

Постановка задачи машинного обучения

Типы задач машинного обучения

Характеристики и отличие контролируемого обучения (с учителем) и неконтролируемого обучения (без учителя).

Типы данных

Постановка задачи машинного обучения

Пример задачи

- Сеть ресторанов
- Хотим открыть еще один
- Несколько вариантов размещения
- Какой из вариантов принесет максимальную прибыль?

* см. [kaggle.com](https://www.kaggle.com/tfi-restaurant-revenue-prediction), TFI Restaurant Revenue Prediction

Обозначения

- x — объект, sample — для чего хотим делать предсказания
 - Конкретное расположение ресторана
- X — пространство всех возможных объектов
 - Все возможные расположения ресторанов
- y — ответ, целевая переменная, target — что предсказываем
 - Прибыль в течение первого года работы
- Y — пространство ответов — все возможные значения ответа
 - Все вещественные числа

Обучающая выборка

- Мы ничего не понимаем в экономике
 - Зато имеем много объектов с известными ответами
 - $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
 - ℓ — размер выборки
-
- Нельзя вывести корректную формулу будущей прибыли из каких-либо знаний об устройстве мира.
 - Предсказываемая прибыль зависит от признаков, но при этом вид зависимости достаточно сложный, и подобрать его вручную может быть слишком трудно.
 - Можно набрать достаточное количество примеров (т.е. объектов с известными ответами), по которым можно оценить зависимость целевой переменной от признаков.

Признаки

- Объекты — абстрактные сущности
- Компьютеры работают только с числами
- Признаки, факторы, features — числовые характеристики объектов
- d — количество признаков
- $x = (x^1, \dots, x^d)$ — признаковое описание

