Технологии искусственного интеллекта: основные понятия, направления развития

Искусственный интеллект является одной из сквозных технологий, которые заложены в национальную программу «Цифровая экономика».

Сквозные технологии — это технологии, которые «проходят сквозь» разные отрасли и применяются в разных областях экономики, бизнеса и в жизни в целом.

Скажем, искусственный интеллект мы можем применять в нескольких сферах, например здравоохранении, образовании, строительстве, в сельском хозяйстве, в логистике и т. д. — значит, это сквозная технология. Работа с большими данными востребована во всех сферах деятельности — это сквозная технология.

В самом широком значении искусственным интеллектом называют способность компьютера решать те же интеллектуальные задачи, которые способен решать человек:

- **воспринимать и понимать мир через сенсоры (например, анализ изображений и звука);** https://play.boomstream.com/w1pLddcm
- **придумывать и создавать новые объекты (например, изображения, видео и тексты);** https://play.boomstream.com/06IluV8I
- **решать интеллектуальные задачи (например, играть в шахматы или го);**
- **переключаться между задачами и творчески решать сложные интеллектуальные задачи.** https://play.boomstream.com/REczzi2o

Во-первых, если говорить очень кратко, то искусственный интеллект – это достаточно широкая отрасль, которая в свою очередь охватывает и машинное и глубокое обучение. МО является подвидом ИИ, а ГО является подвидом МО.

Под ИИ подразумевается, что компьютер может выполнять такие задачи, которые может выполнять и человек. И здесь дело касается не просто каких-то механических действий, например, поднять и отнести какой-то предмет, а задачи, которые требуют интеллектуального мышления, то есть, когда надо принять правильное решение. Например, задача выиграть в шахматы, или распознать что изображено на картинке, или понять, что было произнесено собеседником и выдать правильный ответ.

Машинное обучение

Класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

Простыми словами, машинное обучение — это попытка научить компьютеры самостоятельно обучаться на большом количестве данных вместо работы на основе жестких и неизменных правил и алгоритмов.

Машинное обучение дает компьютерам возможность обучаться самостоятельно, практически без участия человека. Это становится возможным, если у системы есть доступ к большим объемам данных.

Одно из самых частых применений машинного обучения — распознавание объектов: изображений, речи и т. д. Например, по такому алгоритму работает система отметки на фото в социальной сети «ВКонтакте»: соцсеть сама может отметить на изображении вашего друга испросить у вас, он ли на этой фотографии. Эта же соцсеть предлагает распознавание речи в голосовых сообщениях (вы можете включить текстовую расшифровку, если прослушать сообщение сейчас не можете) и добавление автоматических субтитров на видео.
Такие же автоматические субтитры предлагают и другие ресурсы по просмотру видео, например известный видеохостинг YouTube.

Глубинное обучение
Иногда называют «глубокое обучение» (от англ. Deep Learning). Подобласть машинного обучения, где в качестве алгоритмов используются нейронные сети. Нейронная сеть (искусственная нейронная сеть) — это попытка воспроизведения работы человеческого мозга на компьютере при помощи слоев нейронов. Глубинное обучение позволяет обучать модель предсказывать результат по набору входных данных, например предсказывать цены на авиа-или ж/д билеты в зависимости от сезона.

Data Science

Это концепция объединения статистики, анализа данных, машинного обучения и связанных с ними методов для понимания и анализа реальных явлений.

Специалисты из этой области работают с массивами данных, чтобы алгоритмы ИИ могли функционировать эффективно и правильно. Data Scientist ищет в массивах данных связи и закономерности, которые позволят ему создать модель, предсказывающую результат, то есть понять, например, какие факторы нужно сравнивать и анализировать, какие данные закладывать в модель, чтобы с высокой точностью предсказывать, пойдет ли завтра дождь.

Data Mining

Широкое понятие, означающее извлечение знаний из данных.

Очень часто под Data Mining подразумевают методологию и процесс обнаружения новых данных в больших массивах данных, которые накопились в информационных системах какой-либо компании, были ранее неизвестны. Эти данные должны быть нетривиальны, практически полезны и доступны для интерпретации, чтобы быть полезными при принятии решений.

Большие данные

Большие данные предполагают не просто большой объем данных, они имеют определенные характеристики.

В 2001 году Meta Group был выработан набор VVV:

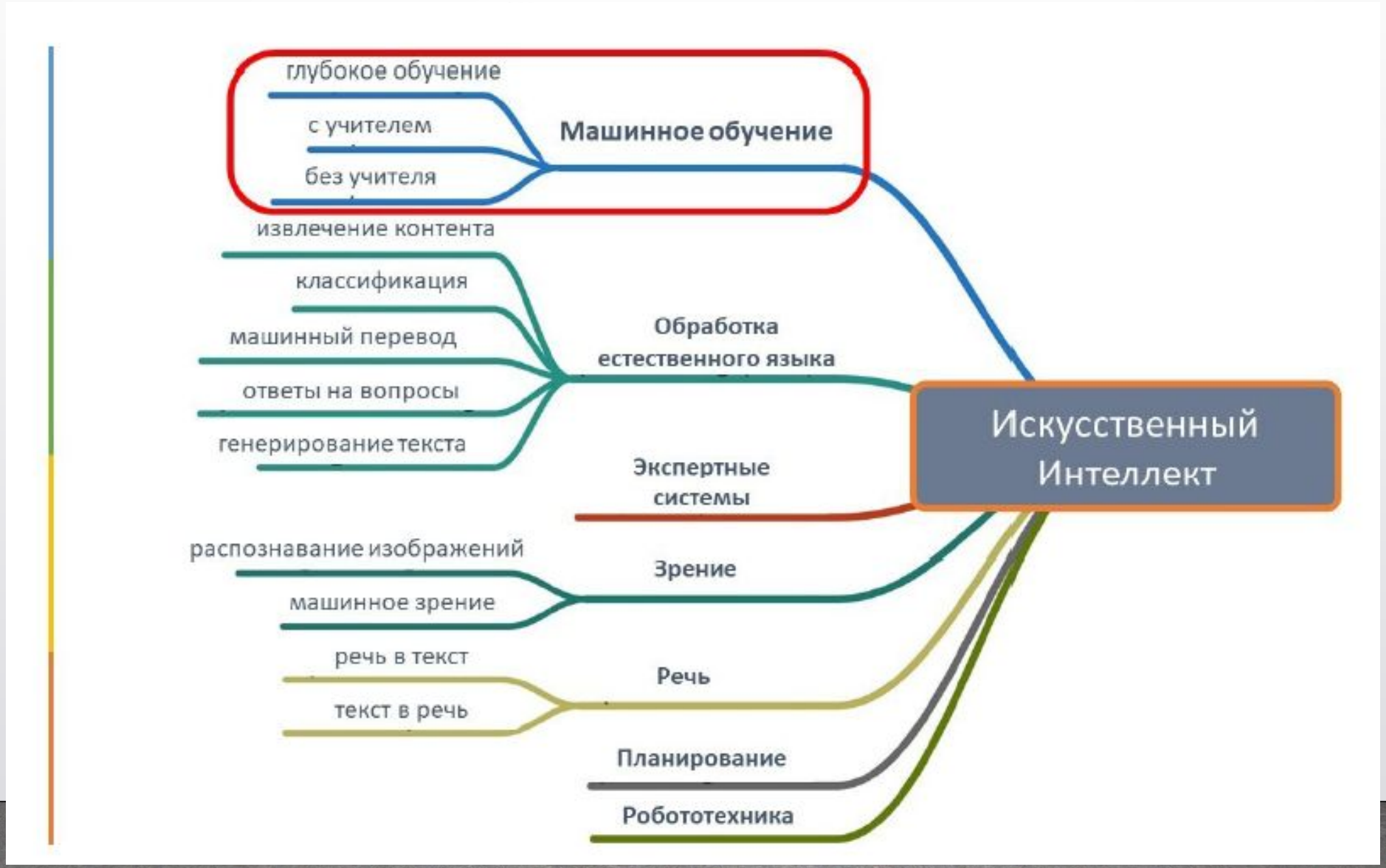
volume — большой объем;

velocity — регулярное обновление данных и постоянная их обработка;

variety — возможность одновременной обработки разных типов информации.

Позднее появились интерпретации с «четырьмя V» (добавлялась **veracity** — достоверность), «пятью V» (в этом варианте прибавляли **viability**— жизнеспособность, или value — ценность), и даже «семью V» (кроме всего, добавляли также **variability** — переменчивость, и **visualization**— визуализацию).

IDC (International Data Corporation, международная корпорация данных) интерпретирует «четвертое V» как value (ценность) с точки зрения экономической целесообразности.


Искусственный Интеллект
Машинное обучение
глубокое обучение
с учителем
без учителя
Обработка естественного языка
извлечение контента
классификация
машинный перевод
ответы на вопросы
генерирование текста
Экспертные системы
Зрение
распознавание изображений
машинное зрение
Речь
речь в текст
текст в речь
Планирование
Робототехника

Правила
Данные
Традиционное программирование
Ответы
Ответы
Данные
Машинное обучение
Правила

Например, если взять пример шахмат, то в примере с ИИ, мы даем машине много логических правил, и на их основе она учится играть. А в примере с МО, мы даем машине много примеров прошлых игр, она изучает их и анализирует почему одни игроки выигрывали, а другие проигрывали, какие шаги вели к успеху, а какие – к поражению. И на основе этих примеров, машина сама создает алгоритмы и правила как надо играть в шахматы, чтобы выиграть.

Другой пример, предположим, нам надо понять, как будет вести себя цена квартиры при изменении тех или иных параметров, например, в зависимости от площади, удаленности от метро, этажности и прочих факторов. Мы загружаем данные с разными квартирами, и компьютер создает модель, по которой можно будет предсказать цены в зависимости от этих факторов. Мы можем регулярно обновлять эти данные, и наш алгоритм будет обучаться на основе этих новых данных и каждый раз будет усовершенствовать свою точность по предсказанию цены в зависимости от параметров.

Методы машинного обучения: классическое обучение

Категории машинного обучения:

- с учителем (Supervised learning)
- без учителя (Unsupervised learning)

В первом случае у машины есть некий учитель, который говорит ей как правильно. Рассказывает, что на этой картинке кошка, а на этой собака. То есть учитель уже заранее разделил или, как мы говорим разметил все данные на кошек и собак, а машина учится на конкретных примерах.

При обучении без учителя, машине просто вываливают кучу фотографий животных на стол и говорят: «Разберись, кто здесь на кого похож». Данные не размечены, у машины нет учителя, и она пытается сама найти любые закономерности.

