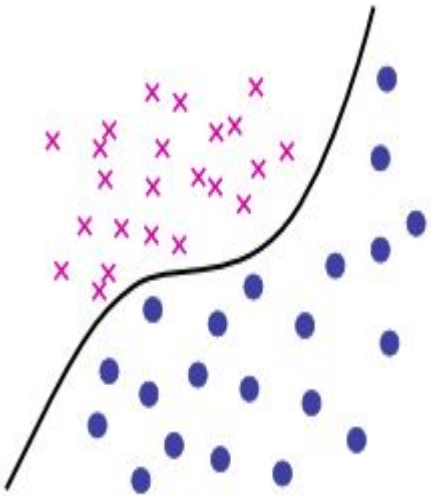
Классификация

Есть обучающая выборка, в которой представлены объекты в виде их признакового описания (вектор признаков) и метки класса.

Надо найти алгоритм, который для каждого нового объекта (его признакового описания) определит метку класса этого объекта.

Это эквивалентно построению разделяющей поверхн

многомерном признаковом пространстве.



Классификация: Примеры

- Предсказание состояния хроматина (открытый/закрытый)
- Предсказание сайта связывания транскрипционного фактора
- Предсказание растворимости нового химического вещества
- Предсказание летальности мутации
- Предсказание пола для неизвестного пользователя
- Предсказание категории письма: спам/не спам, важное/не важное
- Вероятность ухода сотрудника/клиента
- Вероятность невозврата кредита
- Определение языка для неизвестного текста
- Определение темы новостного сообщения
- Определение объекта на фотографии: человек, дом, автомобиль.
- Классификация типов клеток по фотографиям
- Определение эмоциональной окраски твита
- Определение состояния человека по ЭЭГ

Классификация: Примеры

Предсказание кристаллизации селенидов ванадия в присутсвии разнообразных органических аминов.

“Оказалось, что полученная таким образом система правильно предсказывает результат реакции в 89 процентах случаев. При этом сами химики, руководствуясь профессиональной интуицией, правильно угадывают исход эксперимента только в 78 процентах случаев. По статистическому анализу, который приводят авторы, достоверность (неслучайность) этого преимущества машины над человеком составляет более 95 процентов (P <0,05).”

Компьютер обошел человека в неорганическом синтезе
https://nplus1.ru/news/2016/05/05/neoneo

Machine-learning-assisted materials discovery using failed experiments
http://www.nature.com/nature/journal/v533/n7601/full/nature17439.html

Классификация: Важные моменты

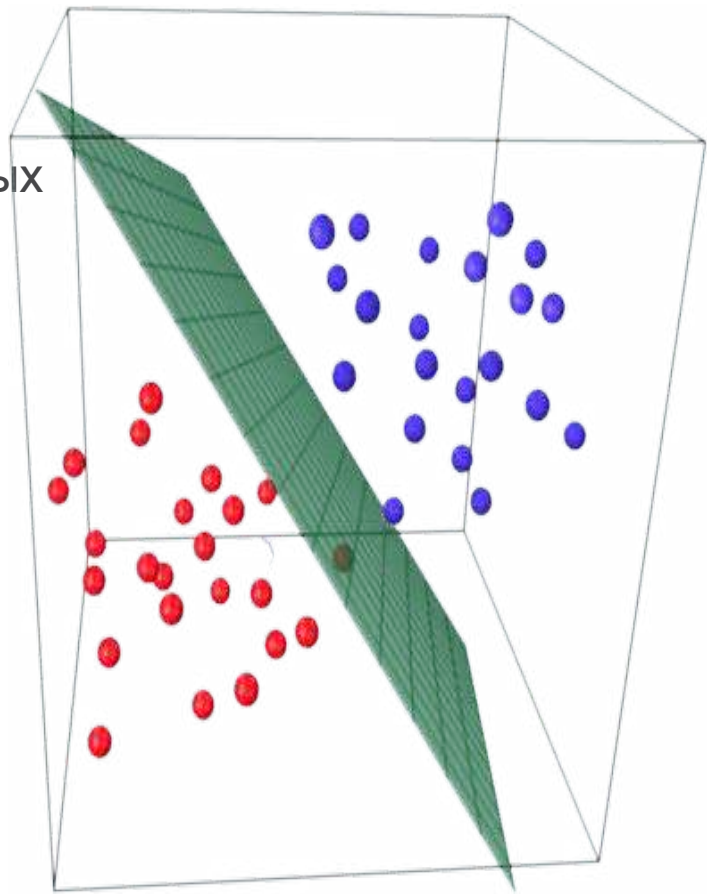
- Классификация — это метод обучения с учителем. Требуется размеченная выборка
- Метка принимает набор дискретных значений (эквивалентно номерам классов + словарю)
- Вид признакового пространства и расположение классов внутри него определяют какой классификатор справится с задачей (но визуализировать это практически невозможно для большого числа измерений)
- Частный популярный класс классификаторов — линейные классификаторы с разделяющей гиперплоскостью в качестве границы.

Классификация: Популярные методы

- Логистическая регрессия
- Машина опорных векторов (Support vector machine, SVM)
- Наивный Байес
- Деревья решений
- Нейросети
- Ансамбли (Random Forests, Gradient Boosted Trees)
- k-NN
- ...

Классификация: Логистическая регрессия

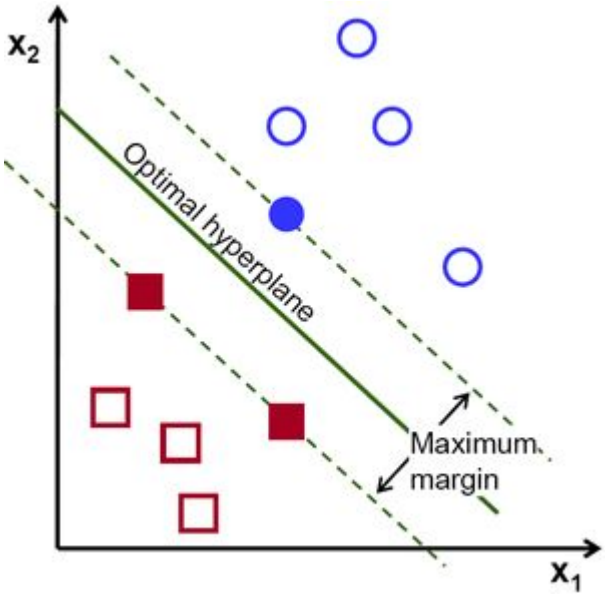
- Один из самых простых классификаторов
- Довольно эффективен на малом объёме данных и может превзойти более сложные методы (деревья решений, deep learning), но проиграет на большом объёме.
- Линейный классификатор (но можно сделать нелинейным с помощью генерации новых признаков, например, квадратичных)
- Вид модели: линейное правило + нелинейная функция после него (для приведения к диапазону от 0 до 1)



$$z = \theta^T x = \theta_1 x_1 + \ldots + \theta_n x_n \qquad f(z) = \frac{1}{1+e^{-z}}.$$

Классификация: SVM

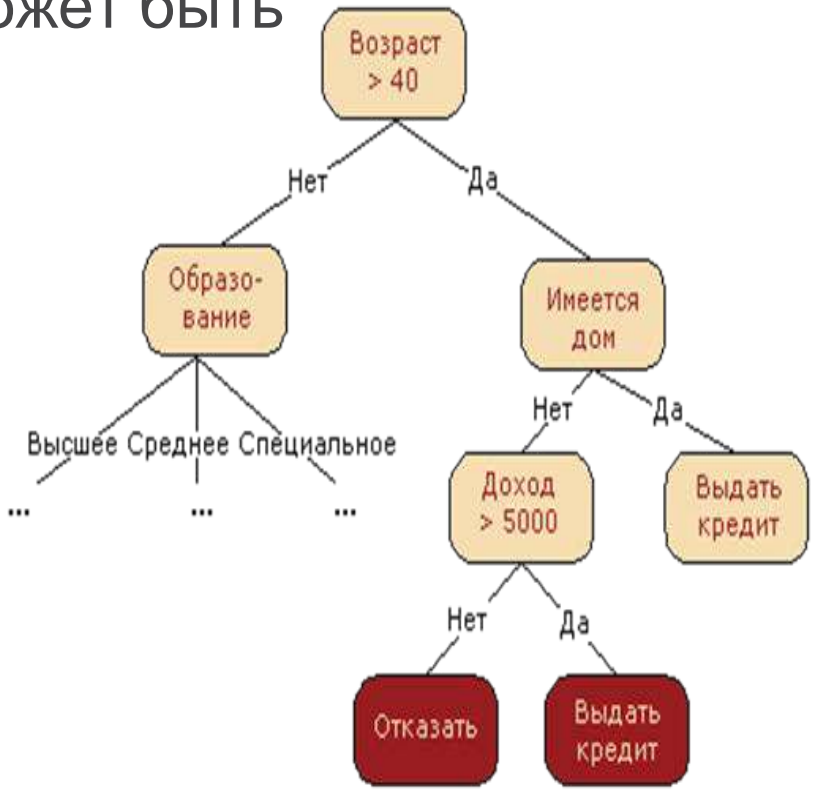
- SVM = Support Vector Machine = Машина опорных векторов
- Простой вариант линейный, но часто используются более сложные варианты с использованием kernel trick, который по сути производит перевод в новое пространство
- Идея: различных разделяющих плоскостей может быть много, надо выбрать “лучшую”.
- Алгоритм использует предположение, что чем больше расстояние между разделяющей гиперплоскостью и ближними к ней объектами классов, тем меньше будет средняя ошибка классификатора в последующей работе.
  Те самые ближайшие к границе (разделяющей гиперпроскости) вектора и называются опорными.



Классификация: Деревья решений

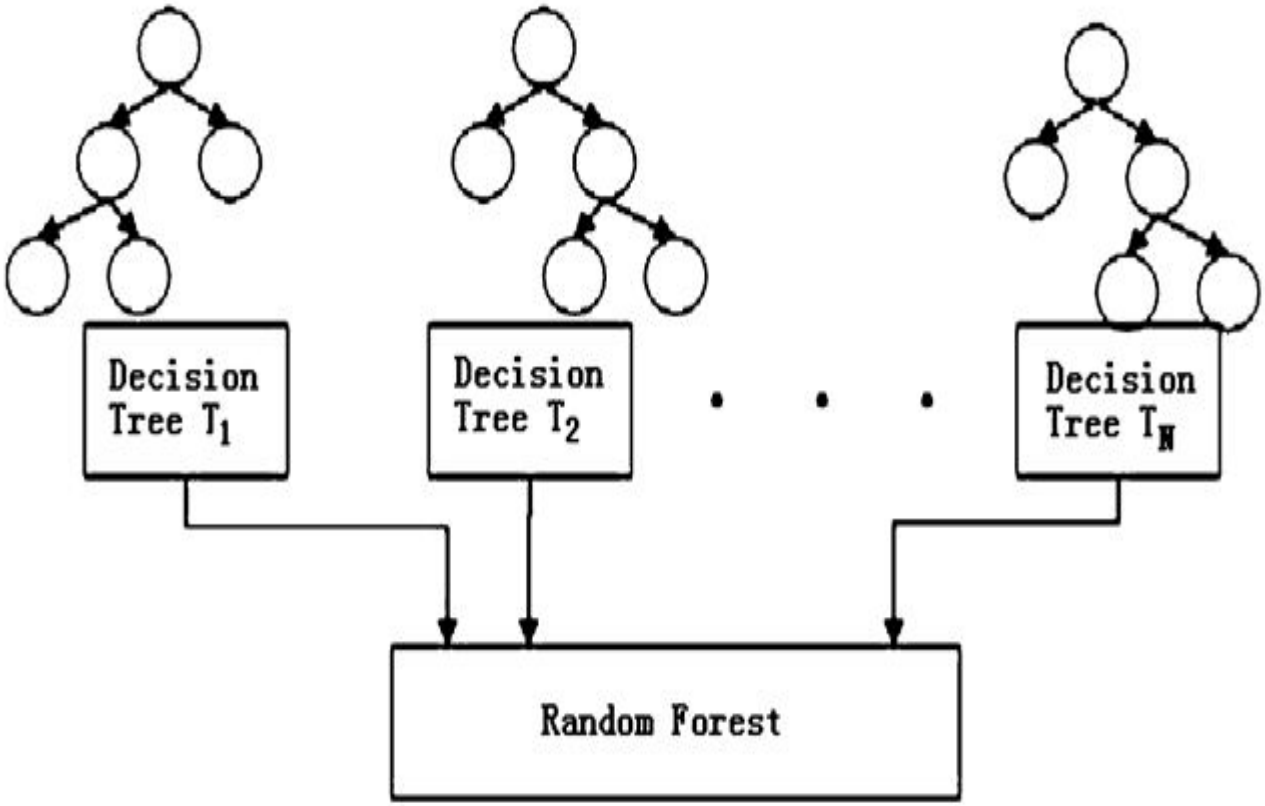
Набор узлов, в каждом из которых принимается решение по одной из переменных, куда двигаться дальше.

Удобны в интерпретации (это может быть требованием к алгоритму)



Классификация: Ансамбли

Идея: Использовать много простых классификаторов (например, 50 простых деревьев) и агрегировать результат. Пример: Random Forest.