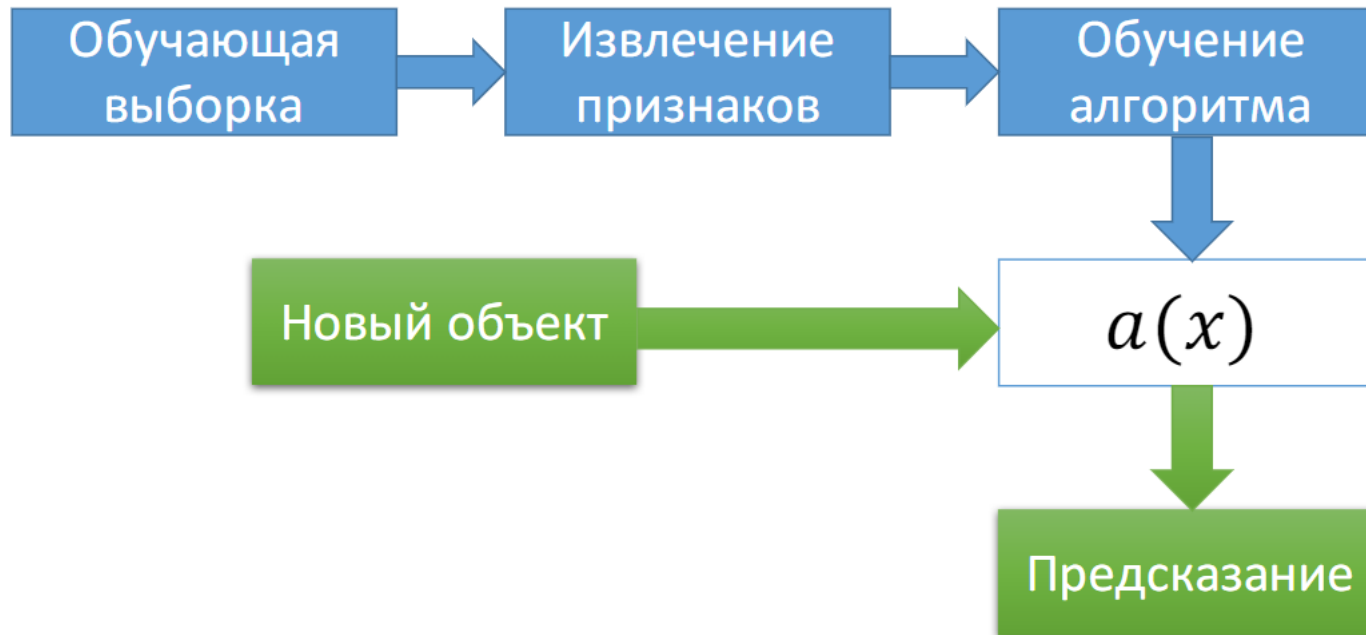


Вектор

Алгоритм

- $a(x)$ — алгоритм, модель — функция, предсказывающая ответ для любого объекта
- Отображает \mathbb{X} в \mathbb{Y}
- Линейная модель: $a(x) = w_1 x^1 + \dots + w_d x^d$

Машинное обучение



Пусть

$X = R^d$ - множество объектов (входов, признаков)

$Y = R$ - множество ответов

$f : X \rightarrow Y$ - *неизвестная зависимость*

Дано:

$\{x_1, \dots, x_d\} \subset X$ - *обучающая выборка (training sample)*

$y_i = y(x_i), y_i \in Y, i = 1, \dots, l$ - *известные ответы*

$X = X^L + X^T$, обучающая + тестовая выборка

Найти:

$a : X \rightarrow Y$ - решающую функцию (decision function), приближающую f на множестве X .

$X = X^L + X^T$

Задача с открытием сети ресторанов

$f_j: X \rightarrow D_j, j = 1, \dots, n$ — признаки объектов (features).

Типы признаков:

- $D_j = \{0, 1\}$ — бинарный признак f_j ;
- $|D_j| < \infty$ — номинальный признак f_j ;
- $|D_j| < \infty, D_j$ упорядочено — порядковый признак f_j ;
- $D_j = \mathbb{R}$ — количественный признак f_j .

ПРИЗНАКИ

› Описывают объекты

› D_j — множество значений j -го признака

БИНАРНЫЕ ПРИЗНАКИ

› $D_j = \{0, 1\}$

› Доход клиента выше среднего по городу?

› Цвет фрукта — зеленый?

ВЕЩЕСТВЕННЫЕ ПРИЗНАКИ

› $D_j = \mathbb{R}$

› Возраст

› Площадь квартиры

› Количество звонков в колл-центр

КАТЕГОРИАЛЬНЫЕ ПРИЗНАКИ

› D_j — неупорядоченное множество

› Цвет глаз

› Город

› Образование (может быть упорядоченным)

ПОРЯДКОВЫЕ ПРИЗНАКИ

› D_j — упорядоченное множество

› Роль в фильме (первого плана, второго плана, массовка)

› Тип населенного пункта

› Образование (может быть неупорядоченным)

МНОЖЕСТВОЗНАЧНЫЕ ПРИЗНАКИ

› (set-valued)

› D_j — множество всех подмножеств некоторого множества

› Какие фильмы посмотрел пользователь?

› Какие слова входят в текст?

Сбор данных и проверка их качества. Виды данных.



Особенности реальных данных

В реальных приложениях данные бывают ...

- разнородные (признаки измерены в разных шкалах)
- неполные (признаки измерены не все, имеются пропуски)
- неточные (признаки измерены с погрешностями)
- противоречивые (объекты одинаковые, ответы разные)
- избыточные (сверхбольшие, не помещаются в память)
- недостаточные (объектов меньше, чем признаков)
- неструктурированные (нет признаков описаний)
- «грязные» (ошибочные, грубо не соответствующие истине)

*со всем этим
можно
работать*



*но только не
с грязными
данными!*



Предварительная обработка и очистка данных — это важные задачи, которые необходимо выполнить, прежде чем набор данных можно будет использовать для обучения модели. Необработанные данные зачастую искажены и ненадежны, и в них могут быть пропущены значения. Использование таких данных при моделировании может приводить к неверным результатам. Эти задачи являются частью процесса обработки и анализа данных группы и обычно подразумевают первоначальное изучение набора данных, используемого для определения и планирования необходимой предварительной обработки.