Методы машинного обучения: классическое обучение

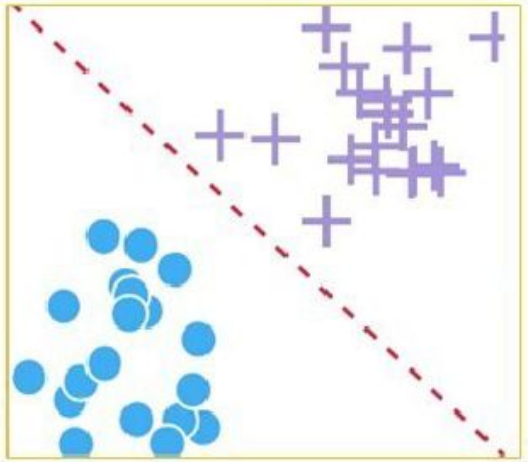
Задачи, которые ставятся при машинном обучении:

- классификация — предсказание категории объекта;
- регрессия — предсказание места на числовой прямой

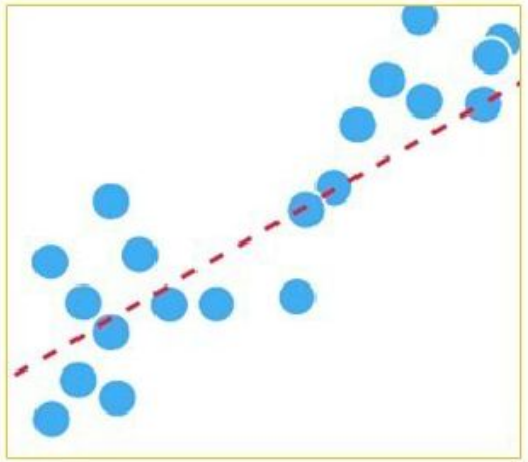
Регрессия
Классификация
Обучение с учителем
Кластеризация
Обучение без учителя
Снижение размерности
Машинное Обучение

Обучение с учителем (supervised)

Классификация



Регрессия



Задача классификации

Это будет пример цветков ириса Фишера. Этот набор данных стал уже классическим, и часто используется для иллюстрации работы различных статистических алгоритмов. (https://gist.github.com/curran/a08a1080b88344b0c8a7)

В природе существует три вида цветков ириса. Они отличаются друг от друга размерами лепестка и чашелистника. Все данные по цветкам занесены в таблицу, в столбиках указаны длина и ширина лепестка, а также длина и ширина чашелистника. В последнем столбце указан вид ириса – Ирис щетинистый (*Iris setosa*), Ирис виргинский (*Iris virginica*) и Ирис разноцветный (*Iris versicolor*). Тот или иной вид ириса и является в нашем случае меткой.

Обучение с учителем (supervised)

Задача классификации

sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica

Метки (labels) – указывают какому классу ирисов принадлежит тот или иной цветок, в зависимости от размеров лепестка и чашелистника

https://gist.github.com/curran/a08a1080b88344b0c8a7#file-iris-csv

Iris setosa

Iris virginica

Iris versicolor

Длина чашелистика	sepal_width	Длина лепестка	лепесток_width	виды
5,1	3.5	1.4	0.2	setosa
4,9	3.0	1.4	0.2	setosa
4,7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	разноцветный
6.4	3.2	4.5	1.5	разноцветный
6.9	3.1	4.9	1.5	разноцветный
3,4	3.4	1.4	0.3	setosa
3,4	3.4	1.5	0.2	setosa
2,9	2.9	1.4	0.2	setosa
3,1	3.1	1.5	0.1	setosa
3,7	3.7	1.5	0.2	setosa

Формула и диаграмма цветка.

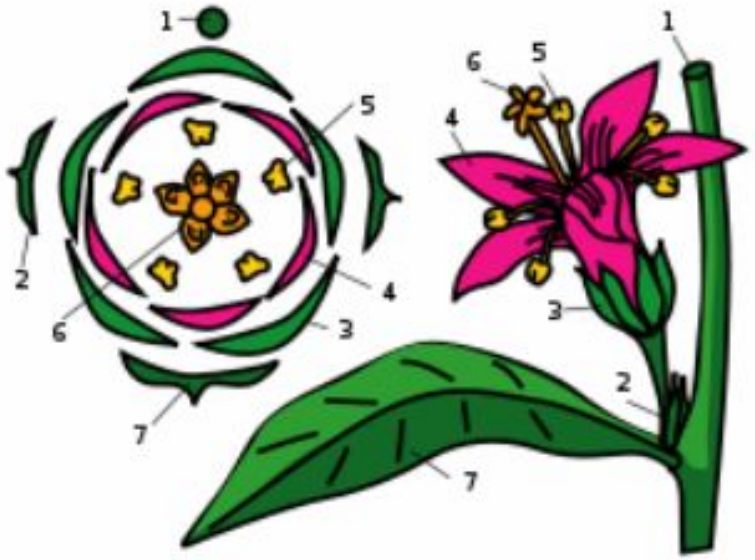


Диаграмма цветка. 1 - ось соцветия, 2 - прицветники, 3 - чашелистики, 4 - лепесток, 5 - тычинка, 6 - пестик, 7 - кроющий лист.

Iris setosa

Iris virginica

Iris versicolor

На основании этого набора данных требуется построить правило классификации, определяющее вид растения в зависимости от размеров. Это задача многоклассовой классификации, так как имеется три класса – три вида ириса.

В данном случае с помощью алгоритма классификации, мы разделяем наши ирисы на три вида в зависимости от длины и ширины лепестка и чашелистника. В следующий раз, если нам попадется новый представитель ирисов, с помощью нашей модели мы сможем сразу же его поместить в тот или иной из трех классов.

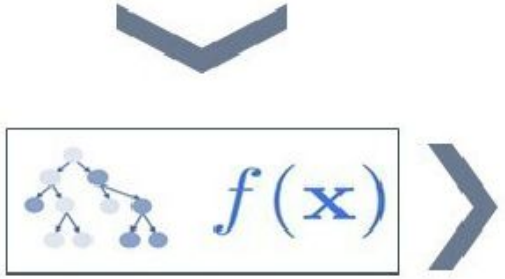
Обучение с учителем (supervised)

Задача классификации

sepal_length	sepal_width	petal_length	petal_width	species
6.3	4.5	1.4	0.4	?
4.9	3.0	3.2	0.2	?
4.7	3.2	3.1	0.5	?
5.1	3.2	5.1	1.4	?
6.4	3.0	4.4	2.5	?
5.9	2.1	4.9	1.5	?
6.3	3.3	5.2	2.0	?

Загружаем **новые данные** чтобы узнать к какому классу относятся те или иные цветки

sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica



sepal_length	sepal_width	petal_length	petal_width	species
6.3	4.5	1.4	0.4	versicolor
4.9	3.0	3.2	0.2	setosa
4.7	3.2	3.1	0.5	setosa
5.1	3.2	5.1	1.4	virginica
6.4	3.0	4.4	2.5	versicolor
5.9	2.1	4.9	1.5	setosa
6.3	3.3	5.2	2.0	virginica

Загружаем labeled data в компьютер для обучения

Компьютер создает модель на основе данных

Компьютер предсказывает вид цветка на основе созданной им модели

length	width	species
5,1	3,8	setosa
4,8	3	setosa
5	3,5	setosa
4,7	3,2	setosa
5,3	3,7	setosa
6,2	2,2	versicolor
5,6	2,5	versicolor
5,9	3,2	versicolor
6,1	2,8	versicolor
6,3	2,5	versicolor
7,7	2,6	virginica
6,9	3,2	virginica
6,3	2,7	virginica
7,2	3,2	virginica
7,3	2,9	virginica
5,7	2,9	

	A	B	C	D	E	F	G
1	length	width	species				Евклидово расстояни
2	5,1	3,8	setosa				1,081665383
3	4,8	3	setosa				0,905538514
4	5	3,5	setosa				0,921954446
5	4,7	3,2	setosa				1,044030651
6	5,3	3,7	setosa				0,894427191
7	6,2	2,2	versicolor				0,860232527
8	5,6	2,5	versicolor				0,412310563
9	5,9	3,2	versicolor				0,360555128
10	6,1	2,8	versicolor				0,412310563
11	6,3	2,5	versicolor				0,721110255
12	7,7	2,6	virginica				2,022374842
13	6,9	3,2	virginica				1,236931688
14	6,3	2,7	virginica				0,632455532
15	7,2	3,2	virginica				1,529705854
16	7,3	2,9	virginica				1,6
17							
18	5,7	2,9					
19							

	A	B	C	D	E	F	G	H	I	J	K
1	length	width	species			Ранг	Евклидово расстояние			k	species
2	5,1	3,8	setosa			11	1,081665383	setosa		1	versicolor
3	4,8	3	setosa			8	0,905538514	setosa		2	versicolor
4	5	3,5	setosa			9	0,921954446	setosa		3	versicolor
5	4,7	3,2	setosa			10	1,044030651	setosa		4	virginica
6	5,3	3,7	setosa			7	0,894427191	setosa		5	versicolor
7	6,2	2,2	versicolor			6	0,860232527	versicolor		6	versicolor
8	5,6	2,5	versicolor			3	0,412310563	versicolor		7	setosa
9	5,9	3,2	versicolor			1	0,360555128	versicolor		8	setosa
10	6,1	2,8	versicolor			2	0,412310563	versicolor		9	setosa
11	6,3	2,5	versicolor			5	0,721110255	versicolor		10	setosa
12	7,7	2,6	virginica			15	2,022374842	virginica		11	setosa
13	6,9	3,2	virginica			12	1,236931688	virginica		12	virginica
14	6,3	2,7	virginica			4	0,632455532	virginica		13	virginica
15	7,2	3,2	virginica			13	1,529705854	virginica		14	virginica
16	7,3	2,9	virginica			14	1,6	virginica		15	virginica
17											
18	5,7	2,9									

	A	B	C	D	E	F	G	H	I	J	K
1	length	width	species			Ранг	Евклидово расстояние			k	species
2	5,1	3,8	setosa			2	0,848528137	setosa		1	setosa
3	4,8	3	setosa			7	1,664331698	setosa		2	setosa
4	5	3,5	setosa			3	1,140175425	setosa		3	setosa
5	4,7	3,2	setosa			5	1,562049935	setosa		4	versicolor
6	5,3	3,7	setosa			1	0,806225775	setosa		5	setosa
7	6,2	2,2	versicolor			14	2,256102835	versicolor		6	versicolor
8	5,6	2,5	versicolor			10	1,902629759	versicolor		7	setosa
9	5,9	3,2	versicolor			4	1,216552506	versicolor		8	virginica
10	6,1	2,8	versicolor			6	1,64924225	versicolor		9	virginica
11	6,3	2,5	versicolor			12	1,992485885	versicolor		10	versicolor
12	7,7	2,6	virginica			15	2,690724809	virginica		11	virginica
13	6,9	3,2	virginica			8	1,697056275	virginica		12	versicolor
14	6,3	2,7	virginica			9	1,802775638	virginica		13	virginica
15	7,2	3,2	virginica			11	1,920937271	virginica		14	versicolor
16	7,3	2,9	virginica			13	2,19317122	virginica		15	virginica
17											
18	5,7	4,4	setosa								

Задача регрессии

Классический пример регрессии – это когда мы предсказываем цену квартиры в зависимости от ее площади.

Опять же мы имеем какую-то таблицу с данными разных квартир. В одном столбце площадь, а в другом – цены на эти квартиры. Это очень упрощенный пример регрессии, естественно, что цена квартиры будет зависеть от множества других факторов, но все же он наглядно демонстрирует, что такое регрессия. Так вот, в последнем столбце мы расположили фактические или реальные цены на квартиры с таким метражом.