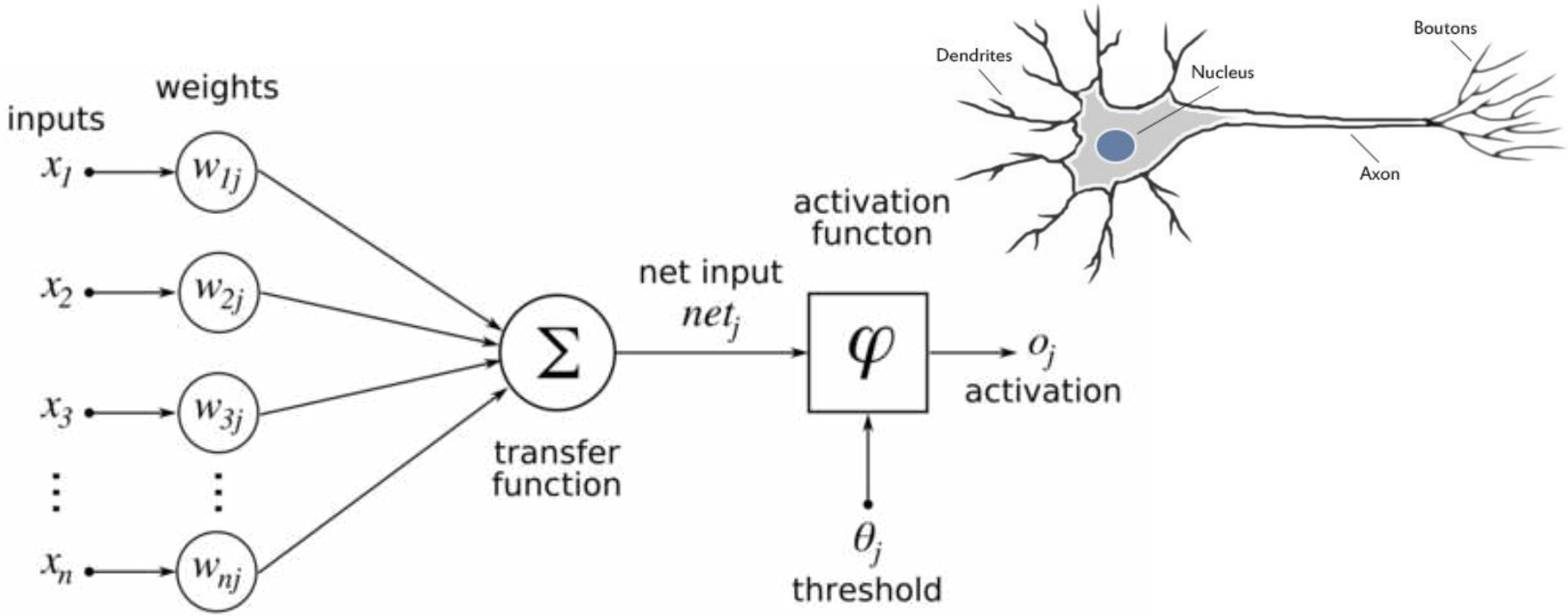
Классификация: Нейросети

- В настоящее время происходит возрождение нейросетей под новым брендом deep learning
- Иерархический классификатор, способный самостоятельно выделять признаки в исходном сигнале
- Очень хорошо работают для изображений, видео, звука.
- Хорошо работают для текстов.
- В ближайшем будущем ожидается массовое применение для анализа временных рядов, сигналов с датчиков.

- Как правило для обучения нейросетей используется так называемый метод обратного распространения ошибки (back-propagation).

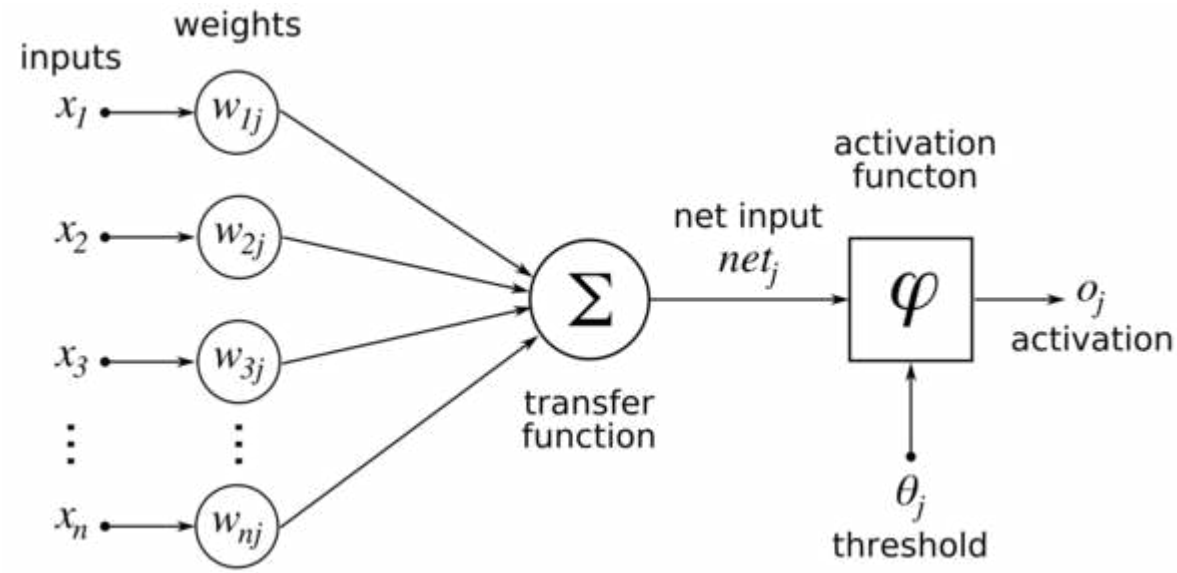
Классификация: Нейросети

Нейросети: базовый элемент -- искусственный нейрон (отдалённое подобие биологического).



Классификация: Нейросети

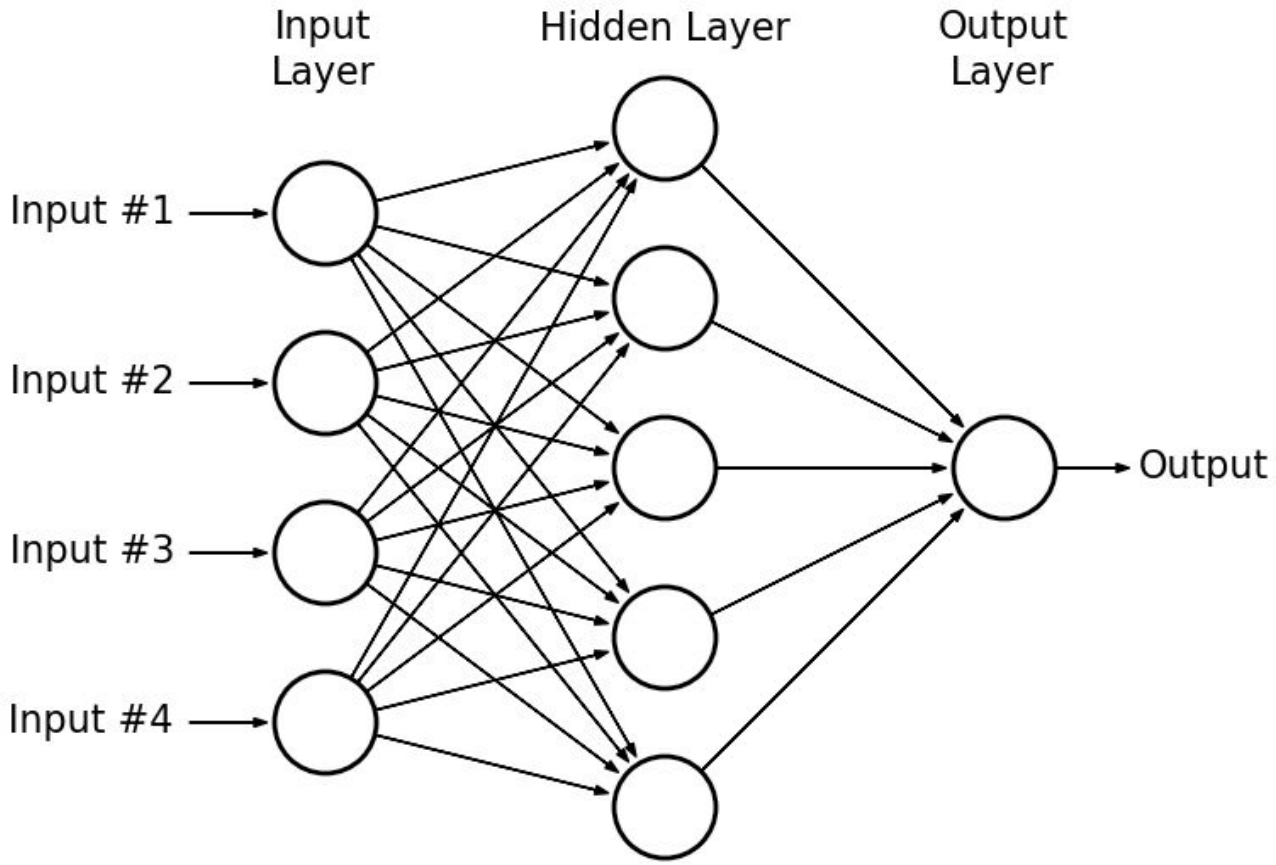
Отдельный нейрон математически полностью аналогичен логистической регрессии (в случае если используется функция активации сигмоида)



$$f(z) = \frac{1}{1+e^{-z}}.$$

Классификация: Нейросети

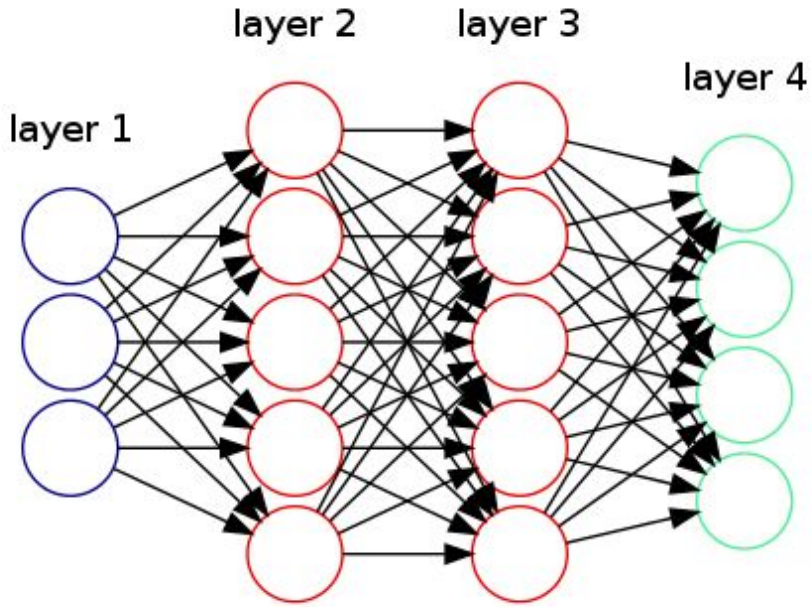
Пример: нейросеть с одним скрытым слоем (hidden layer)



Классификация: Нейросети

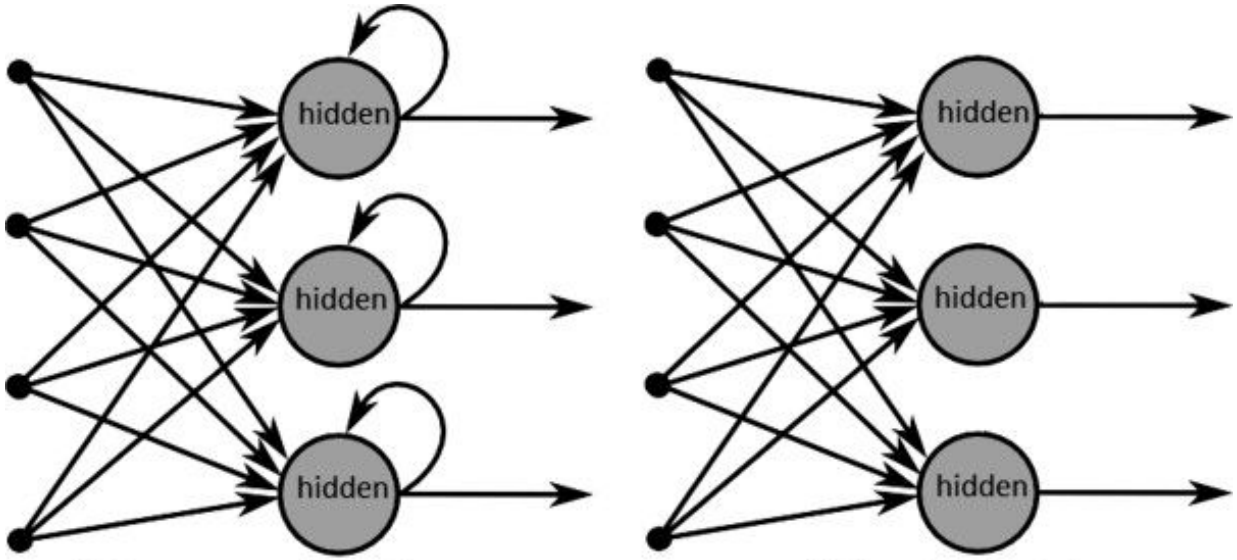
Наличие более одного скрытого слоя даёт возможность строить иерархические представления.

Пример: нейросеть с двумя скрытыми слоями (современные нейросети для обработки изображений уже могут иметь сотни слоёв):



Классификация: Нейросети

Пример: рекуррентная нейросеть (Recurrent Neural Network, RNN), ещё более сложный и мощный класс нейросетей, способный работать с последовательностями, а не только с отдельными объектами, как обычные сети прямого распространения (Feedforward Neural Networks, FNN). Также может быть многослойной.