Типичные проблемы с качеством данных:

- **Неполнота:** данные не содержат атрибутов, или в них пропущены значения.
- **Шум:** данные содержат ошибочные записи или выбросы.
- **Несогласованность:** данные содержат конфликтующие между собой записи или расхождения.

Что нужно оценить, чтобы проверить качество данных:

- **Количество записей.**
- количество **атрибутов** (или **компонентов**);
- **Типы данных** атрибута (номинальные, порядковые или непрерывные).
- **Количество пропущенных значений.**
- **Правильно сформированные данные.**
 - Если данные имеют формат TSV или CSV, проверьте правильность разделения столбцов и строк соответствующими разделителями.
 - Если данные имеют формат HTML или XML, убедитесь, что формат данных соответствует надлежащим стандартам.
 - Для извлечения структурированной информации из частично структурированных или неструктурированных данных также может потребоваться синтаксический анализ.
- **Несогласованные записи данных.** Проверьте допустимость диапазона значений. Например, если данные содержат GPA-запись учащегося (среднее значение точки), проверьте, находится ли GPA в указанном диапазоне, скажем 0 ~ 4.

Работа с номинальными признаками

- Каждой категории сопоставляет некоторое целое число
- Димми-кодирование. Для кодируемого категориального признака создаются N новых признаков, где N — число категорий. Каждый i-й новый признак — бинарный характеристический признак i-й категории.

	city	class	degree	income
0	Moscow	A	1	10.2
1	London	B	1	11.6
2	London	A	2	8.8
3	Kiev	A	2	9.0
4	Moscow	B	3	6.6
5	Moscow	B	3	10.0
6	Kiev	A	1	9.0
7	Moscow	A	1	7.2



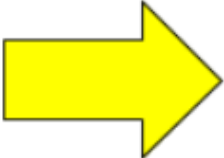
	city	class	degree	income	city_le
0	Moscow	A	1	10.2	2
1	London	B	1	11.6	1
2	London	A	2	8.8	1
3	Kiev	A	2	9.0	0
4	Moscow	B	3	6.6	2
5	Moscow	B	3	10.0	2
6	Kiev	A	1	9.0	0
7	Moscow	A	1	7.2	2



	city	class	degree	income	city=0	city=1	city=2
0	Moscow	A	1	10.2	0	0	1
1	London	B	1	11.6	0	1	0
2	London	A	2	8.8	0	1	0
3	Kiev	A	2	9.0	1	0	0
4	Moscow	B	3	6.6	0	0	1
5	Moscow	B	3	10.0	0	0	1
6	Kiev	A	1	9.0	1	0	0
7	Moscow	A	1	7.2	0	0	1

One-hot кодирование

Бинарное кодирование категориальных признаков:

Color		Red	Yellow	Green
Red		1	0	0
Red		1	0	0
Yellow		0	1	0
Green		0	0	1
Yellow				

Работа с пропущенными значениями

Как обрабатывать пропущенные значения

При работе с пропущенными значениями лучше сначала определить причину их появления в данных, что поможет решить проблему. Вот какие бывают методы обработки пропущенных значений:

- **Удаление:** удаление записей с пропущенными значениями.
- **Фиктивная подстановка** — замена пропущенных значений фиктивными, например подстановка значения *unknown* (неизвестно) вместо категориальных или значения 0 вместо чисел.
- **Подстановка среднего значения:** пропущенные числовые данные можно заменить средним значением.
- **Подстановка часто используемого элемента:** пропущенные категориальные значения можно заменить наиболее часто используемым элементом.
- Присовить пропущенным значениям некоторое другое значение, которое не встречалось прежде. Тем самым мы создадим отдельную категорию «пропущенные значения». Это не очень работает для линейных моделей и нейронных сетей.
- Постараться как-нибудь хитро реконструировать пропущенные значения.

Вектор $(f_1(x), \dots, f_n(x))$ — *признаковое описание* объекта x .

Матрица «объекты–признаки» (features data)

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}$$

Матрица объекты–признаки

Числовая матрица:

	Признак 1	Признак 2	...	Признак К
Объект 1				
Объект 2				
Объект 3				
...				
Объект N				

Задача кредитного скоринга

Объект — заявка на выдачу кредита.