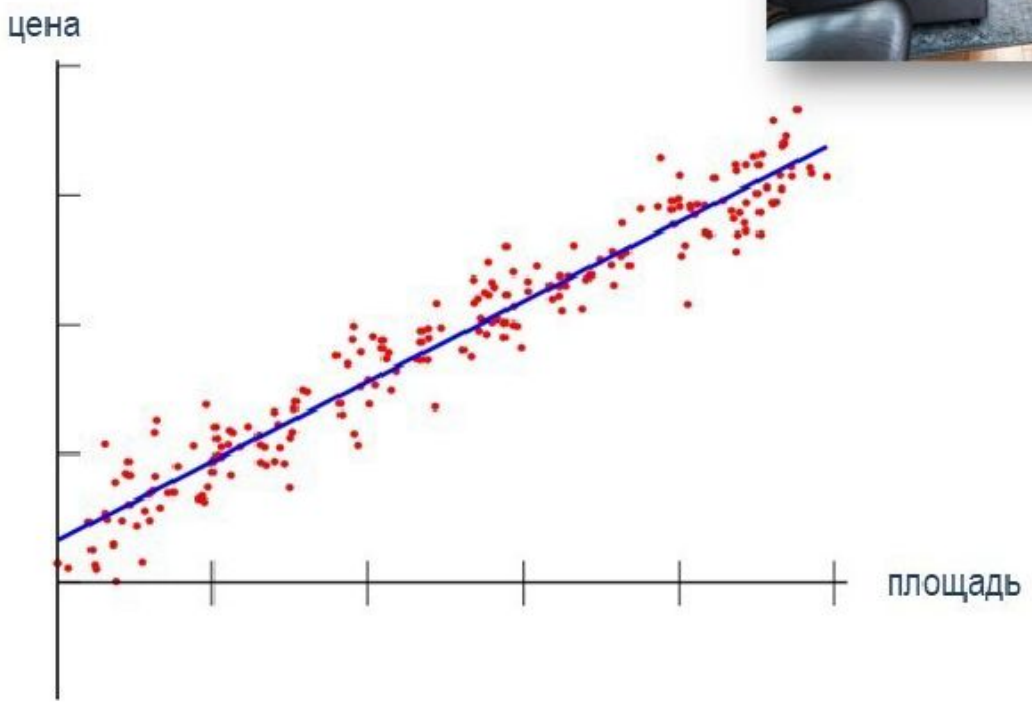
То есть, мы как учитель, показываем нашей модели, что вот, если видишь, что метраж такой-то, то цена будет такая-то и т.д. На основе этих данных модель учится, и потом выдает алгоритм, на основе которого мы можем предсказывать, какая будет цена квартиры, если условная площадь будет такая-то.

Обучение с учителем (supervised)

Задача регрессии

Площадь (кв.м.)	Фактическая цена (млн. руб.)
28	2,4
42	3,7
45	3,9
56	4,5
68	5,7
75	6,4
90	7,8

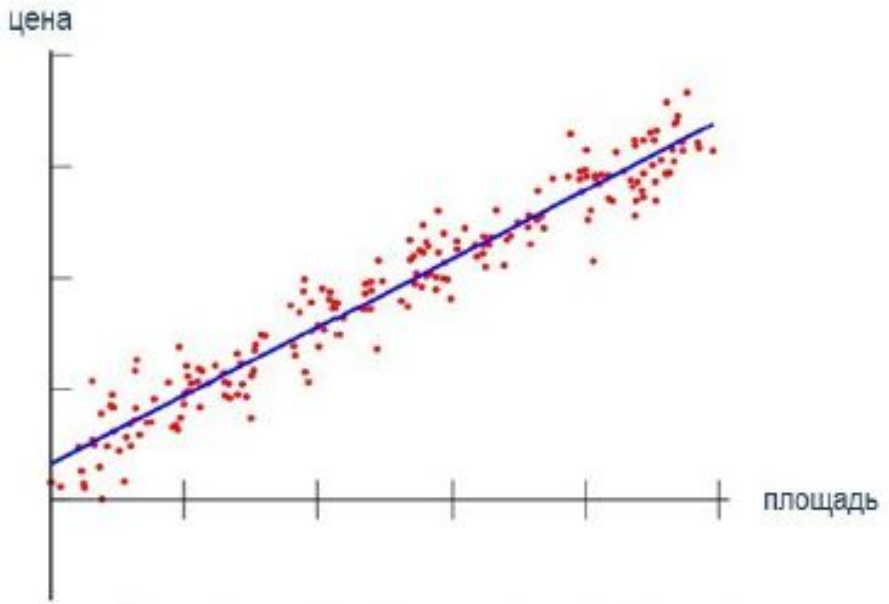
Labeled data – даем компьютеру ответы (реальные цены), на основе которых он будет строить модель для предсказывания цены в зависимости от площади



Компьютер выстраивает регрессию на основе множества имеющихся данных и может делать прогнозы цен для новых квартир

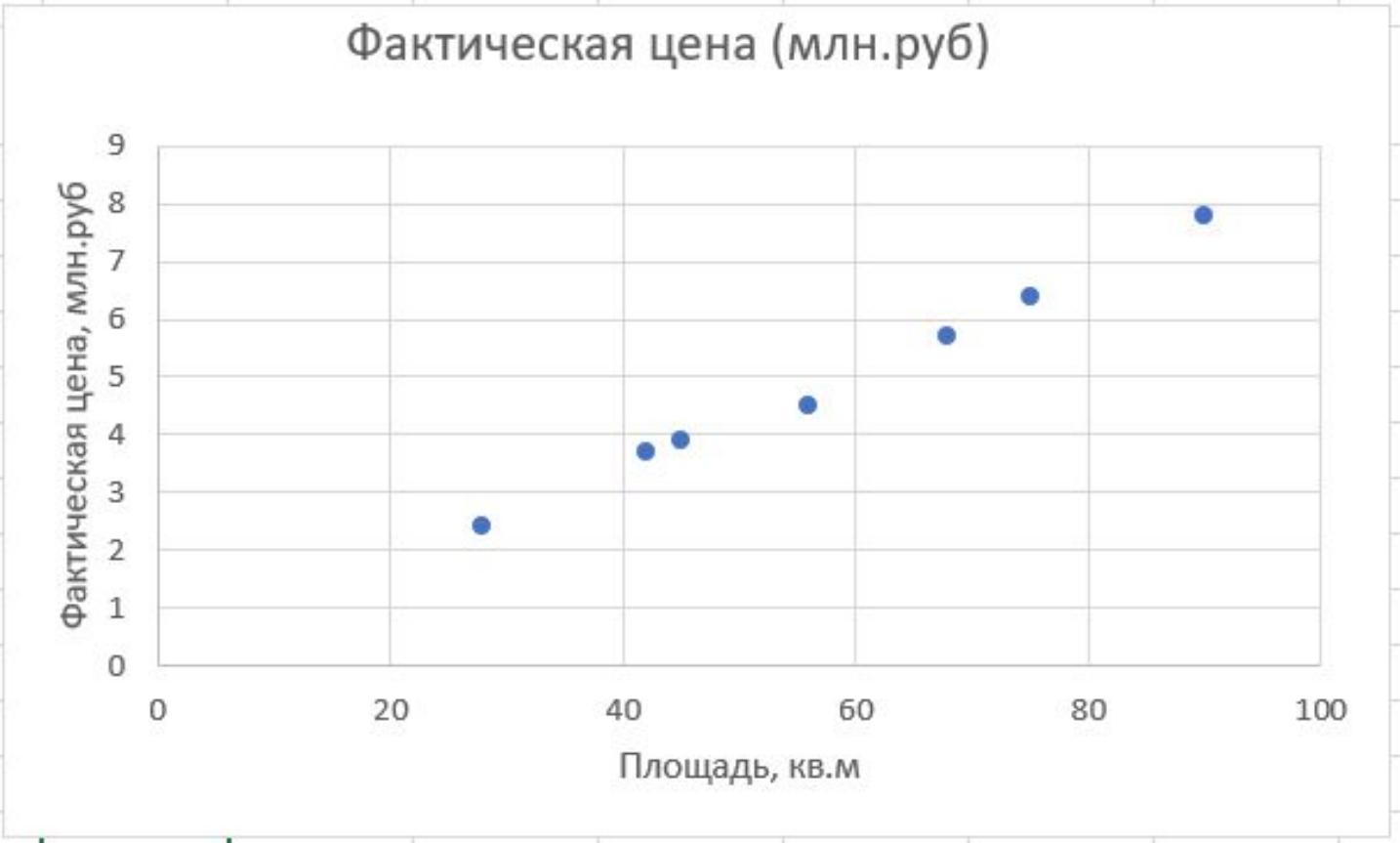
Пример Машинного обучения

Стоимость квартиры в зависимости от площади

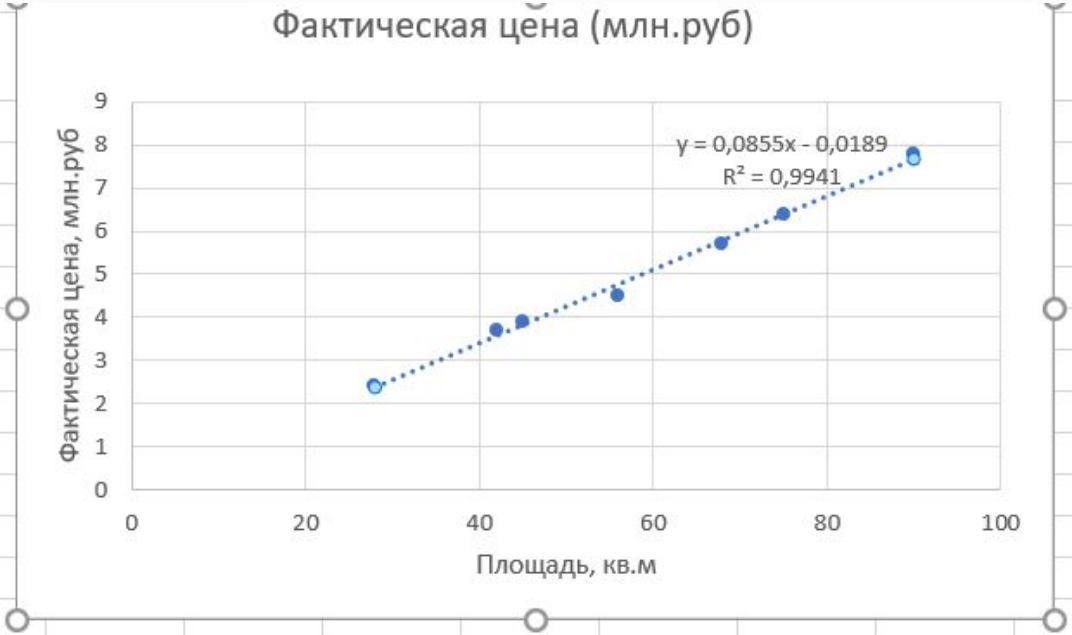


Компьютер выстраивает регрессию на основе множества имеющихся данных и может делать прогнозы цен для новых квартир

Площадь (кв.м)	Фактическая цена (млн.руб)
28	2,4
42	3,7
45	3,9
56	4,5
68	5,7
75	6,4
90	7,8



Площадь (кв.м)	Фактическая цена (млн.руб)
28	2,4
42	3,7
45	3,9
56	4,5
68	5,7
75	6,4
90	7,8



Параметры линии тренда

Параметры линии тренда

Экспоненциальная

Линейная

Логарифмическая

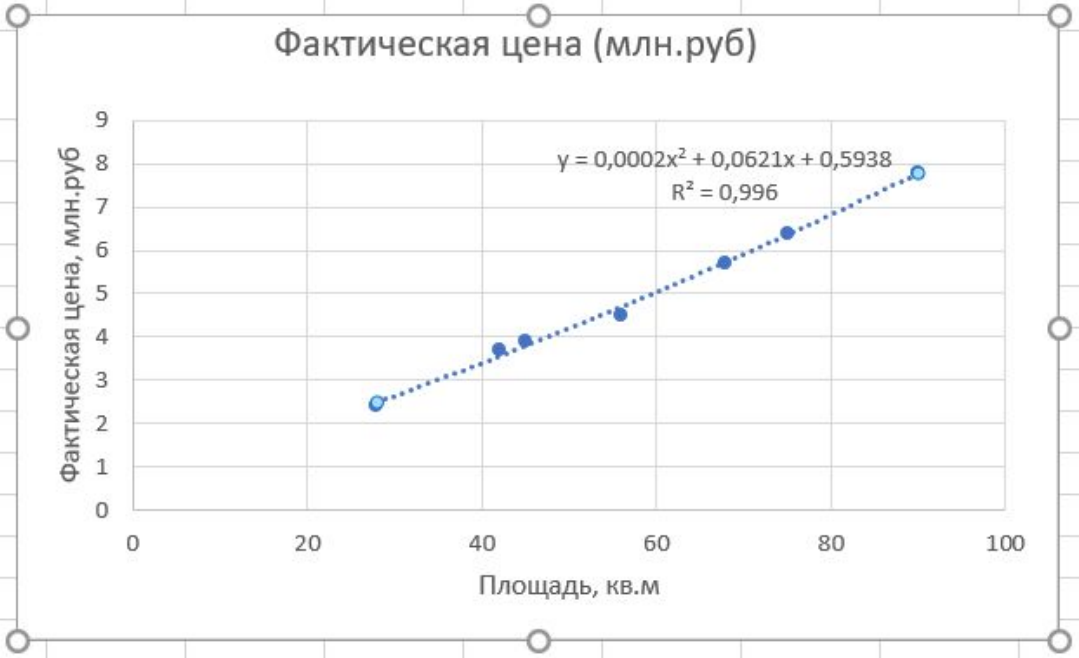
Полиномиальная Степень

Степенная

Скользящее среднее Период

Название линии тренда

Площадь (кв.м)	Фактическая цена (млн.руб)
28	2,4
42	3,7
45	3,9
56	4,5
68	5,7
75	6,4
90	7,8