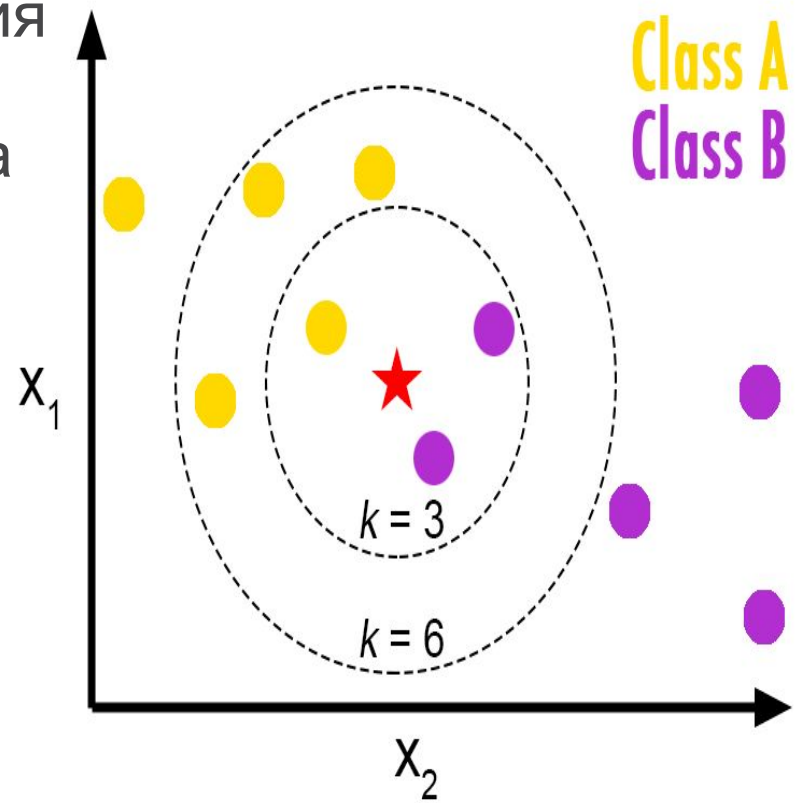


(a) Recurrent neural network (b) Forward neural network

Классификация: kNN

K ближайших соседей (k nearest neighbors). Не путать с NN (нейросети). Простой классификатор с сильно нелинейной границей.

Требует запоминания обучающих примеров, а сама классификация происходит по “голосованию” среди K соседей нового объекта ещё не известного класса.



Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

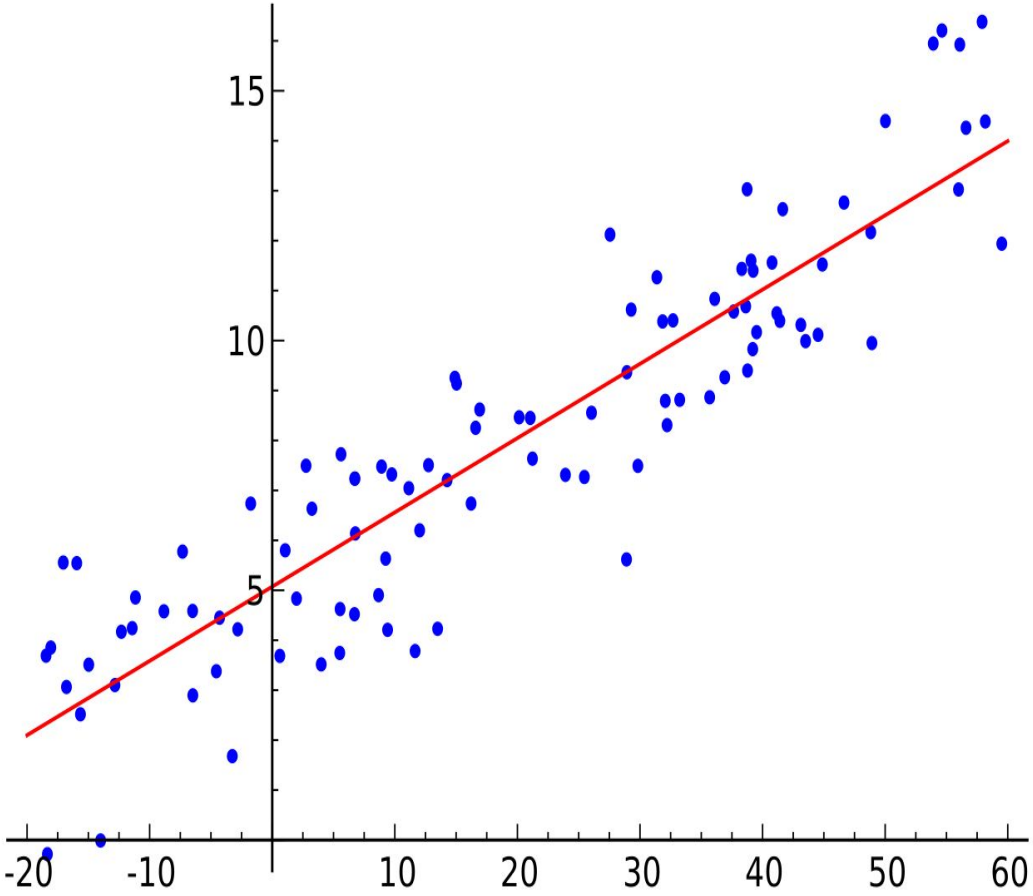
Регрессия

Есть обучающая выборка, в которой представлены объекты в виде их признакового описания (вектор признаков) и значения целевой переменной (непрерывной в отличие от классификации).

Надо найти алгоритм, который для каждого нового объекта (его признакового описания) спрогнозирует значение целевой переменной.

Регрессия

Геометрически определяет прямую (в случае линейной регрессии), наиболее близко проходящую ко всем точкам.



Регрессия: Примеры

- Прогнозирование уровня экспрессии гена
- Предсказание цены дома по его характеристикам
- Предсказание спроса на товар на ближайший месяц
- Предсказание уровня воды в водохранилище
- Предсказание температуры воздуха
- …

Регрессия: Важные моменты

- Регрессия – это метод обучения с учителем. Требуется размеченная выборка
- Целевое значение может быть любым действительным числом
- Аналогично классификации есть линейные и нелинейные модели. Нелинейную регрессию часто можно получить с помощью генерации новых признаков из старых.
- Может быть чувствительна к выбросам (но есть методы борьбы)

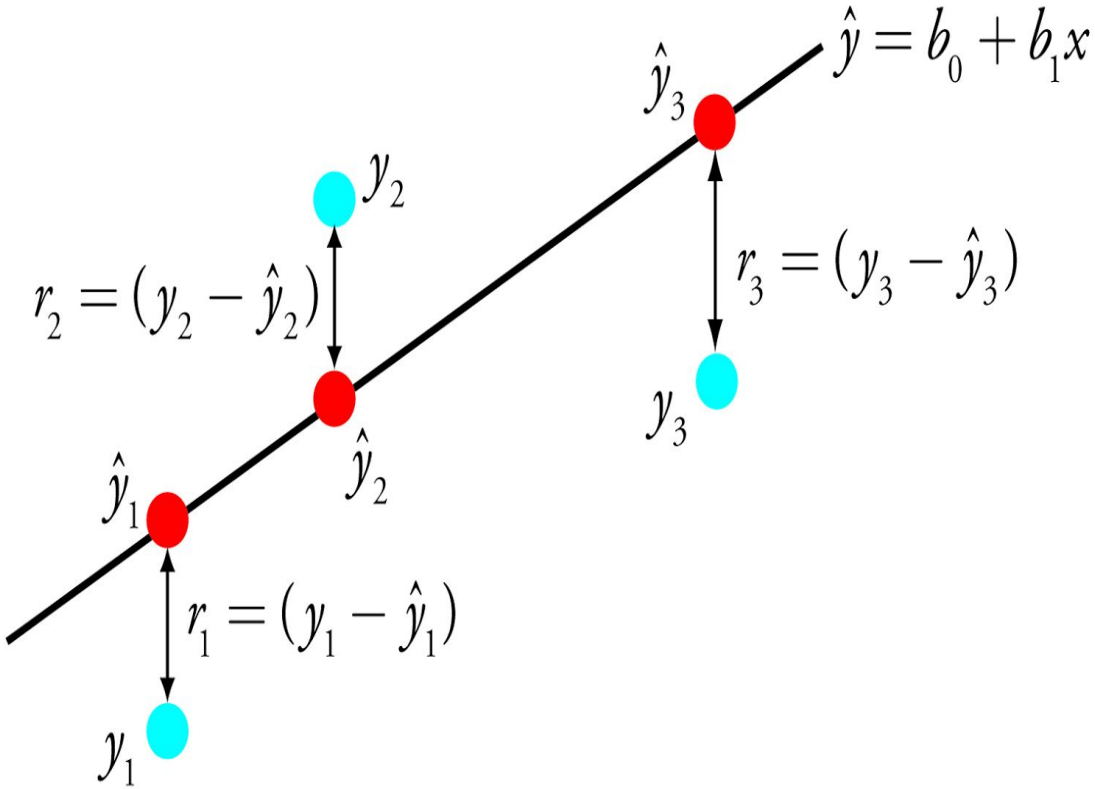
Регрессия: Методы

- Линейная регрессия (в том числе множественная)
- Регрессионные деревья
- Регрессия через SVM
- Нейросети
- ...

Регрессия: Линейная регрессия

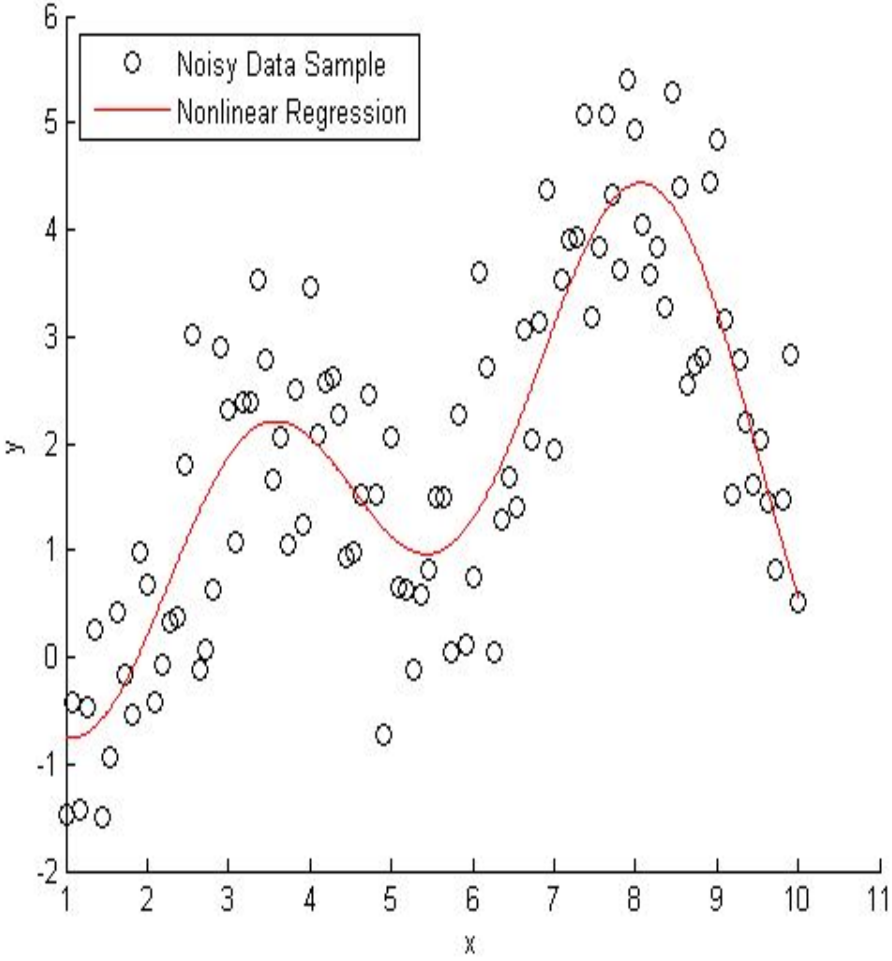
Простой алгоритм. Вписать линию, которая “лучшим” образом проходит через облако точек (наших примеров). Решение методом наименьших квадратов.