Классы — bad или good.

Примеры признаков:

- бинарные: пол, наличие телефона, и т. д.
- номинальные: место проживания, профессия, работодатель, и т. д.
- порядковые: образование, должность, и т. д.
- количественные: возраст, зарплата, стаж работы, доход семьи, сумма кредита, и т. д.

Особенности задачи:

- нужно оценивать вероятность дефолта $P(\text{bad})$.

Объект — абонент в определённый момент времени.

Классы — уйдёт или не уйдёт в следующем месяце.

Примеры признаков:

- **бинарные:** корпоративный клиент, включение услуг, и т. д.
- **номинальные:** тарифный план, регион проживания, и т. д.
- **количественные:** длительность разговоров (входящих, исходящих, СМС, межгород, и т. д.), частота оплаты, и т. д.

Особенности задачи:

- нужно строить признаки по потоку действий абонентов;
- нужно оценивать вероятность ухода;
- сверхбольшие выборки.

Задача регрессии: прогноз стоимости недвижимости

Объект — квартира в Москве.

Примеры признаков:

- **бинарные:** наличие балкона, лифта, мусоропровода, охраны, гаража, чердака, и т. д.
- **номинальные:** район города, тип дома (кирпичный/панельный/блочный/монолит), и т. д.
- **количественные:** число комнат, жилая площадь, расстояние до центра, до метро, возраст дома, и т. д.

Особенности задачи:

- выборка неоднородна, стоимость меняется со временем;
- разнотипные признаки;
- для линейной модели нужны преобразования признаков.

Вернемся к ресторанам

Обучение с учителем используется всякий раз, когда мы хотим предсказать определенный результат (ответ) по данному объекту, и у нас есть пары объект-ответ. Мы строим модель машинного обучения на основе этих пар объект-ответ, которые составляют наш обучающий набор данных. Наша цель состоит в том, чтобы получить точные прогнозы для новых, никогда ранее не встречавшихся данных. Машинное обучение с учителем часто требует вмешательства человека, чтобы получить обучающий набор данных, но потом оно автоматизирует и часто ускоряет решение трудоемких или неосуществимых задач.

Обучение с учителем

Обучение с учителем (Supervised learning) — один из разделов машинного обучения, посвященный решению следующей задачи.

Имеется множество *объектов* (ситуаций) и множество возможных *ответов* (откликов, реакций). Существует некоторая зависимость между ответами и объектами, но она неизвестна. Известна только конечная совокупность *прецедентов* — пар «объект, ответ», называемая обучающей выборкой. На основе этих данных требуется восстановить зависимость, то есть построить алгоритм, способный для любого объекта выдать достаточно точный ответ. Для измерения точности ответов определённым образом вводится *функционал качества*.

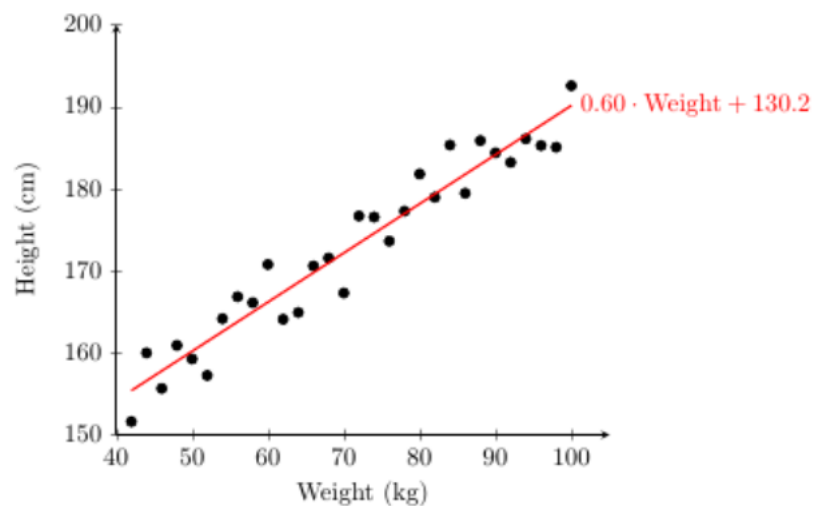
Под *учителем* понимается либо сама обучающая выборка, либо тот, кто указал на заданных объектах правильные ответы