Формат линии тренда

Параметры линии тренда

Параметры линии тренда

Экспоненциальная

Линейная

Логарифмическая

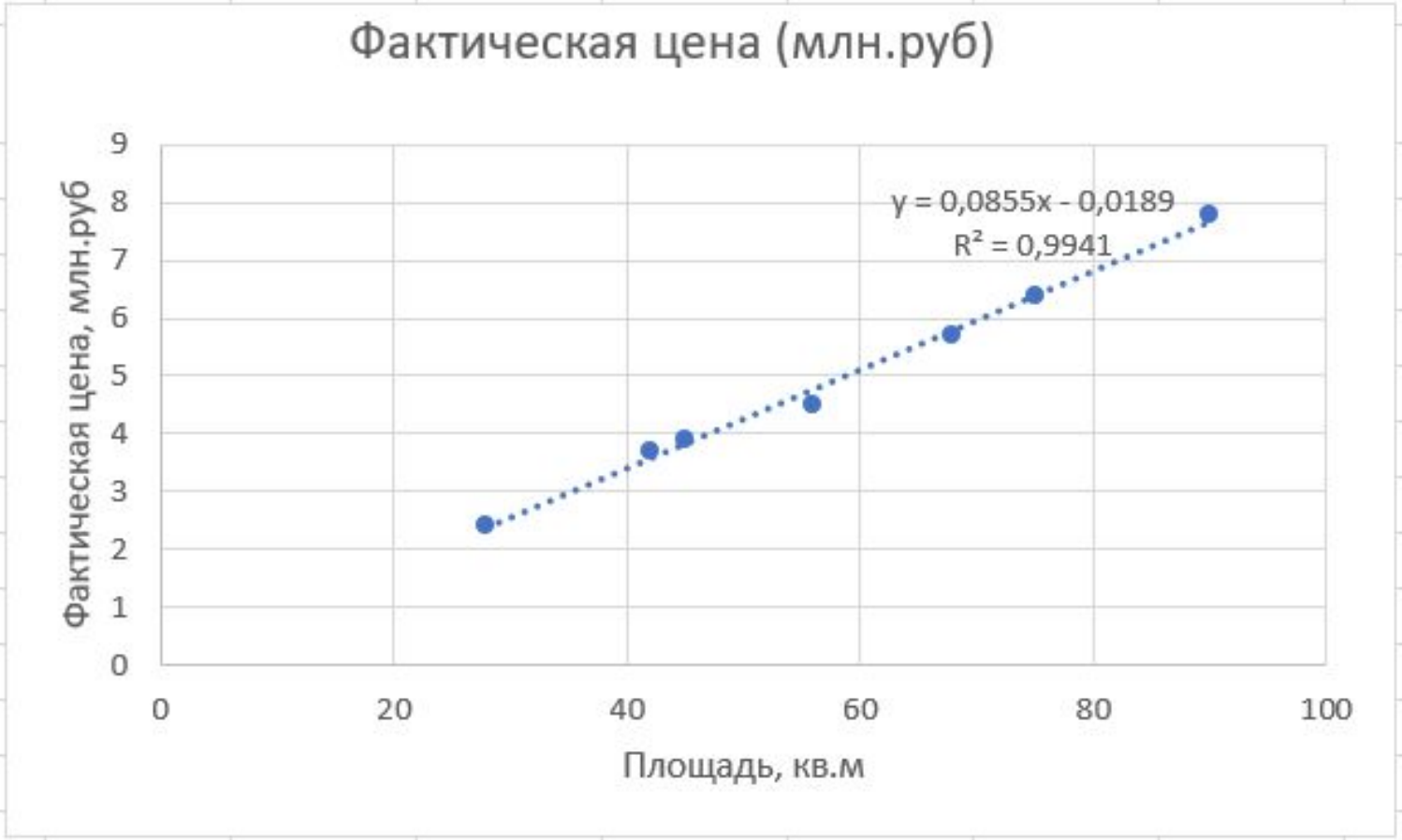
Полиномиальная

Степенная

Скользящее

Площадь (кв.м)	Фактическая цена (млн.руб)
28	2,4
42	3,7
45	3,9
56	4,5
68	5,7
75	6,4
90	7,8

34	2,8881
49	4,1706



Второй вид машинного обучения – это **обучение без учителя**. Это когда мы позволяем нашей модели обучаться самостоятельно и находить информацию, которая может быть не видна для человека.

В отличие от обучения с учителем, модели, которые используются в обучении без учителя, выводят закономерности и выводы на основе немаркированных данных (или unlabeled data).

Задачи классификации можно решить с помощью разных методов. Наиболее часто используемыми являются следующие:
– Дерево решений
– Логистическая регрессия В логистической регрессии немного по-другому, с помощью алгоритмов мы находим такую линию, которая разделяет наш набор данных на классы.
– Случайный лес
– Ансамбли и бэггинг
– Метод опорных векторов
– Метод К-ближайших соседей

Регрессия
Классификация
Обучение с учителем
Кластеризация
Обучение без учителя
Снижение размерности
Машинное Обучение

Задача кластеризации

В задачах **кластеризации** у нас имеется набор объектов и нам надо выявить его внутреннюю структуру. То есть нам надо найти группы объектов внутри этого набора, которые наиболее похожи между собой, и отличаются от других групп объектов из этого же набора.

Например, разобрать все движущиеся средства по категориям, например, все средства, похожие на велосипед, в одну группу или кластер, а похожие на автобус – в отдельную группу.

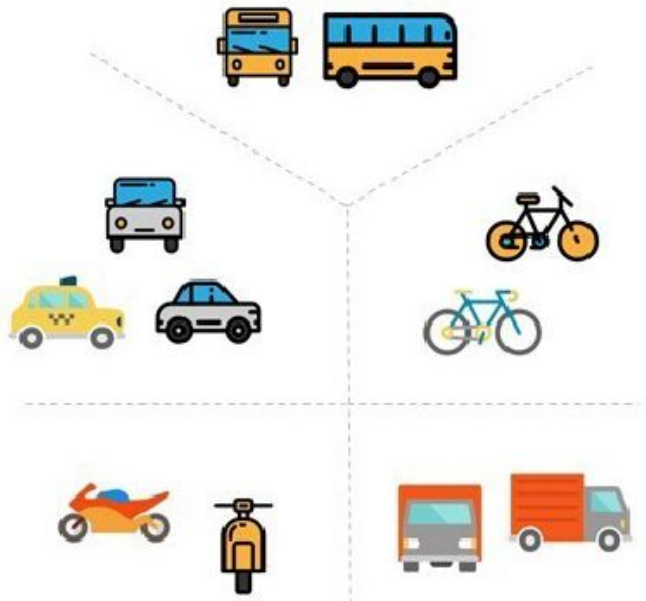
Причем, мы не говорим компьютеру, что чем является, он должен самостоятельно найти схожие признаки и определить похожие объекты в ту или иную группу. Поэтому это и называется обучение без учителя, потому что мы не говорим изначально компьютеру к какой группе принадлежат те или иные объекты.

Задача кластеризации

Обучение без учителя (unsupervised)

Модель обучается с помощью
неразмеченных данных (unlabeled data)

unlabeled data



Такие задачи бывают очень полезны для крупных ритейлеров, если они, например, хотят понять из кого состоят их клиенты. Предположим, есть крупный гипермаркет, и чтобы делать точечные рекламные акции для своих потребителей, ему необходимо будет разбить их по группам или кластерам. И если сейчас акция на спортивные товары, то отправлять информацию об этой акции не всем подряд потребителям, а только тем, кто в прошлом уже покупали спортивные товары.

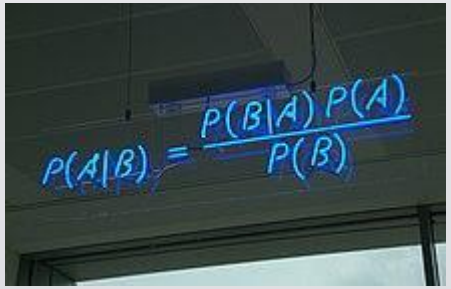
Методы машинного обучения: классическое обучение

Пример метода

Алгоритм наивного байесовского классификатора

Ба́йесовская фильтра́ция спа́ма — метод для фильтрации спама, основанный на применении наивного байесовского классификатора, опирающегося на прямое использование теоремы Байеса. Теорема Байеса названа в честь её автора Томаса Байеса (1702—1761) — английского математика и священника, который первым предложил использование теоремы для корректировки убеждений, основываясь на обновлённых данных.

Теорема Байеса - одна из основных теорем элементарной теории вероятностей, которая позволяет определить вероятность какого-либо события при условии, что произошло другое статистически взаимозависимое с ним событие.



Методы машинного обучения: классическое обучение

Первой известной программой, фильтрующей почту с использованием байесовского классификатора, была программа iFile Джейсона Ренни, выпущенная в 1996 году.

При обучении фильтра для каждого встреченного в письмах слова высчитывается и сохраняется его «вес» — оценка вероятности того, что письмо с этим словом — спам. В простейшем случае в качестве оценки используется частота: «появлений в спаме / появлений всего». В более сложных случаях возможна предварительная обработка текста: приведение слов в начальную форму, удаление служебных слов, вычисление «веса» для целых фраз, транслитерация и прочее.

При проверке вновь пришедшего письма вероятность «спамовости» вычисляется по указанной выше формуле для множества гипотез. В данном случае «гипотезы» — это слова, и для каждого слова «достоверность гипотезы» $P(A_i) = N_{word_i} / N_{words\ total}$ — доля этого слова в письме, а «зависимость события от гипотезы» $P(B \mid A_i)$ — вычисленный ранее «вес» слова. То есть «вес» письма в данном случае — усреднённый «вес» всех его слов.

Отнесение письма к «спаму» или «не-спаму» производится по тому, превышает ли его «вес» некую планку, заданную пользователем (обычно берут 60-80 %). После принятия решения по письму в базе данных обновляются «веса» для вошедших в него слов.

Методы машинного обучения: классическое обучение

Первой известной программой, фильтрующей почту с использованием байесовского классификатора, была программа iFile Джейсона Ренни, выпущенная в 1996 году.