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$



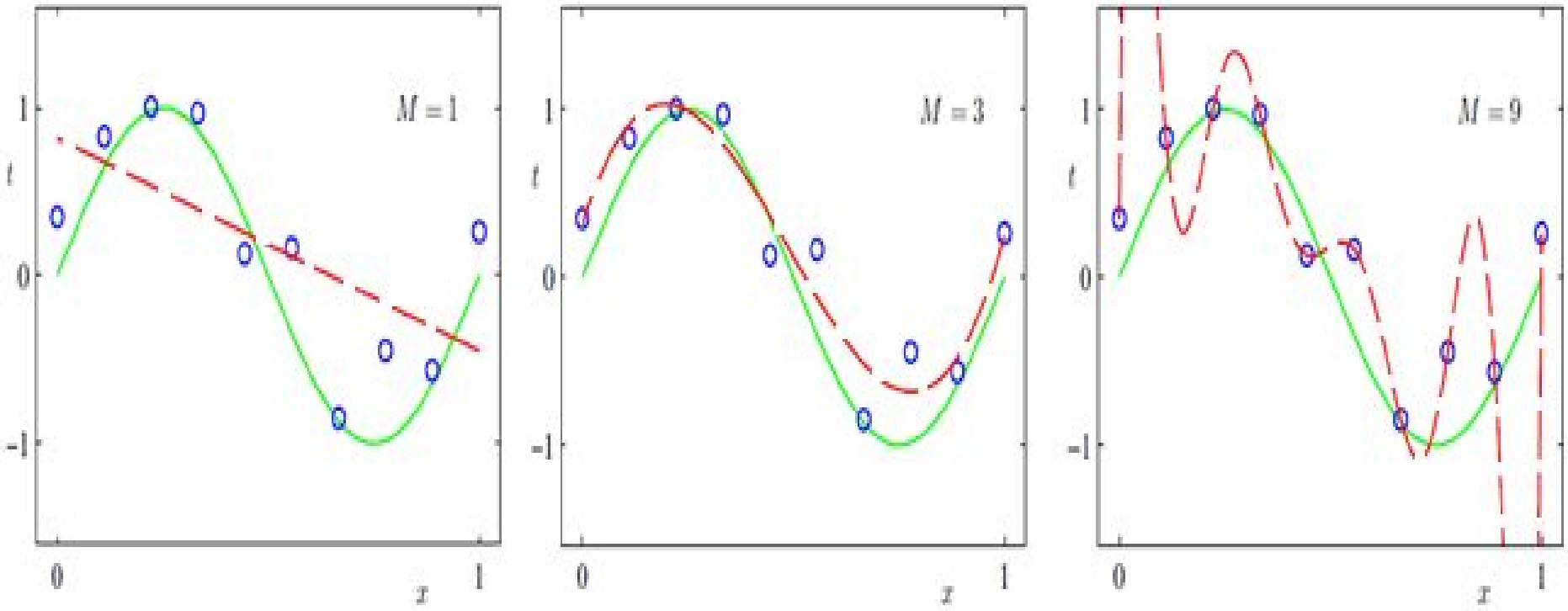
Регрессия: Нелинейная регрессия

Можно получить не меняя математический аппарат, добавив сгенерированные признаки (например, полиномиальные).

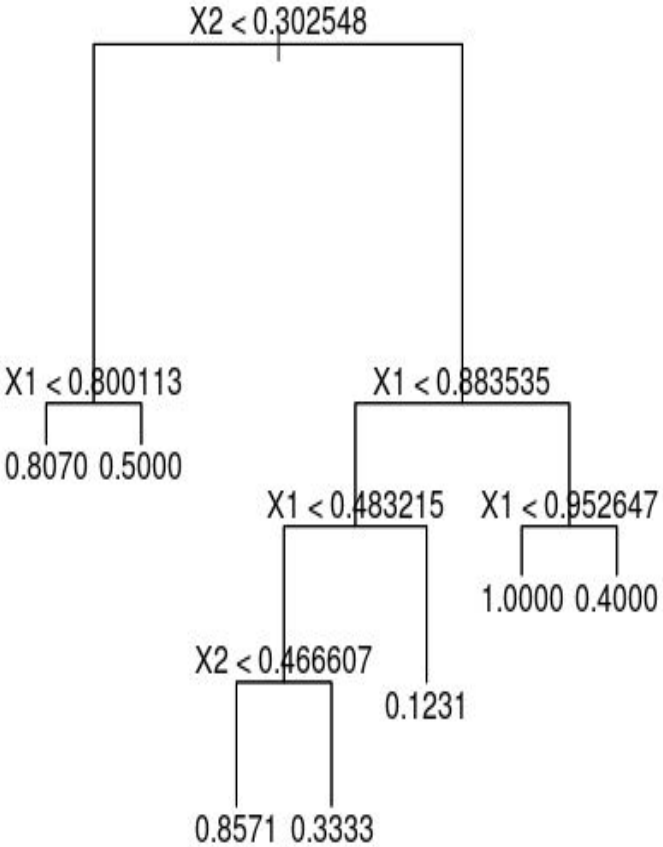


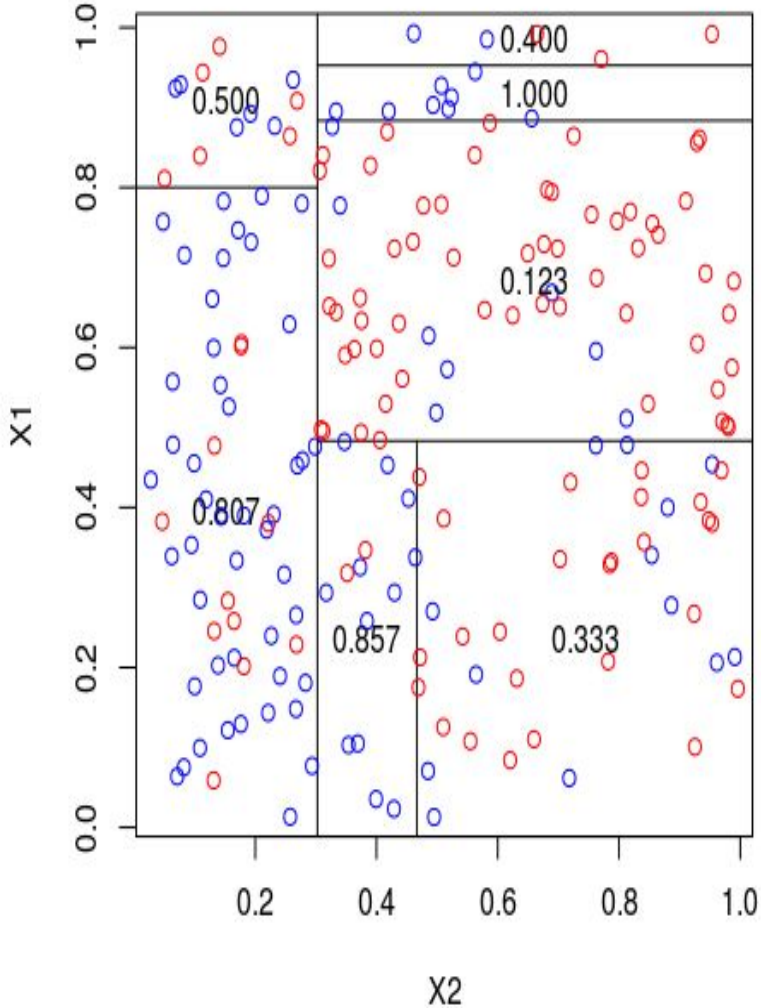
Регрессия: Нелинейная регрессия

Важно не переусердствовать при добавлении нелинейных признаков



Регрессия: Деревья и ансамбли





Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Ранжирование

Даны списки объектов и частичные порядки на этих списках (например, известно, какой элемент идёт за каким).

Задача: построить ранжирующую модель, обобщающую способ ранжирования на новые данные

Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Обучение без учителя

Постановка задачи:

Каждый объект описан вектором признаков $<x_1,x_2,...,x_n>$

Найти механизм, который описывает структуру этих данных (которую мы заранее не знаем)

Обучение без учителя

Типичные задачи:

- Кластеризация
  - Объединить элементы в группы по их похожести
  - Пример: группировка клеток по профилю экспрессии; домохозяйств по паттерну потребления электроэнергии; пользователей сервиса по поведению; кластеризация клиентов на группы.
- Уменьшение размерности
  - Перевести данные в пространство меньшей размерности, с сохранением отношений между элементами
  - Пример: визуализация многомерных данных

Классы моделей в ML

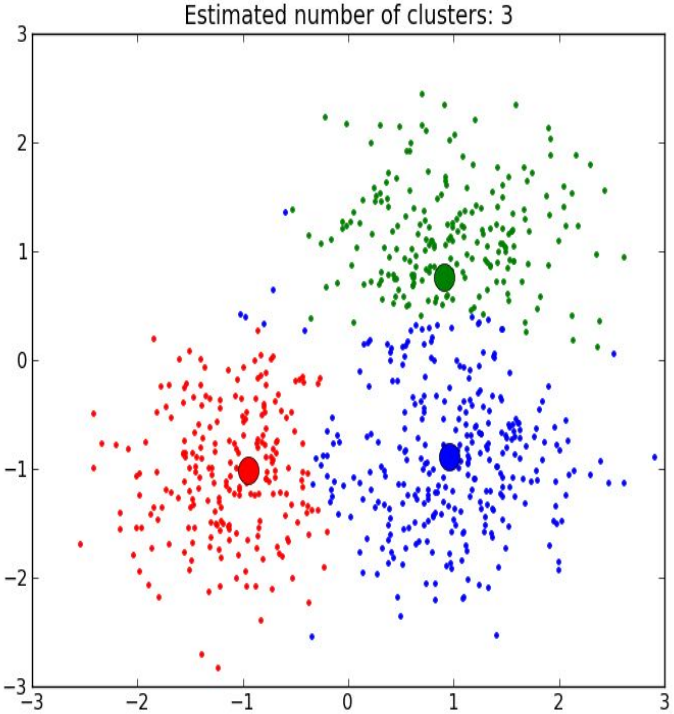
- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Кластеризация

Каждый объект представлен в виде многомерного вектора $<x_1,x_2,...,x_n>$

Задача: сгруппировать объекты по “похожести” так, чтобы:

- Объекты внутри одной группы были более похожи друг на друга, чем на объекты из любой другой группы
- Количество групп может как задаваться изначально, так и определяться по ходу дела
- Под группировкой имеется в виду приписать каждому объекту номер кластера



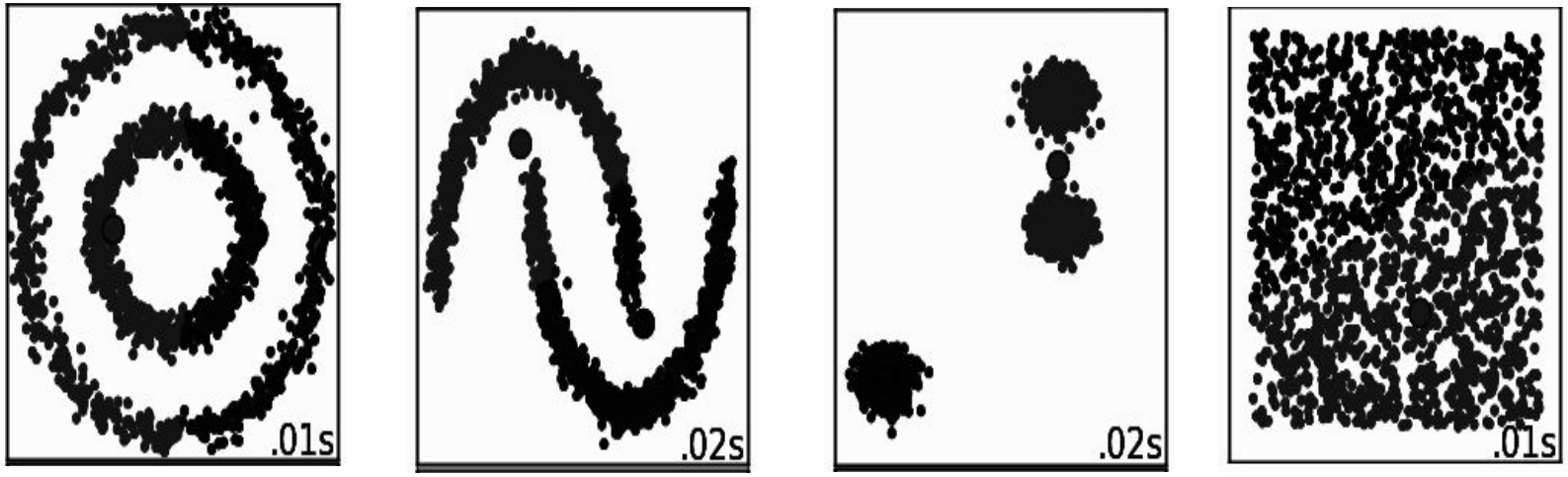
Кластеризация: Примеры

- Группировка клеток по профилю экспрессии
- Группировка домохозяйств по паттерну потребления электроэнергии
- Группировка покупателей по потребительскому поведению (частота покупок, виды товаров в корзине, средний чек, …)
- Группировка пользователей сервиса по модели поведения (число просмотров страниц, продолжительность сессии, активность в течение недели или суток, участие в создании контента, …)
- Кластеризация клиентов на группы по модели поведения или социально-демографическим характеристикам
- Исследование рынка и обработка результатов опросов
- Определение сообществ в социальных сетях
- Определение учеников с похожим стилем обучения и выделение типичных групп