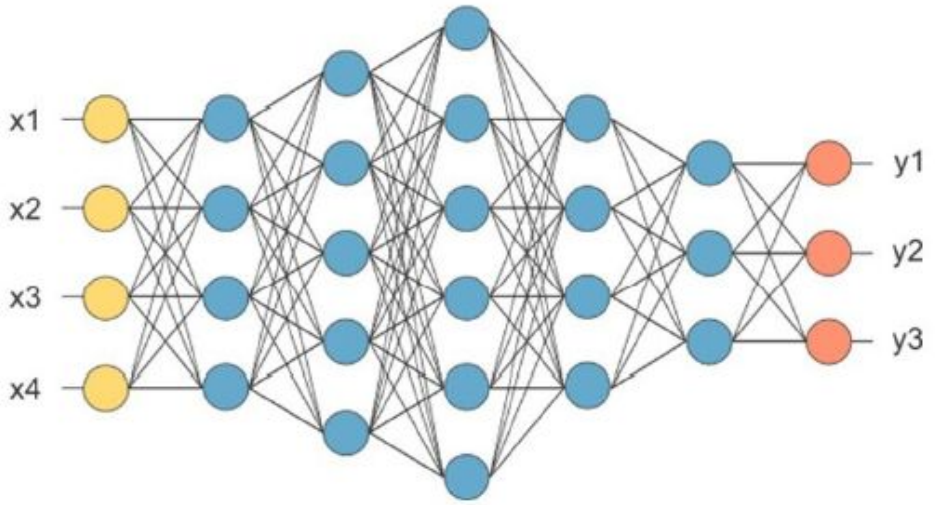
Алгоритм:

- машина считает, сколько раз какое-то слово встречается в спаме, а сколько раз в нормальных письмах;
- перемножает эти две вероятности по формуле Байеса;
- складывает результаты всех слов

Позже спамеры научились обходить фильтр Байеса, просто вставляя в конец письма много слов с «хорошими» рейтингами.

Идем дальше. Глубокое обучение – это подотрасль МО, то есть здесь тоже компьютер обучается, но обучается немного по-другому, чем в стандартном МО. В ГО используются нейронные сети (НС), которые представляют собой алгоритмы, повторяющие логику нейронов человеческого мозга. Большие объемы данных проходят через эти нейронные сети, и на выходе выдаются уже готовые ответы. Нейронные сети намного сложнее, чем обычное машинное обучение, и мы можем не всегда понимать, какие факторы имеют больший вес на тот или иной ответ, но использование нейронных сетей также помогает решать очень запутанные задачи в наше время. Иногда нейронные сети называют даже черным ящиком, потому что мы не всегда можем понять, что происходит внутри этих сетей.

Глубокое обучение (Deep Learning)



Глубокое обучение –
использование нейронных сетей в машинном обучении

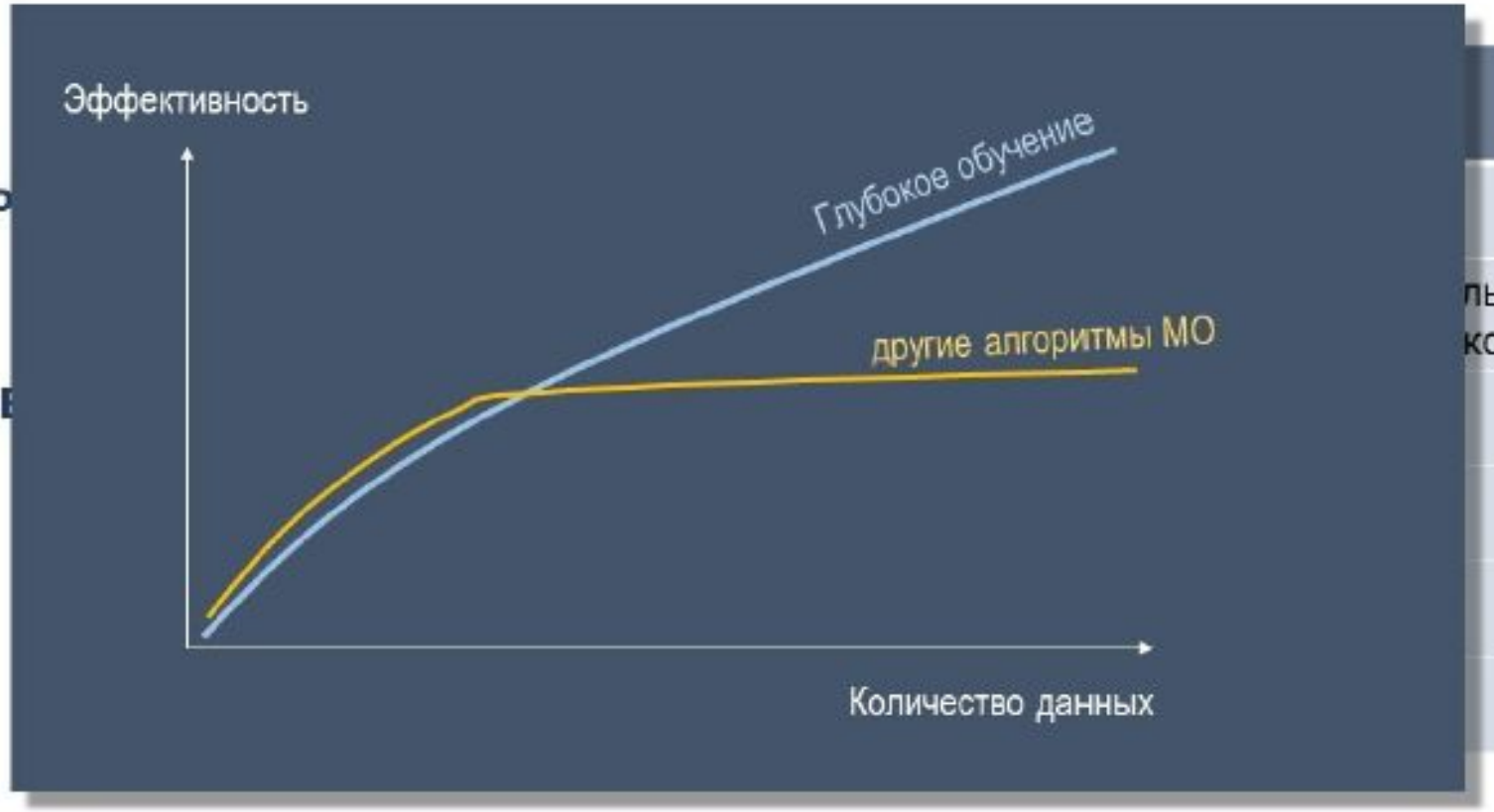
Нейронные сети –
алгоритмы, имитирующие логику человеческого мозга

какой узел НС был активирован, и как эти узлы вели себя вместе, чтобы прийти к этому результату. Если же вы используете МО, например, алгоритм «дерево решений», то там видно какой фактор сыграл решающую роль в определении качества эссе.

Нейронные сети были известны еще в 20 веке, но тогда они были не настолько глубокими, там был всего один или два слоя, и они не давали таких хороших результатов, как другие алгоритмы МО. Поэтому на какое-то время они отошли на второй план. Однако они стали популярны в последнее время, особенно примерно с 2006 года, когда появились огромные наборы данных и сильные компьютерные мощности, в частности, видео карты и мощные процессоры, которые стали способны создавать более глубокие слои НС и делать вычисления более эффективно.

По этим же причинам, ГО является достаточно дорогим. Потому что, во-первых, сложно собрать большие данные по определенным признакам и, во-вторых, серьезные вычислительные способности компьютеров – тоже достаточно дорогое удовольствие.

Сравнение


Эффективность
Глубокое обучение
другие алгоритмы МО
Количество данных

Если вкратце, то каким образом работает ГО.
Предположим, наша задача вычислить
сколько единиц транспорта и какой именно транспорт
(то есть автобусы, грузовики, машины
или велосипеды) проходит через определенную трассу
в день, чтобы в дальнейшем распреде-
лить полосы движения.
Для этой цели нам надо научить наш компьютер
распознавать виды транспорта. Если
бы мы решали эту задачу с помощью МО, мы бы
написали алгоритм, в котором указывали
бы характеристики машин, автобусов, грузовиков и
велосипедов, например, если количество
колес 2, то мотоцикл, если длина движущегося
средства более 5 метров, то грузовик либо
автобус, если много окон, то автобус, и т.д. Но как
понимаете, здесь много подводных камней.
Например, автобус может быть затонированным и
будет трудно понять, где там окна, либо
грузовик может выглядеть как автобус или наоборот,
да и крупные машины пикапы выглядят
как некоторые небольшие грузовики.

ражений с разными видами транспорта в наш
компьютер и просто указать ему, на каких изоб-
ражениях изображен мотоцикл, автомобиль, грузовик
или автобус. Компьютер сам начнет под-
бирать характеристики, по которым можно определить,
что за вид транспорта изображен и как
их можно отличить друг от друга. После этого мы
загрузим еще некоторое количество изоб-
ражений и протестируем насколько хорошо компьютер
справляется с задачей. Если он будет
ошибаться, мы укажем ему, что вот здесь ты ошибся,
здесь не грузовик, а автобус. Компьютер,
в свою очередь, вернется назад к своим алгоритмам
(это называется backpropagation) и внесет
туда какие-то изменения, и мы начнем заново по кругу
до тех пор, пока компьютер не начнет
угадывать, что изображено на картинке с очень
большой долей вероятности. Это и называется
глубокое обучение на основе нейронных сетей. Как вы
понимаете, это может занимать доста-
точно долгое время, может быть несколько недель, в
зависимости от сложности поставленной
задачи, также требует наличия большого количества
данных, желательно, чтобы было от мил-
лиона изображений и выше, и все эти изображения
должны либо быть промаркированы, либо

Давайте еще раз сравним МО и ГО по разным параметрам.
Если суммировать:
ГО является подобластью МО, и они оба подпадают под более широкое определение ИИ.

взвешенные решения на основе обученного.
ГО делает то же самое, так как оно тоже является разновидностью МО, но специфика ГО
в том, что при нем алгоритмы структурируются в несколько слоев, чтобы создать искусствен-
ную нейронную сеть, которая может тоже обучаться и принимать умные решения.
МО может использоваться при небольших наборах данных. И на маленьких объемах дан-
ных, МО и ГО имеют примерно одинаковую эффективность, но при возрастании объемов дан-
ных, ГО намного выигрывает по эффективности.
В МО мы сами задаем характеристики, на которые будут опираться наши алгоритмы.
В примере с определением цены квартиры, мы сами указываем параметры, от которых будет
зависеть цена, например, метраж, расстояние от метро, возраст дома, район и т. д. А в ГО, ком-
пьютер или можно сказать нейронная сеть сама методом проб и ошибок выводит определенные
параметры и их вес, от которых зависят наши выходные данные.
По времени обучения алгоритмов, ГО как правило занимает больше времени чем МО.

Расшифровка или интерпретация алгоритмов МО легче, потому что мы видим какой параметр играет важную роль для определения выходных данных. Например, в вопросе определения цены квартиры, мы можем увидеть, что вес метража в цене составляет, скажем, 60%.
В ГО же, расшифровать что именно привело к такому результату порой бывает очень сложно, потому что там несколько слоев нейронных сетей и много параметров, которые компьютер выводит сам и которые он может посчитать важными.
Поэтому, использование ГО или МО будет также зависеть от целей ваших задач.
Например, если вам надо понимать, почему компьютер принял то или иное решение, какой фактор сыграл важную роль, то вам надо будет выбрать использование МО вместо ГО.
Вследствие того, что ГО требует большего объема данных, а также более мощных вычислительных способностей компьютера, и занимает больше времени для обучения, оно также является более дорогим по сравнению с МО.