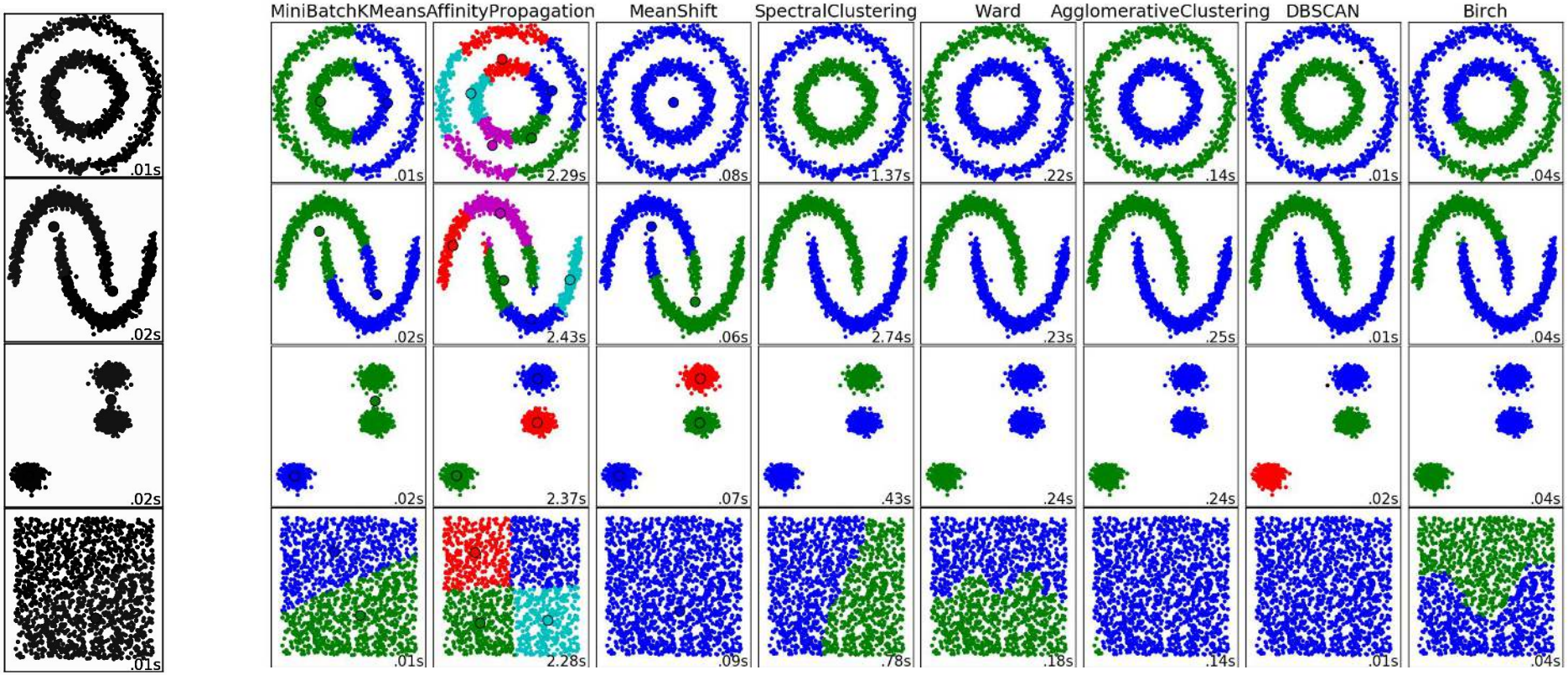
Кластеризация: Важные моменты

- Обучение без учителя (нет размеченной выборки)
- Похожесть или близость объектов обычно определяется через расстояние в многомерном пространстве признаков
- Надо определить само пространство (что учитываем) и метрику близости (как именно считаем близость)
- Может быть полезно для исследования датасета и получения интуиции о его структуре
- Результат трудно визуализировать, если число измерений больше четырёх
- Результат можно использовать для других методов (например, классификации и регрессии) как новый признак

Кластеризация: Пример



Кластеризация: Пример



Кластеризация: Методы

- K-means (K-means++, K-means||)
- Иерархическая кластеризация
- Кластеризация на основе распределений (EM-алгоритм)
- Кластеризация на основе плотности (DBSCAN)
- Графовая кластеризация (определение клик)
- ...

Кластеризация: Методы

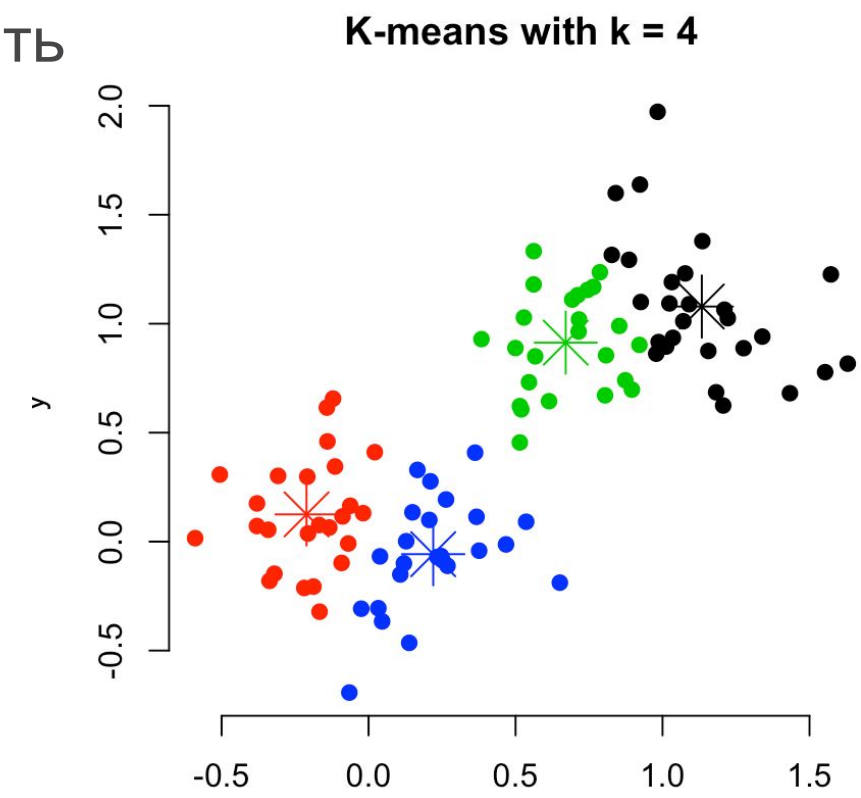
Кластеризация бывает:

- Жёсткая (hard) кластеризация: каждый объект относится только к одному кластеру
- Мягкая или нечёткая (soft или fuzzy) кластеризация: каждый объект может относиться к нескольким кластерам
- Некоторые методы позволяют не относить объект ни к одному кластеру (например, считать его выбросом).

Кластеризация: K-means

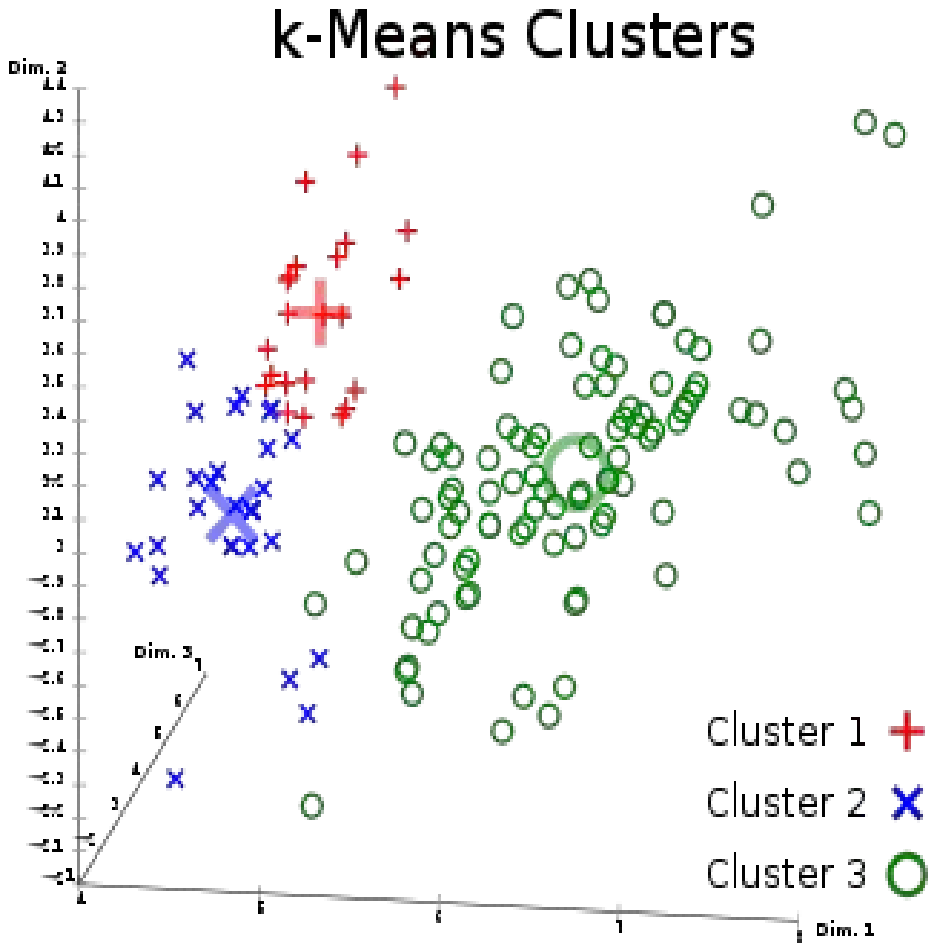
Простой и лёгкий для вычисления алгоритм.
Количество кластеров (k) задаётся изначально.

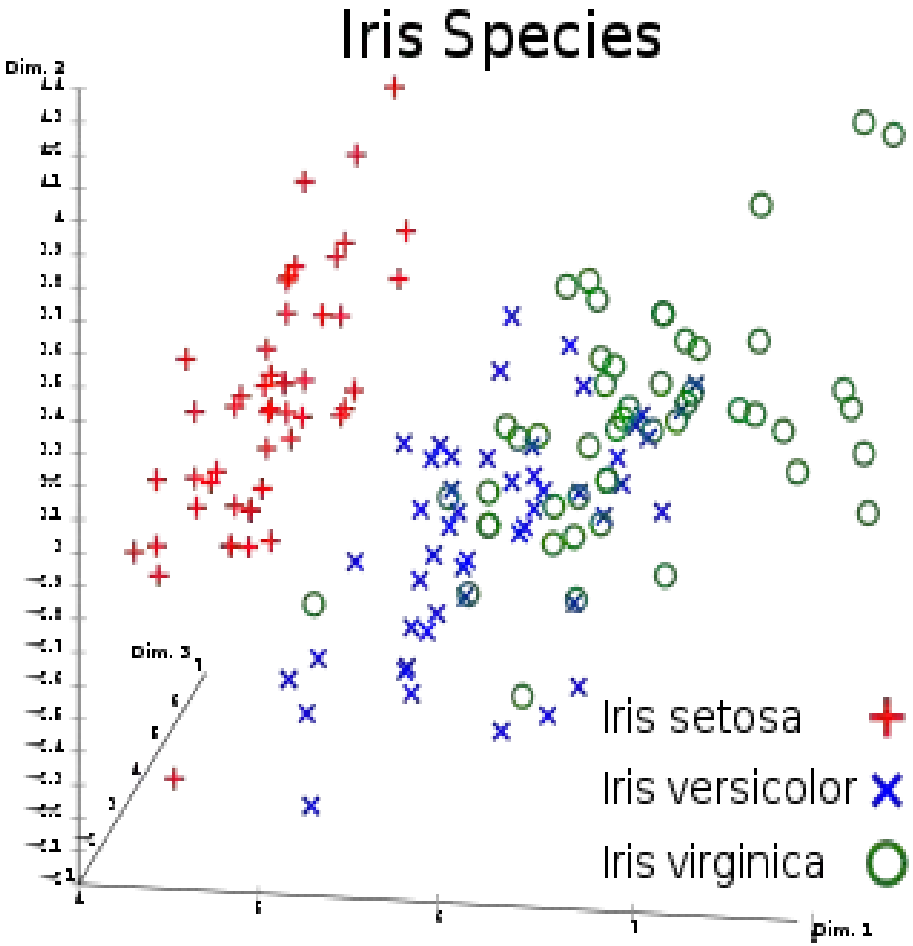
Мы можем не знать заранее, сколько кластеров есть в данных, но можно перебрать несколько вариантов и выбрать лучший (по какому-либо критерию, в том числе и автоматически)



Кластеризация: K-means

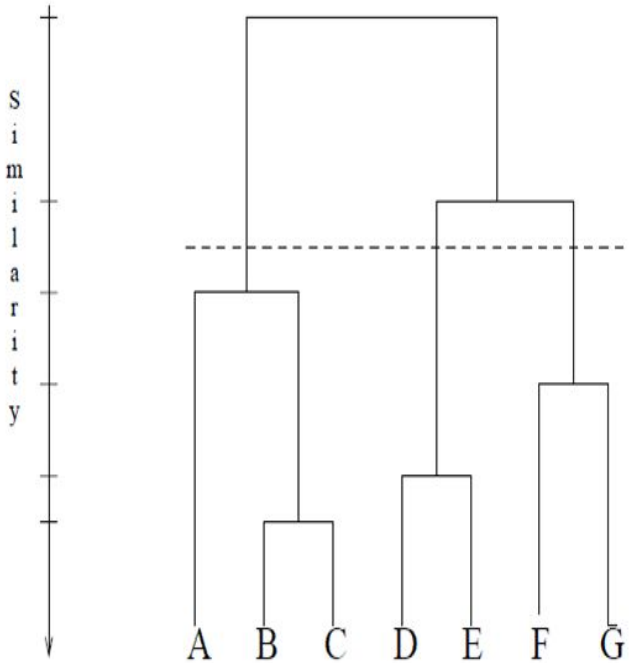
Для кластеров сложной формы может давать плохой результат

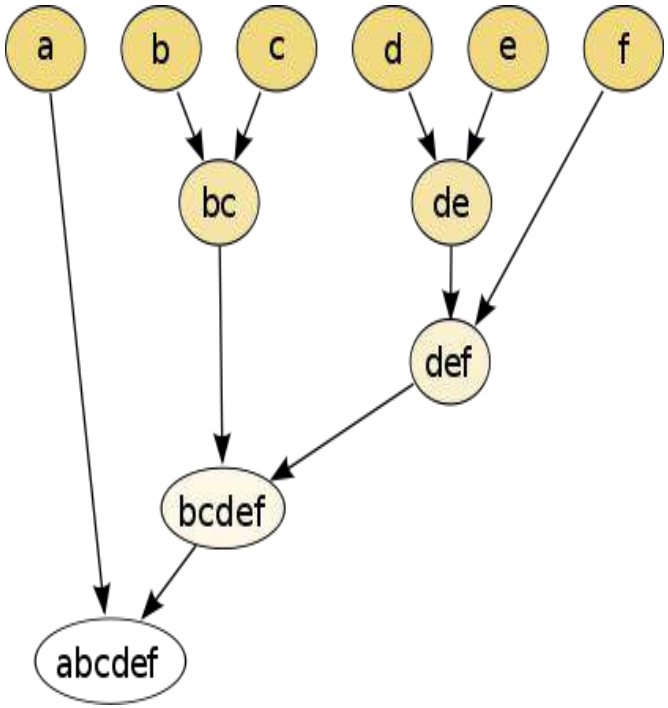




Иерархическая кластеризация

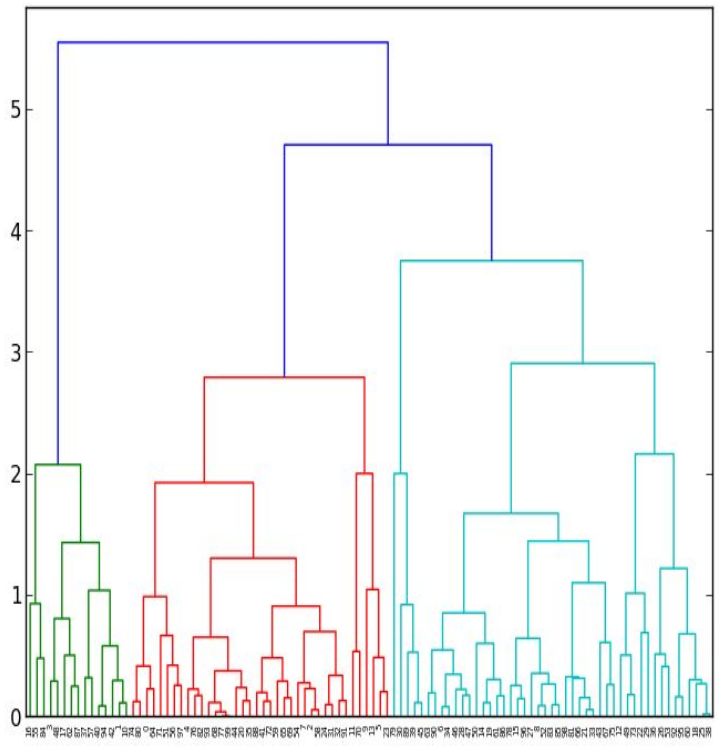
- Алгоритм простой, но вычислительно сложный (не используется с Big Data)
- Использует идею последовательного объединения самых похожих объектов

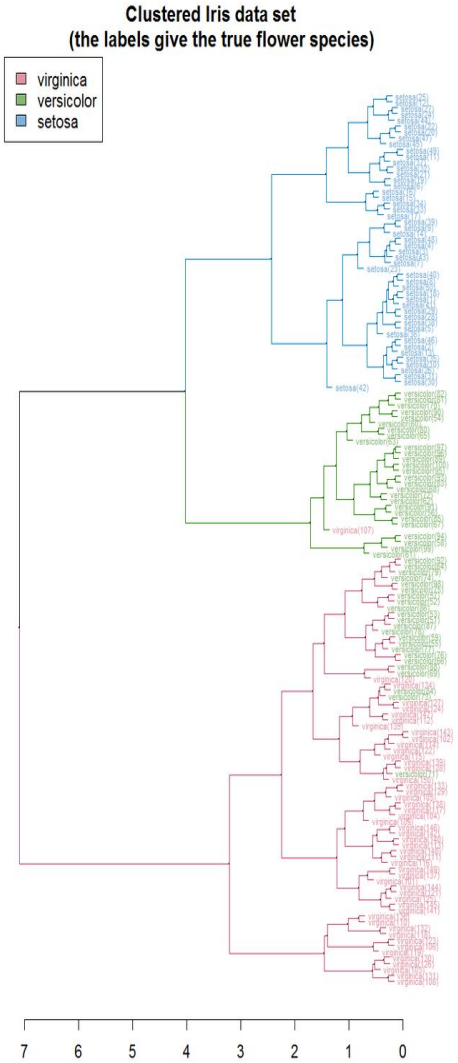




Иерархическая кластеризация

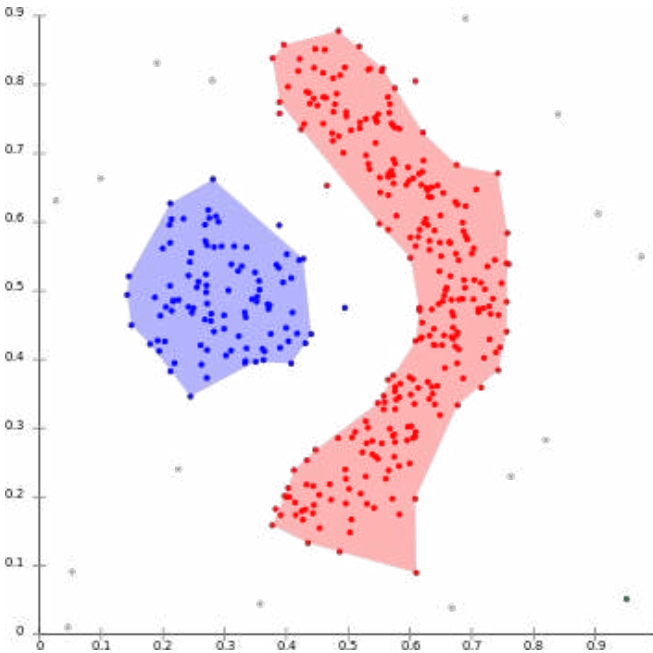
- Результатом является дендрограмма
- Проблемы с отображением при большом числе объектов





Кластеризация: DBSCAN

- Плотностный алгоритм: группирует вместе “плотные” участки
- Может находить кластеры очень сложной формы
- Часть точек не относятся никуда и считаются шумом
- Проблемы при кластеризации данных с сильно различающейся плотностью.
- Может быть трудно выбрать параметры для новых неизученных датасетов (а уже изученные кластеризовать неинтересно)

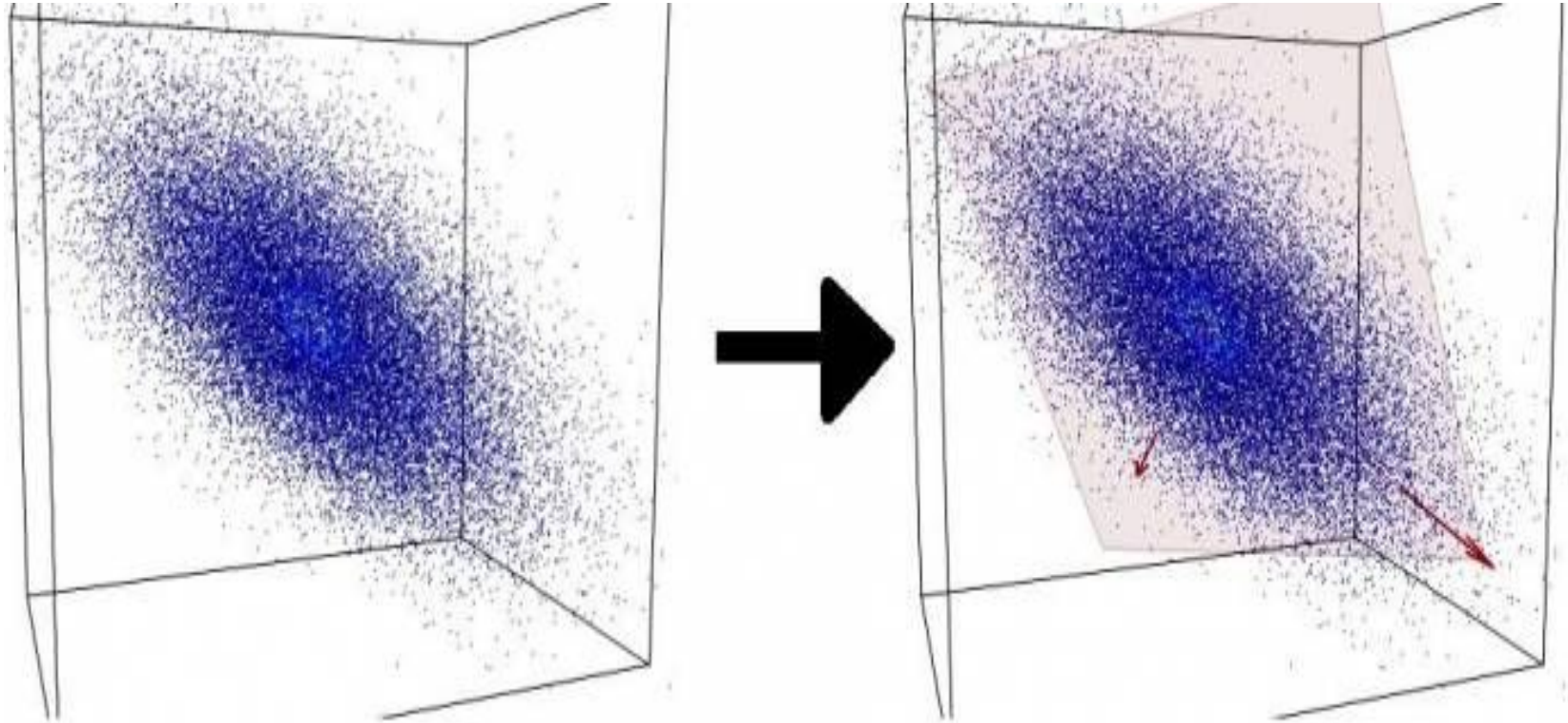


Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Уменьшение размерности

Каждый объект представлен в виде многомерного вектора $<x_1,x_2,...,x_n>$
Задача: получить более компактное признаковое описание объекта $<x_1,x_2,...,x_k>$ где $k<n$



Уменьшение размерности: Важные моменты

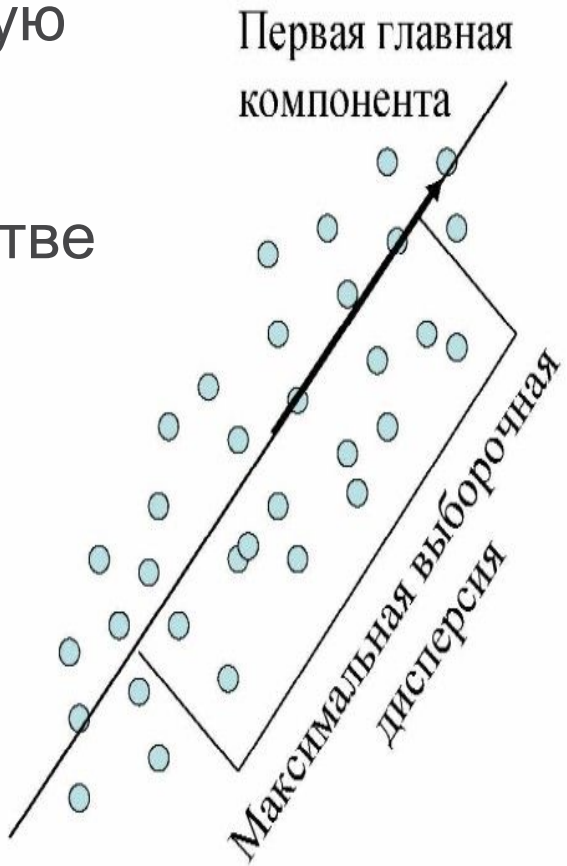
- Обучение без учителя (нет размеченной выборки)
- Помогает устранять “проклятие размерности”: данные быстро становятся разреженными при увеличении размерности
- Удобно использовать для визуализации, когда многомерный набор данных сводится к 2–3 измерениям, на которых могут быть видны закономерности
- Удобно использовать для разведочного анализа
- Уменьшает размер датасета, требуемое для хранения место и время для обработки
- Может помочь другим методам машинного обучения за счёт устранения избыточных данных (например, сильно скореллированных)

Уменьшение размерности: методы

- Метод главных компонент (principal component analysis, PCA)
- Метод независимых компонент (ICA)
- Многомерное шкалирование (Multidimensional scaling, MDS)
- Автоэнкодеры
- t-distributed stochastic neighbor embedding (t-SNE)
- ...