Сравнение

	Машинное обучение	Глубокое обучение
Количество данных	небольшое	очень большое
Признаки	необходимо предоставить / создать самому	обучается самостоятельно без указания признаков
Время обучения	коротко	долго
Точность	хорошая	очень высокая
Расшифровка	легко	очень сложно
Стоимость	недорого	дорого

Таким образом, если суммировать всю данную главу, то везде, где применяется распознавание речи или изображений, робототехника, устный или письменный перевод, чат-боты, беспилотное вождение транспортных средств, предсказание каких-то параметров на основе имеющихся данных, во всех этих примерах присутствуют элементы ИИ, потому что ИИ – это

очень широкое понятие, которое охватывает все эти направления, когда компьютер имитирует мышление и поведение человека.
Случаи, когда мы вместо того, чтобы давать компьютеру написанные инструкции и правила для решения вопроса, даем ему набор данных и он сам учится на них и находит необходимые алгоритмы и закономерности самостоятельно, такие случаи называются Машинным обучением. И одним из вариантов нахождения компьютером таких закономерностей является глубокое обучение, в котором используется несколько слоев нейронных сетей, что делает такие вычисления с одной стороны, более эффективными, с другой стороны, более трудными для расшифровки.

Примеры использования ИИ, МО и ГО

Их не надо контролировать каждую секунду, они сами принимают решения как объезжать препятствия и как собрать грунт в том или ином труднодоступном месте.

ИИ применяется и в беспилотных автомобилях. С помощью множества сенсоров, такие автомобили анализируют находящуюся вокруг них обстановку, определяют другие движущиеся машины, пешеходов, знаки дорожного движения, разметку, выбирают кратчайший путь и т.д.

Наше взаимодействие с голосовыми помощниками. Когда мы просим Алексу, Сири, или Алису от Яндекса сделать или найти что-то, они конвертируют наш голос в команды, обрабатывают их и выдают то, что нам необходимо.

Кроме голосовых помощников, очень развиты сейчас чат-боты, когда вы можете переписываться с компьютером, и он будет отвечать на ваши запросы. А в последнее время участились и звонки роботов на наши мобильные телефоны. Они могут предлагать какие-то рекламные акции или даже расспрашивать у вас информацию, например, когда вы планируете погасить кредитную задолженность. Такие роботы уже

то, то Google понимает, что это менее
нужная информация, и в следующий раз когда другой
человек зайдет на Google и спросит его
об этом же, то Google будет знать, что лучше выдать в
первой строчке на первой странице.
Решение о выдаче кредита банком. Компьютер
анализирует большое количество пара-
метров потенциального заемщика и потом
распределяет его в категорию хороший или плохой
заемщик, либо дает ему конкретный кредитный
скоринг. Все это происходит на основе кре-
дитной истории предыдущих заемщиков и как они
схожи с потенциальным новым заемщиком.
Выборка постоянно дополняется историей каждого
нового заемщика, расплатился ли он с кре-
дитом и выплатил ли его вовремя, она обновляется, и
также обновляется и алгоритм, находятся
новые закономерности, которые позволяют принимать
правильные решения о выдаче кредита
новому заемщику.
Выбор места для ритейла. В ритейле одним из самых
главных факторов, которые влияют
на прибыльность бизнеса, является местоположение.
У сети кофеен Старбакс имеется около 30
000 локаций по всему миру. Вы накопили большой

Глубокое обучение

Очень часто ГО используется для распознавания объектов на изображениях. Кроме того, с помощью ГО черно-белые изображения или фильмы можно сделать цветными. До этого компьютер уже обработал большое количество данных и информации в интернете либо в базе данной, которую ему предоставили для этого, и он уже знает различные оттенки серого и может легко понять в какой цвет необходимо преобразить тот или иной пиксель изображения.

Машинный перевод. Возможно, кто-то из вас использовал Google Translate, и вы могли заметить насколько хорошо он переводит в последнее время. Практически ничего не надо исправлять. Но если вспомнить примерно 5 или 7 лет назад, то качество перевода было далеко от идеального. А все потому, что сейчас вместо множества правил как надо переводить, используются нейронные сети, через которые уже прошли миллионы переводов художественной, технической и другой литературы, и эти алгоритмы ГО все продолжают улучшаться.

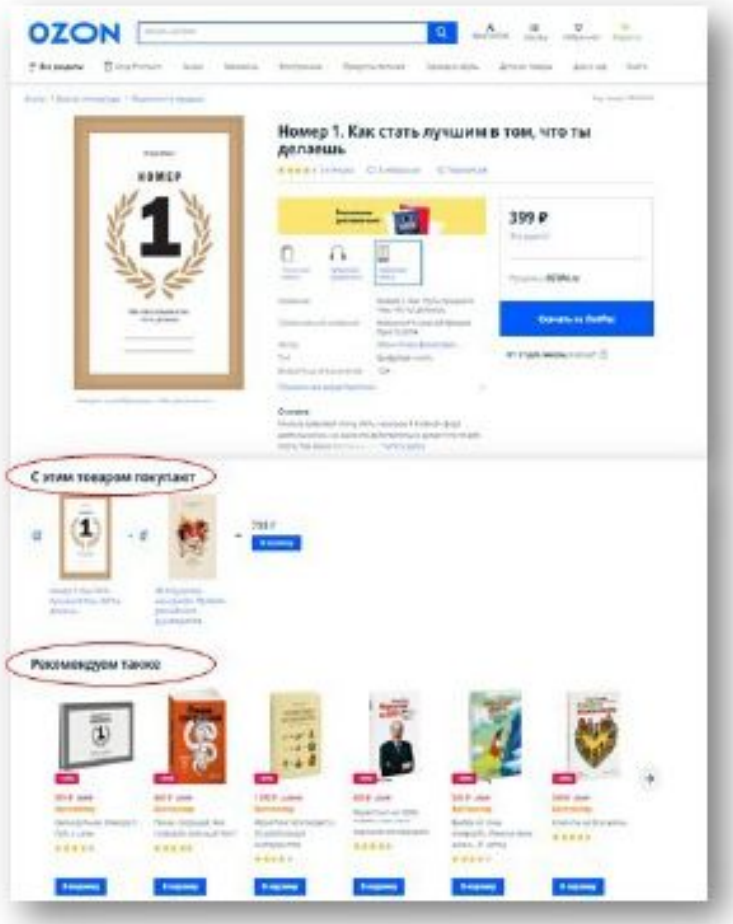
Примеры Глубокого обучения

Машинный перевод

Еще одним популярным применением ГО являются так называемые рекомендательные системы: когда при покупке одного товара нам предлагают другой. Наверное, вы видели, когда на сайте появляется фраза: «с этим товаром часто покупают». Или при просмотре фильма, или книги на сайте агрегаторе, вам начинают предлагать фильмы и книги похожей категории или те фильмы, которые смотрели пользователи, похожие на вас по различным параметрам. Все это алгоритмы ИИ, подкрепленные НС.

Примеры Глубокого обучения

Персонализированные рекомендации



И в конце, на что еще хотелось бы обратить внимание. Как уже было сказано, и ГО и МО являются только частью более общей области под названием ИИ. Так вот, в сложных проектах, как правило, присутствует несколько видов алгоритмов ИИ, и глубокое обучение и машинное обучение, и другие виды. Например, во время движения беспилотного автомобиля участвует более 100 различных алгоритмов, которые ответственны за распознавание объектов, управление движением, навигацию, безопасность, и т.д. Как вы заметили по приведенным примерам, ИИ уже используется во многих областях в нашей повседневной жизни. Считается, что в ближайшие пару десятилетий ИИ будет использоваться большинством компаний и охватывать большую часть нашей жизнедеятельности.

В 40-х годах XX в. с появлением ЭВМ искусственный интеллект обрел второе рождение. Произошло выделение искусственного интеллекта в самостоятельное научное направление. Сам термин «искусственный интеллект» (artificial intelligence) был предложен в 1956 году на семинаре с аналогичным названием в Станфордском университете (США).

Искусственный интеллект

- нейрокибернетика
- кибернетика «черного ящика»

Несмотря на наличие множества подходов как к пониманию задач ИИ, так и созданию интеллектуальных информационных систем можно выделить два основных подхода к разработке ИИ:
восходящий, биологический — изучение нейронных сетей и эволюционных вычислений, моделирующих интеллектуальное поведение на основе биологических элементов, а также создание соответствующих вычислительных систем, таких как нейрокомпьютер или биокомпьютер (нейрокибернетика);
нисходящий, семиотический — создание экспертных систем, баз знаний и систем логического вывода, имитирующих высокоуровневые психические процессы: мышление, рассуждение, речь, эмоции, творчество и т. д. (кибернетика «черного ящика»).

Пионером искусственного интеллекта по праву можно считать коллежского советника С.Н. Корсакова, ставившего задачу усиления возможностей разума посредством разработки научных методов и устройств, перекликающуюся с современной концепцией искусственного интеллекта, как усилителя естественного.

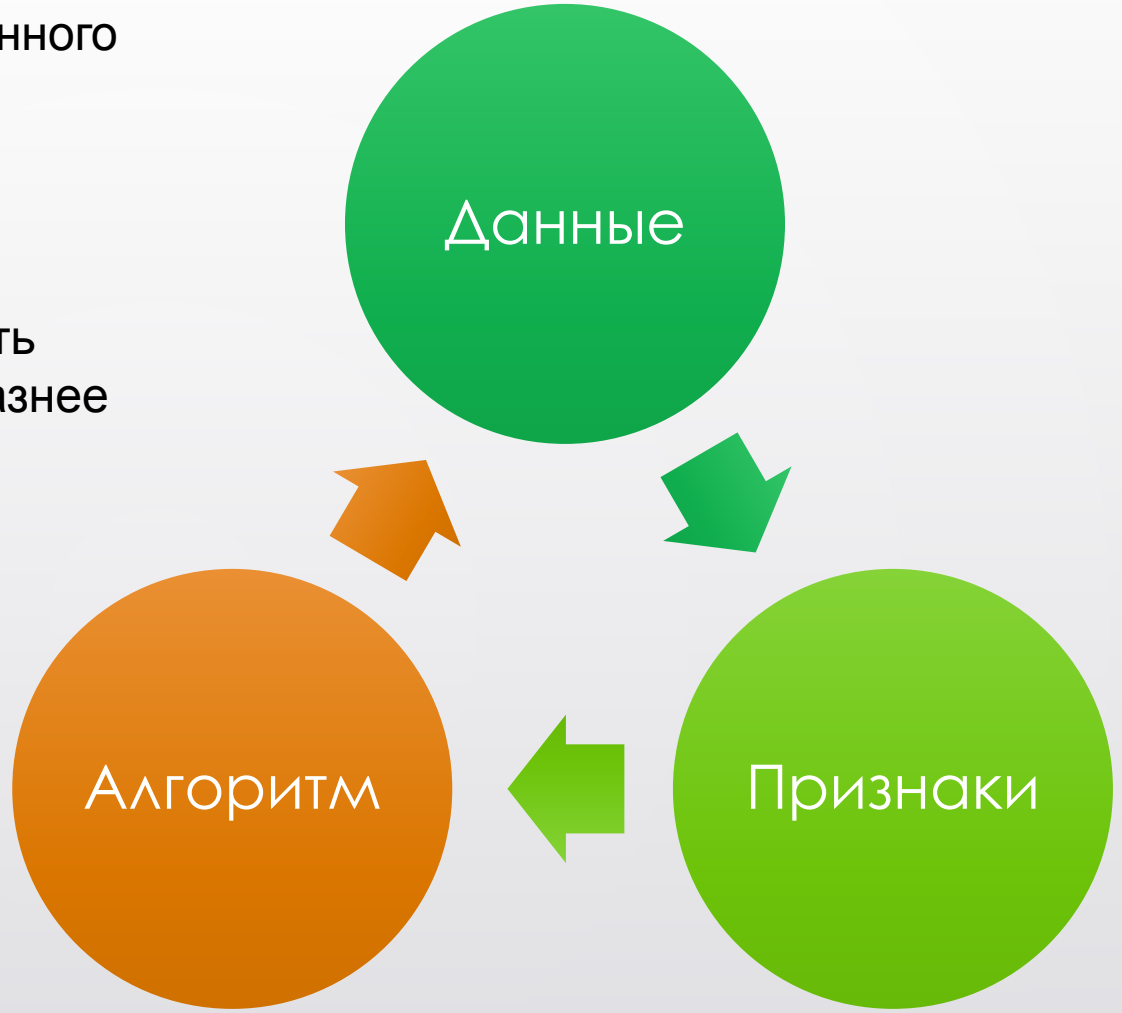
Работы в области искусственного интеллекта в России начались в 1960-х годах, возглавленных Вениамином Пушкиным и Д. А. Поспеловым.