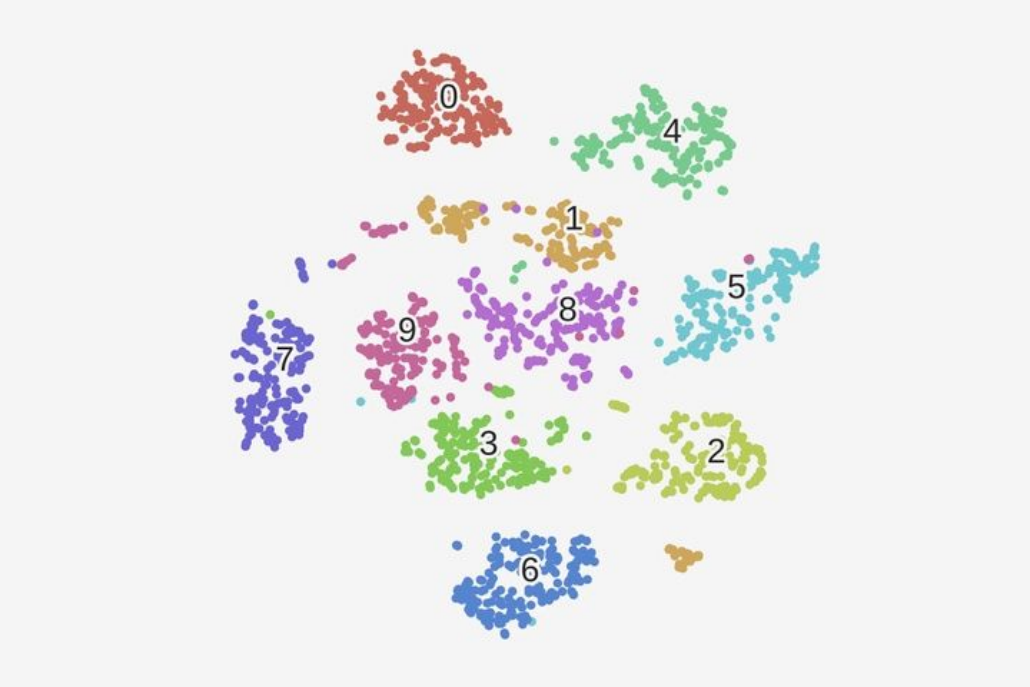
Уменьшение размерности: PCA

- Преобразует данные (возможно, коррелированные) в набор линейно независимых компонент (главных компонент).
- Первая компонента имеет максимальную дисперсию, вторая следующую по величине и т.д.
- Фактически находим базис в пространстве меньшей размерности.
- Очень широко применяемый метод.



Уменьшение размерности: t-sne

- An illustrated introduction to the t-SNE algorithm https://www.oreilly.com/learning/an-illustrated-introduction-to-the-t-sne-algorithm
- How to Use t-SNE Effectively http://distill.pub/2016/misread-tsne/



Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Обучение с частичным привлечением учителя

Вариант обучения с учителем (supervised learning), в котором также используются неразмеченные данные.

Например,

- Есть небольшой размеченный датасет (большой датасет д**о**рог или его не всегда можно получить).
- Есть большой неразмеченный датасет (неразмеченных данных много).
- По неразмеченным данным модель “учит” структуру и закономерности.
- По размеченным данным на уже выделенной структуре учим модель с использованием разметки.

Классы моделей в ML

- **Обучение с учителем** (supervised learning)
  - Классификация
  - Регрессия
  - Ранжирование
- **Обучение без учителя** (unsupervised learning)
  - Кластеризация
  - Уменьшение размерности
- **Обучение с частичным привлечением учителя** (semi-supervised learning)
- **Обучение с подкреплением** (reinforcement learning)

Обучение с подкреплением

Постановка задачи:

- Есть среда, в которой действует агент. У среды есть состояние.
- Агент может совершать действия.
- Действия приносят некий результат (reward) (не обязательно мгновенно)
- Надо научиться такой модели поведения, которая максимизирует результат

Типичные задачи:

- Управление роботом
- Оптимизация в играх

Процесс машинного обучения: #3. Метрики качества

Выбор метрики качества

Разные задачи подразумевают различные метрики качества.

4 класса метрик:

- **Бизнес-метрики** (KPI и другие важные метрики)
- **Онлайн-метрики** (доступны в работающей системе, можно использовать в эксперименте, например, среднее время просмотра сайта пользователем)
- **Офлайн-метрики** (оценка качества классификации, регрессии, …)
- **Метрики обучения** (функции потерь, внутренняя кухня алгоритмов)

На практике они крайне редко соответствуют друг другу.

Метрики качества регрессии

Общая идея: насколько хорошо вписывается в данные линия регрессии. Например, как далеко вокруг неё разбросаны все наблюдения:

- **MAE/MAD** (Mean Absolute Error, Mean Absolute Deviation) — средний модуль ошибки
- **MSE/MSD** (Mean Squared Error/Deviation) — среднеквадратическая ошибка
- **RMSE/RMSD** (Root Mean Squared Error) — лучше, чем MSE, потому что выражается в тех же единицах, что и измеряемая величина
- **MAPE/MAPD** (Mean Absolute Percentage Error) — ошибка в процентах от самой величины

Метрики качества регрессии

Mean squared error	$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$
Root mean squared error	$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
Mean absolute error	$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$
Mean absolute percentage error	$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n} \left\lvert \frac{e_t}{y_t} \right\rvert$

Метрики качества классификации

Идея: насколько хорошо мы угадываем классы. Важно, что цена ошибки может быть разной для разных случаев.

При классификации на два класса (по сути да/нет) у нас есть 4 различных исхода:

- True Positive (TP) -- истинное значение было “да” и мы предсказали “да”
- True Negative (TN) -- истинное значение было “нет” и мы предсказали “нет”
- False Positive (FP) -- истинное значение было “нет”, а мы предсказали “да”. Ложное срабатывание, ошибка I рода.
- False Negatie (FN) -- истинное значение было “да”, а мы предсказали “нет”. Пропуск цели, ошибка II рода.

		Предсказанный	
		true	**false**
Действительный	**true**	TP	FN
	false	FP	TN

Метрики качества классификации

- **Accuracy** (аккуратность) — процент верных предсказаний
- **Precision** (точность) — сколько верных среди предсказанных как “Класс 1/да”
- **Recall** (полнота) — сколько из настоящих “Класс 1/да” мы определили верно (то же самое, что True Positive Rate, Sensitivity)
- **F-мера** (гармоническое среднее P и R)
- **Log-loss** (учитывает значения вероятности)

...

$$precision(p.) = \frac{TP}{TP + FP}$$

$$recall(r.) = \frac{TP}{TP + FN}$$

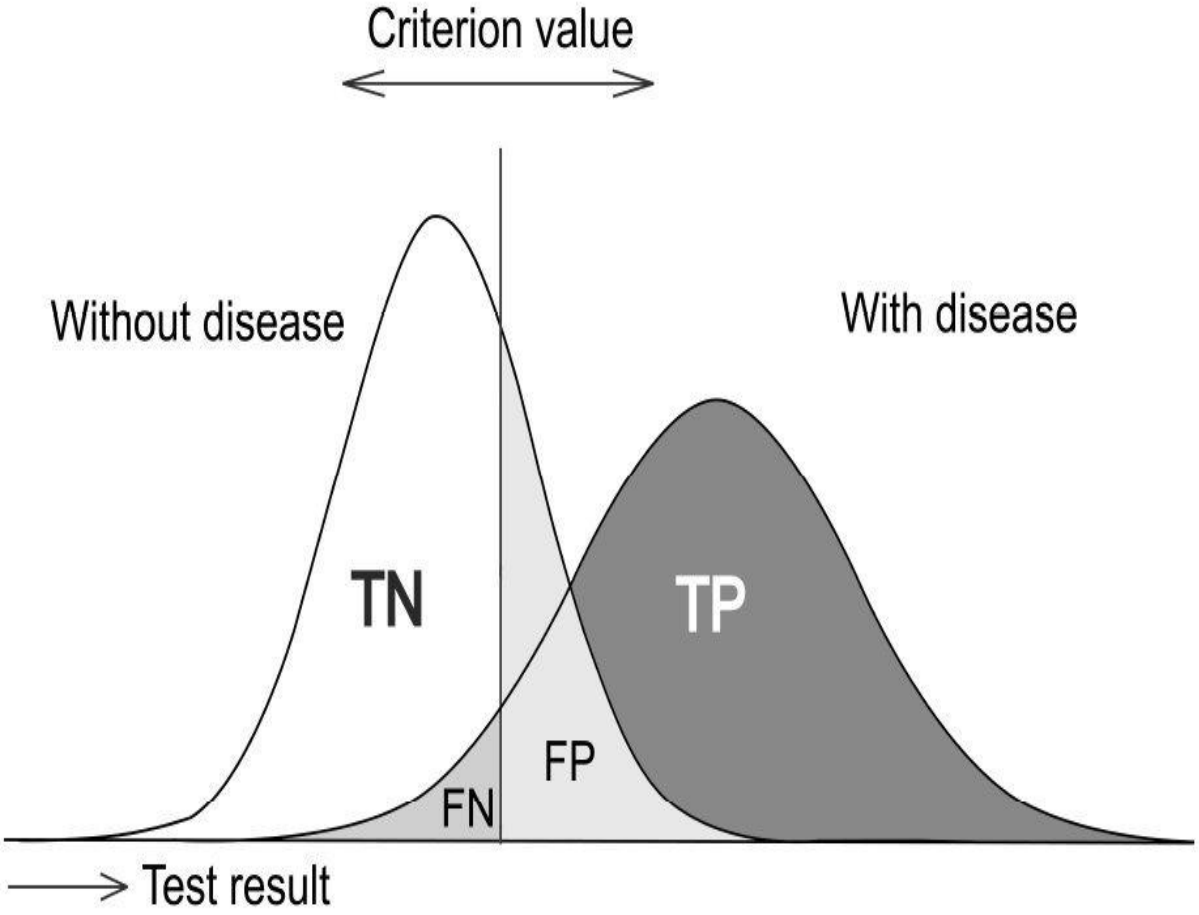
$$accuracy(acc.) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

		Предсказанный	
		true	**false**
Действительный	**true**	TP	FN
	false	FP	TN

Метрики качества классификации

В реальности у классификатора часто есть порог, который можно регулировать (тем самым играя на разнице в цене ошибок I и II рода)



ROC-кривая

- Поскольку у классификатора есть порог, мы можем получить общую картину для всех значений порога.
- Зависимость количества **верно классифицированных положительных примеров** (чувствительность, sensitivity, true positive rate, TPR, hit rate, recall) от количества **неверно классифицированных отрицательных примеров** (1-специфичность, fall-out, false positive rate, FPR).

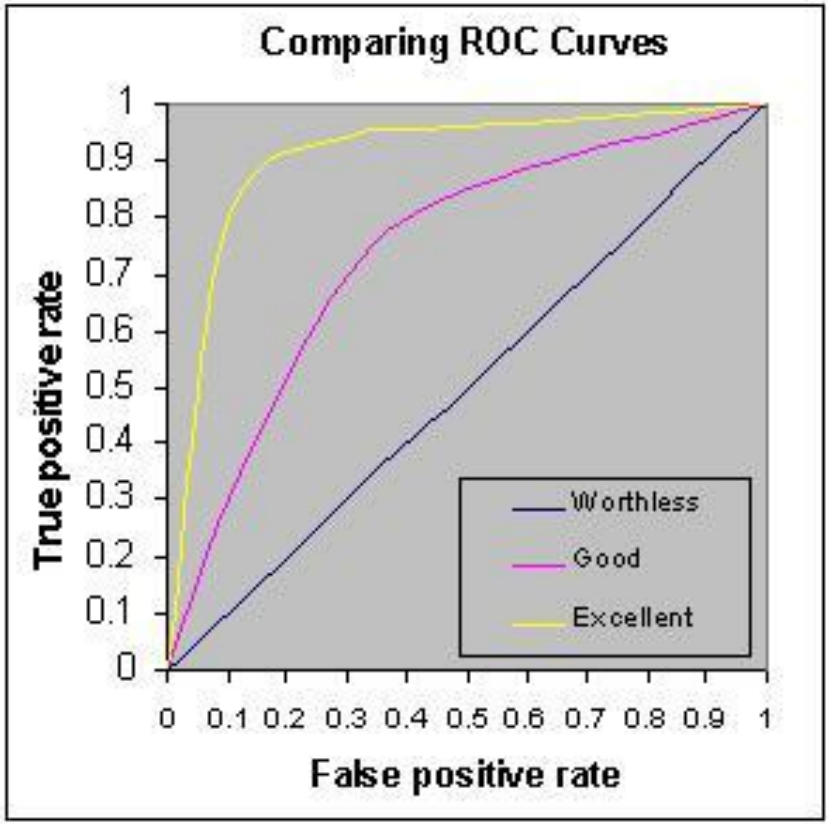
$$TPR = TP/P = TP/(TP + FN)$$

$$FPR = FP/N = FP/(FP + TN) = 1 - SPC$$

		Предсказанный	
		true	**false**
Действительный	**true**	TP	FN
	false	FP	TN

ROC-кривая

- Зависимость количества **верно классифицированных положительных примеров** от количества **неверно классифицированных отрицательных примеров**.
- Разные кривые можно сравнивать по площади, которую они ограничивают (ROC AUC).
- 0.5 -- случайный классификатор
  1.0 -- идеальный классификатор
  В промежутке между ними реальные



Матрица неточностей (Confusion matrix)

Когда классов больше двух.