До 1970-х годов в СССР все исследования ИИ велись в рамках кибернетики. Только в конце 1970-х в СССР начинают говорить о научном направлении «искусственный интеллект» как разделе информатики.

В конце 1970-х создается толковый словарь по искусственному интеллекту, трехтомный справочник по искусственному интеллекту и энциклопедический словарь по информатике, в котором разделы «Кибернетика» и «Искусственный интеллект» входят наряду с другими разделами в состав информатики.

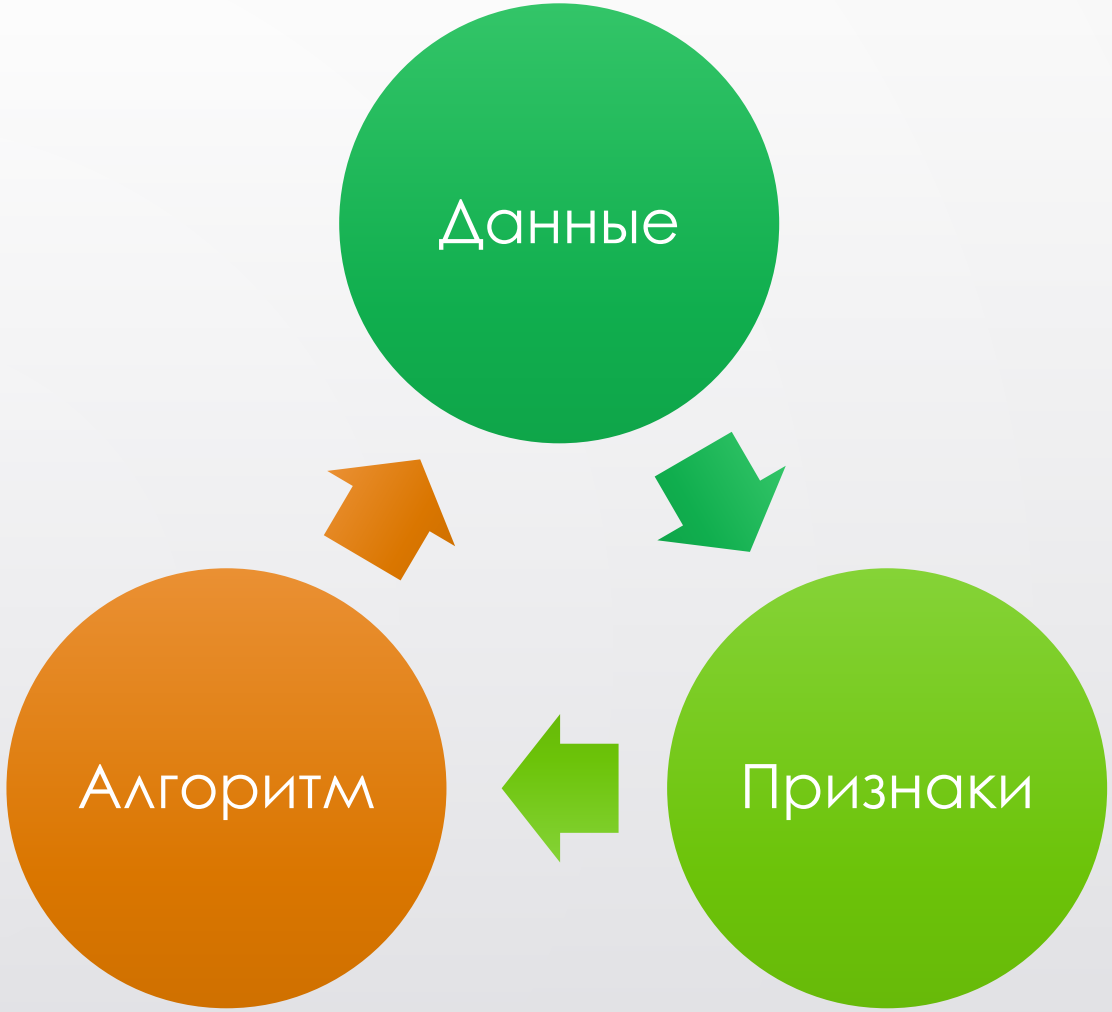
Машинное обучение — это раздел искусственного интеллекта.

Цель машинного обучения — это предсказать результат по входным данным. Чем разнообразнее входные данные, тем проще машине найти закономерности и тем точнее результат.

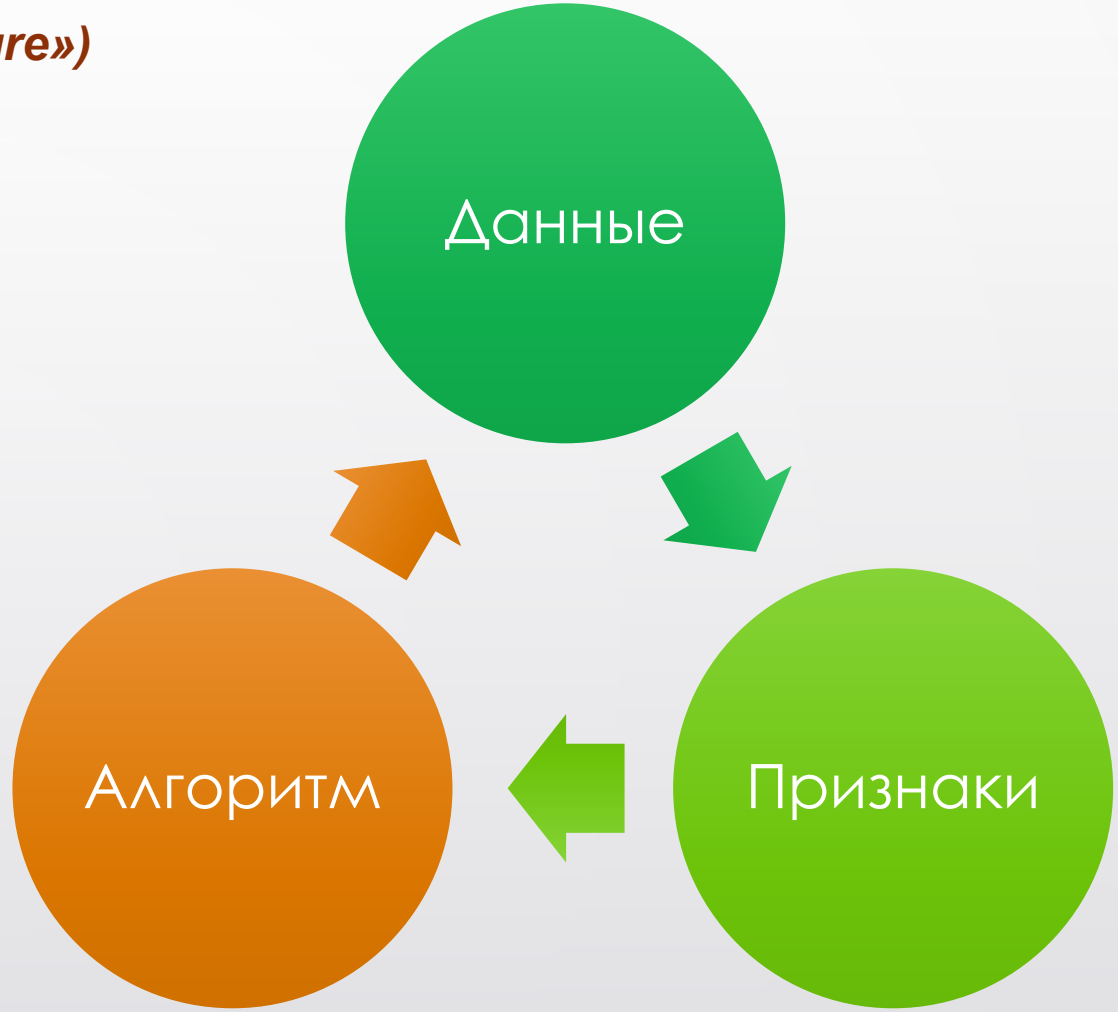


Данные

Цель	Исходные данные
Определить спам	Примеры спам-писем
Предсказать курс акций	История цен
Интересы пользователя	Посты, лайки



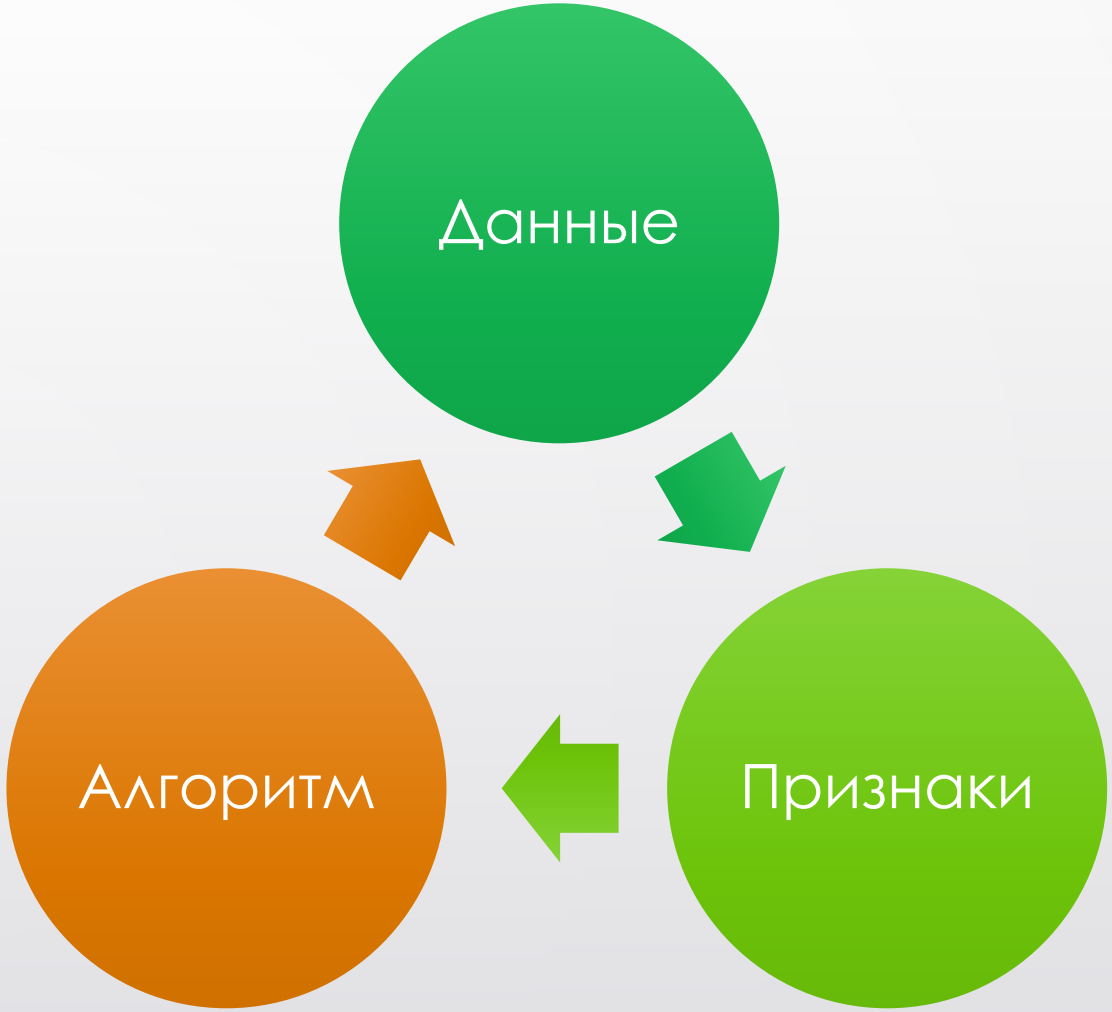
Признаки (фичи от английского слова «feature»)



Машина должна знать, на что ей конкретно смотреть.

Алгоритм

Одну задачу можно решить разными методами.
От выбора метода зависит точность, скорость работы и размер готовой модели.
Однако: если данные плохие, то даже самый лучший алгоритм не поможет.



Машина может:

- предсказывать,
- запоминать,
- воспроизводить,
- выбирать лучшее.

Машина не умеет:

- создавать новое;
- выйти за рамки поставленной задачи

Методы машинного обучения:

- классическое обучение;
- обучение с подкреплением;
- ансамбли;
- нейронные сети и глубокое обучение

Методы машинного обучения: классическое обучение

используется для задач,
когда есть простые данные и понятные признаки

Первые алгоритмы - из чистой статистики ещё в пятидесятых годах прошлого века.
Они решали формальные задачи — искали закономерности в цифрах, оценивали близость точек в пространствах и вычисляли направления.
Сегодня на классических алгоритмах держится добрая половина Интернета. Когда вы встречаете блок «Рекомендованные статьи» на сайте, или банк блокирует все ваши деньги на карточке после первой же покупки кофе за границей — это почти всегда дело рук одного из этих алгоритмов.

Методы машинного обучения: классическое обучение

Классификация
Пример метода
Дерево решений

Вы берёте кредит в банке. Как банку удостовериться, вернёте вы его или нет? Точно никак, но у банка есть тысячи профилей других людей, которые уже брали кредит до вас. Там указан их возраст, образование, должность, уровень зарплаты и главное — кто из них вернул кредит, а с кем возникли проблемы.

Для этой задачи придумали **деревья решений.** Машина автоматически разделяет все данные по вопросам, ответы на которые «да» или «нет». Вопросы могут быть не совсем адекватными с точки зрения человека, например «зарплата заёмщика больше, чем двадцать пять тысяч девятьсот тридцать четыре рубля?», но машина придумывает их так, чтобы на каждом шаге разбиение было самым точным. Так получается дерево вопросов. Чем выше уровень, тем более общий вопрос.
Применение деревьев – диагностика, медицина, финансы.

Деревья используются при ранжировании результатов при поиске в Яндексе.

Методы машинного обучения: классическое обучение

Классификация
Пример метода
Метод опорных векторов

Им классифицировали уже всё — виды растений, лица на фотографиях, документы по тематикам. Идея метода опорных векторов по своей сути проста — он ищет, как провести две прямые между категориями, чтобы между ними образовался наибольший зазор.

Методы машинного обучения: классическое обучение

Классификация

У классификации есть полезная обратная сторона — **поиск аномалий.** Когда какой-то признак объекта не вписывается в установленные классы, можно ярко подсветить его на экране. Сейчас так делают в медицине: компьютер подсвечивает врачу все подозрительные области МРТ или выделяет отклонения в анализах. На биржах таким же образом определяют нестандартных игроков, которые скорее всего являются инсайдерами. Научив компьютер «как правильно», мы автоматически получаем и обратный классификатор — как неправильно. И сегодня для классификации всё чаще используют искусственные нейронные сети, ведь по сути их для этого и изобрели.

Методы машинного обучения: классическое обучение

Регрессия

Регрессия — та же классификация, только вместо категории мы предсказываем число. Например, стоимость автомобиля по его пробегу, количество пробок по времени суток, объём спроса на товар от роста компании и так далее. На регрессию идеально ложатся любые задачи, где есть зависимость от времени.

Регрессию очень любят финансисты и аналитики, она встроена даже в Excel. Машина пытается нарисовать линию, которая в среднем отражает зависимость. Делает она это математически точно — считая среднее расстояние до каждой точки и пытаясь всем угодить.

Когда регрессия рисует прямую линию, её называют линейной, когда кривую — полиномиальной.

Методы машинного обучения: классическое обучение

Классификация
Пример метода
Кластеризация, или кластерный анализ

Это классификация, но без заранее известных классов.
Машина сама ищет похожие объекты и объединяет их в кластеры. Количество кластеров можно задать заранее или доверить это машине. Похожесть объектов машина определяет по тем признакам, которые мы ей разметили — у кого много схожих характеристик, те попадают в один класс.

Отличный пример кластеризации — это маркеры на картах в вебе.
Более сложные примеры кластеризации можно вспомнить в приложениях iPhoto или Google Photos, которые находят лица людей на фотографиях и группируют их в альбомы. Приложение не знает, как зовут ваших друзей, но может отличить их по характерным чертам лица. Это типичная кластеризация. Правда для начала им приходится найти эти самые «характерные черты», а это уже только при обучении с учителем.

Методы машинного обучения: классическое обучение

Классификация
Пример метода
Кластеризация, или кластерный анализ

Сжатие изображений — ещё одна популярная проблема. Сохраняя картинку в формате PNG, вы можете установить палитру, скажем, в тридцать два цвета. Тогда кластеризация найдёт все «примерно красные» пиксели изображения, высчитает из них «средний красный по больнице» и заменит все красные на него. Меньше цветов — меньше файл. Проблема только, как быть с цветами типа cyan — вот он ближе к зелёному или синему?

Методы машинного обучения: классическое обучение

Классификация
Пример метода

Кластеризация, или кластерный анализ
метод К-средних

Проблема только, как быть с цветами типа cyan — вот он ближе к зелёному или синему?

В этом случае используют метод К-средних. Мы случайным образом бросаем на палитру цветов наши тридцать две точки, обзывая их «центроидами». Все остальные точки относим к ближайшему центроиду от них — получаются как бы созвездия из самых близких цветов. Затем двигаем центроид в центр своего созвездия и повторяем пока центроиды не перестанут двигаться. Кластеры обнаружены, стабильны и их ровно тридцать два, как и надо было.

Искать центроиды удобно и просто, но в реальных задачах кластеры могут быть совсем не круглой формы.

Методы машинного обучения: классическое обучение

Классификация
Пример метода
Кластеризация, или кластерный анализ
метод DBSCAN

Метод DBSCAN. Он сам находит скопления точек и строит вокруг кластеры. Его легко понять, если представить, что точки — это люди на площади.
Находим трёх любых близко стоящих человека и говорим им взяться за руки. Затем они начинают брать за руку тех, до кого могут дотянуться. Так по цепочке, пока никто больше не сможет взять кого-то за руку — это и будет первый кластер. Повторяем, пока не поделим всех. Те, кому вообще некого брать за руку — это выбросы или аномалии.

Методы машинного обучения: обучение с подкреплением

Обучение с подкреплением — это когда мы бросаем робота в лабиринт, и он сам там ищет из него выход. Сегодня этот метод используют для беспилотных автомобилей, роботов-пылесосов, игр, автоматической торговли, управления ресурсами предприятий.

Обучение с подкреплением используют там, где задачей стоит не анализ данных, а выживаниев реальной среде.

Знания об окружающем мире такому роботу могут быть полезны, но чисто для справки. Неважно, сколько данных он соберёт, у него всё равно не получится предусмотреть все ситуации. Потому его цель — минимизировать ошибки, а не рассчитать все ходы.

Методы машинного обучения: обучение с подкреплением

Умные модели роботов-пылесосов и беспилотные автомобили обучаются именно так: часто на основе карт настоящих городов для них создают виртуальный город, населяют его случайными пешеходами и отправляют учиться никого там не убивать. Когда робот начинает хорошо себя чувствовать в искусственном мире, его выпускают тестировать на реальные улицы.

Запоминать сам город машине не нужно — такой подход называется **Model-Free**. Конечно, есть и классический **Model-Based** подход, но в нём нашей машине пришлось бы запоминать модель всей планеты, всех возможных ситуаций на всех перекрёстках мира. Такое просто не работает.

В обучении с подкреплением машина не запоминает каждое движение, а пытается обобщить ситуации, чтобы выходить из них с максимальной выгодой.

Методы машинного обучения: обучение с подкреплением

Марковский процесс принятия решений — это способ последовательного решения задачи для полностью наблюдаемой среды, в которой осуществляется вознаграждение. При этом эволюция состояния системы в среде описывается марковским процессом, что является случайным переходом из состояния в состояние, когда история таких переходов во внимание не принимается.

Сегодня это широко используемая модель в робототехнике, машинном обучении с подкреплением и Искусственном Интеллекте.

Как это происходит? Например, робот-пылесос, если он устроен на механизмах обучения с подкреплением, в каждый момент времени рассматривает все альтернативы своих возможных действий. Для каждой альтернативы он не учитывает предыдущие свои состояния, и это как раз марковский процесс. Для будущих состояний он узнаёт реакцию среды, то есть среда как раз полностью наблюдаема. Ну и запускается уже марковский процесс принятия решений, при помощи которого робот-пылесос выбирает, что делать дальше.

Пример метода

Стекинг

Мы обучаем несколько разных алгоритмов и передаём их результаты на вход последнему, который принимает итоговое решение.

Ключевое слово здесь — «разных» алгоритмов, ведь один и тот же алгоритм, обученный на одних и тех же данных, не имеет смысла. В качестве решающего алгоритма чаще всего берут регрессию.

Методы машинного обучения: ансамбли

Пример метода

Бэггинг

Бэггинг — это когда мы обучаем один алгоритм много раз на случайных выборках из исходных данных, а в самом конце усредняем ответы. Никакой магии. Данные в случайных выборках могут повторяться. То есть из набора один-два-три мы можем делать выборки два-два-три, один-два-два, три-один-два и так, пока не надоест. На них мы обучаем один и тот же алгоритм несколько раз, а в конце вычисляем ответ простым голосованием.

Самый популярный пример бэггинга — алгоритм Random Forest, то есть бэггинг на деревьях. Когда вы открываете камеру на телефоне и видите, как она очертила лица людей в кадре жёлтыми прямоугольниками — скорее всего это их работа. Нейросеть будет слишком медлительна в реальном времени, а бэггинг идеален, ведь он может считать свои деревья параллельно на всех шейдерах видеокарты.

Методы машинного обучения: ансамбли

Пример метода

Бустинг

При бустинге мы обучаем алгоритмы последовательно, и каждый следующий уделяет особое внимание тем случаям, на которых ошибся предыдущий.
Как и в бэггинге, мы делаем выборки из исходных данных, но теперь не совсем случайно. В каждую новую выборку мы берём часть тех данных, на которых предыдущий алгоритм отработал неправильно. То есть как бы доучиваем новый алгоритм на ошибках предыдущего.

Методы машинного обучения: искусственные нейронные сети

Сегодня нейронные сети используют для определения объектов на фото и видео, распознавания и синтеза речи, обработки изображений, переноса стиля, машинного перевода, да и вообще вместо всех ранее изученных алгоритмов, так как нейронная сеть является универсальным аппроксиматором.

Любая нейросеть — это набор нейронов и связей между ними. Нейрон лучше всего представлять просто как функцию с кучей входов и одним выходом. Задача нейрона — взять числа со своих входов, выполнить над ними функцию и отдать результат на выход. Простой пример полезного нейрона: просуммировать все цифры со входов, и если их сумма больше какого-то числа N — выдать на выход единицу, иначе выдать ноль.