	0.91	0.96	0.94	0.75	1.00	0.83	0.85	0.97	1.00	0.86	1.00	0.79	1.00	0.75	1.00	1.00	0.96	0.90	0.81	0.89	0.94	0.98	0.86	0.89	0.94	0.92	0.96
0.80		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0.95	1	94	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
1.00	2	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.29	3	0	0	6	0	0	3	2	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	3	0	2	0
1.00	4	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.50	5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	1	1
0.92	6	1	0	0	0	0	152	0	0	1	0	0	0	0	0	0	0	1	4	2	3	0	0	0	0	2	0
0.97	7	1	0	1	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	2	0
0.33	8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
0.97	9	0	0	0	0	0	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0.82	10	0	0	0	0	0	2	0	0	0	18	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0.87	11	0	0	0	0	0	0	0	0	0	0	34	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0
1.00	12	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.57	13	0	0	0	0	0	0	0	0	0	0	9	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0
0.63	14	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	3	0	0	0	0	0	0	0	0	0
0.50	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	1	0	0	0	0	0	0
0.77	16	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	47	0	1	3	4	0	0	2	0	1	0
0.87	17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	69	1	2	5	0	0	0	0	0	0
0.97	18	0	0	0	0	1	4	0	0	1	0	0	0	0	0	0	0	0	197	1	0	0	0	0	0	0	0
0.78	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	35	183	13	0	0	2	0	1	0
0.97	20	0	0	0	0	0	10	3	0	1	0	0	0	0	0	0	0	0	0	4	702	0	0	0	0	6	0
0.93	21	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	2	0	0	0
0.29	22	0	0	1	0	0	2	0	0	6	0	0	0	0	0	0	0	0	1	1	1	0	6	2	0	1	0
0.91	23	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	3	6	0	0	115	0	0	0
1.00	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0
0.93	25	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	4	5	0	0	0	1	196	0
0.98	26	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	78

Метрики качества кластеризации

- Внутренние метрики:
  - средняя похожесть объектов внутри кластера: должна быть высокой)
  - среднее расстояние между кластерами: должно быть выше между кластерами, чем внутри одного кластера (расстояние обычно обратная мера к похожести)

Метрики качества кластеризации

- Со внутренними метриками проблема, так как их часто невозможно перевести в бизнес-метрики, поэтому используют внешние метрики, например, оценку по размеченной части датасета:
  - Можно считать метрики аналогичные метрикам для классификации, считая, что целевое значение -- это номер кластера
  - Можно придумать свою метрику на размеченном датасете
  - Это требует времени (ручной труд на разметку), но даёт гораздо более практический и понятный результат

Метрики качества ранжирования

- MRR (Mean Reciprocal Rank)
- NDCG (Normalized Discounted Cumulative Gain)
- Precision@K, Recall@K
- Average Precision@K, Average Recall@K
- … и много других

Общая идея: надо учитывать порядок объектов и их позицию в списке.

Практические соображения по метрикам

1. Если возможно, старайтесь использовать одну и ту же метрику для обучения и оценки качества.
   - Но это не всегда возможно, так как по некоторым метрикам нельзя или трудно напрямую оптимизироваться (AUC, NDCG)

Практические соображения по метрикам

2. Внимательно и осознанно выбирайте метрику под ваш датасет.

- В случае сильно несбалансированных классов можно получить бесполезную метрику
- Пример: Поиск мошеннических транзакций
  - Датасет: 1 млн примеров, из них мошеннических транзакций 1%
  - Метрика: Аккуратность (Accuracy)
  - Классификатор: всех относим к классу “Честные”
  - Получаем качество 99%

Практические соображения по метрикам

3. Учитываете, что может быть сильно разная цена ошибки.

- Возможно, надо скорректировать метрику либо выбирать метрику исходя из этого знания.

4. Выбросы могут сильно исказить метрику (регрессия)

- Используйте устойчивую метрику (например, медиана устойчивее арифметического среднего) или фильтруйте выбросы.

Процесс машинного обучения: #4. Обучение модели

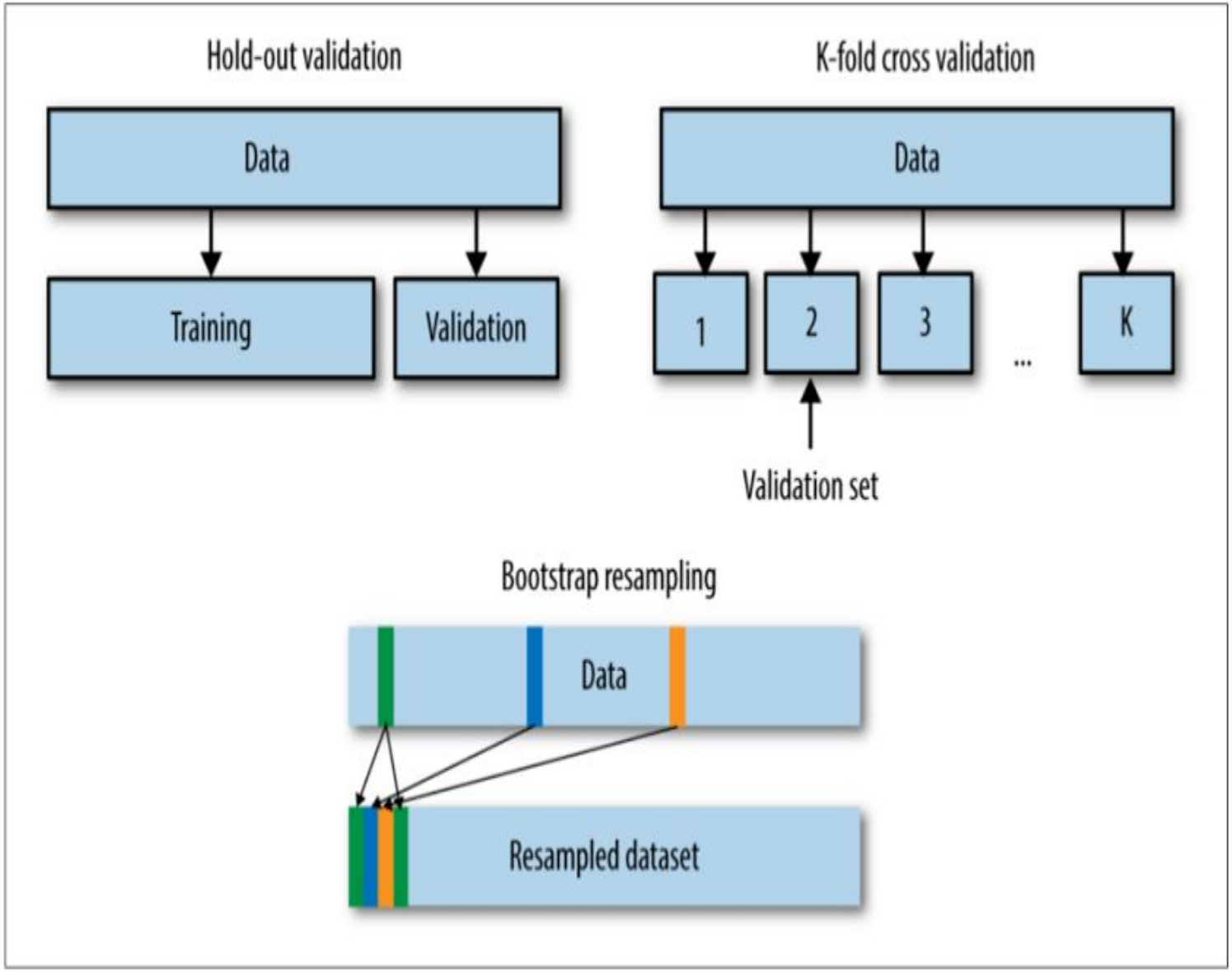
Разбиение датасета для обучения и оценки

- Проблема: если оценивать качество на той же выборке, на которой алгоритм обучался, оценка будет слишком оптимистичная и не соответствовать реальному качеству алгоритма.
- В предельном случае алгоритм “переобучается” и начинает отлично предсказывать примеры из своей выборки. Он их может “выучить” или запомнить (тогда новые примеры он может совсем не угадать), либо же находит закономерности в шуме, который всегда есть в данных (а шум на другой выборке примет иные значения, так что предсказание алгоритма будет неверным).
- Метод борьбы с этой проблемой: использовать отдельные наборы данных для обучения и оценки.

Разбиение датасета для обучения и оценки

- Скрытые данные (Hold-out validation):
  - train/validation/test датасеты
  - Обучаем модель на train
  - Выбираем модель по validation
  - Финальная проверка на test (используется только один раз для получения неоптимистичной оценки качества)
- Кросс-валидация (k-fold cross-validation)
  - Разбиваем данные на k кусочков
  - Обучаемся на (k-1) кусочке, проверяемся на 1
  - Результат усредняется
- Bootstrap resampling
  - Генерируем новый датасет, делая выборку из оригинального

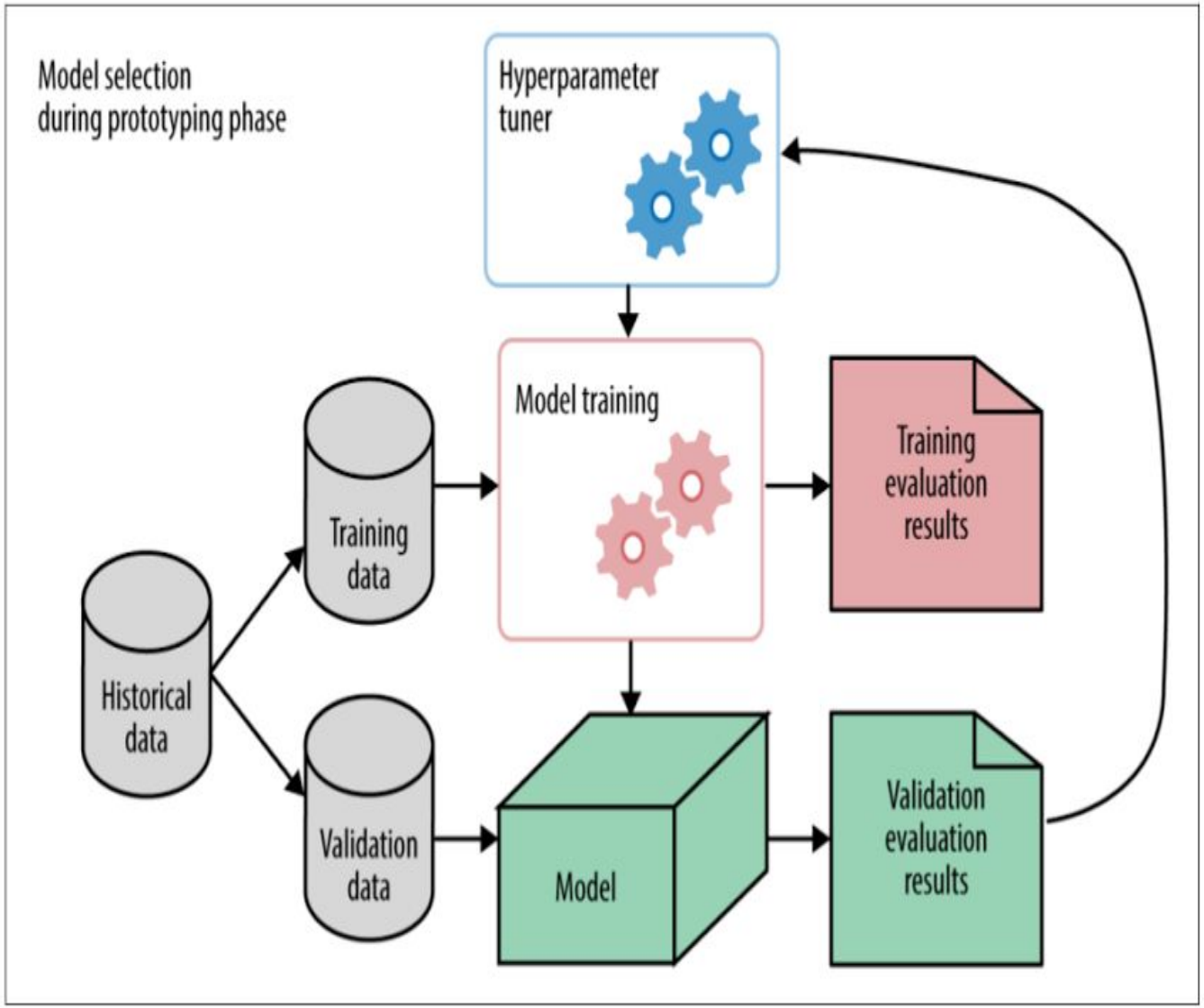
Механизмы валидации



Обучение модели

- Модель обучается на тренировочных данных (train set)
  - Обучение может означать подстройку каких-то численных параметров модели или (реже) изменение структуры модели.
  - В результате мы получаем параметры модели, которые позволяют ей (мы надеемся) эффективно выполнять свою функцию.
- Редко бывает только одна модель, обычно требуется пробовать разные варианты одной модели или модели различных классов
  - Для оценки качества и выбора лучшей модели мы используем валидационную выборку (validation set), на которой модель не обучалась
- В конце работы мы оцениваем модель на тестовой выборке (test set)
  - Это нужно для получения не слишком оптимистичной оценки качества модели
  - В идеале тестовую выборку нужно использовать только один раз!

Процесс создания модели ML



Полезные ресурсы

- Pedro Domingos, A Few Useful Things to Know about Machine Learning http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf
- Pedro Domingos, “The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World”, 2015 (на русском: “Верховный алгоритм. Как машинное обучение изменит наш мир”)
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016, http://www.deeplearningbook.org/
- “Evaluating Machine Learning Models. A Beginner's Guide to Key Concepts and Pitfalls”, 2015 http://www.oreilly.com/data/free/evaluating-machine-learning-models.csp
- http://eclass.cc/courselists/4_machine_learning
- http://eclass.cc/courselists/117_deep_learning
- http://eclass.cc/courselists/42_data_science

Спасибо!

https://ru.linkedin.com/in/grigorysapunov
grigory.sapunov@ieee.org
gs@inten.to