Есть две основные задачи машинного обучения с учителем:

- классификация (classification)
- регрессия (regression)

Регрессия

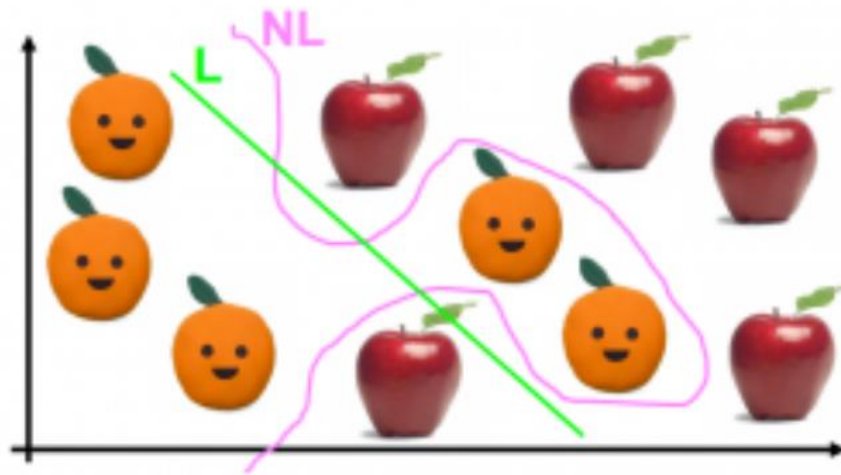
- Вещественные ответы: $Y = \mathbb{R}$
- (вещественные числа — числа с любой дробной частью)
- Пример: предсказание роста по весу



1. $\mathbb{Y} = \{0, 1\}$ — бинарная классификация. Например, мы можем предсказывать, кликнет ли пользователь по рекламному объявлению, вернет ли клиент кредит в установленный срок, сдаст ли студент сессию, случится ли определенное заболевание с пациентом (на основе, скажем, его генома).
2. $\mathbb{Y} = \{1, \dots, K\}$ — многоклассовая (multi-class) классификация. Примером может служить определение предметной области для научной статьи (математика, биология, психология и т.д.).
3. $\mathbb{Y} = \{0, 1\}^K$ — многоклассовая классификация с пересекающимися классами (multi-label classification). Примером может служить задача автоматического проставления тегов для ресторанов (логично, что ресторан может одновременно иметь несколько тегов).

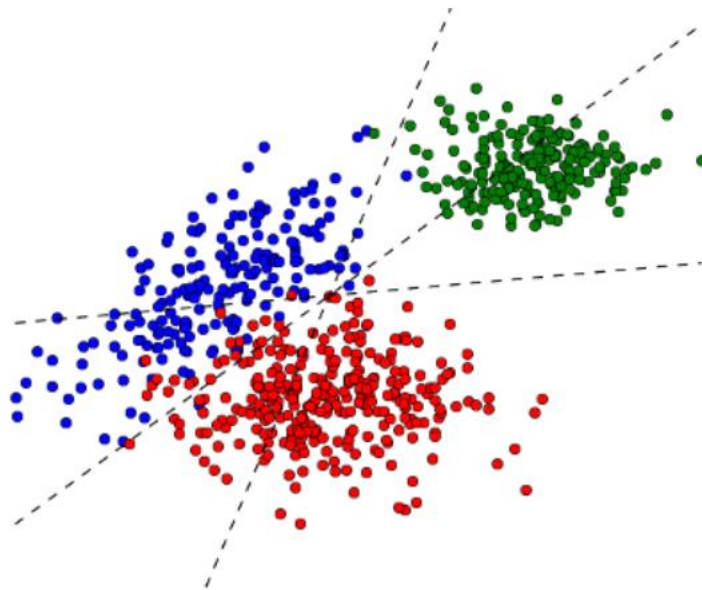
Классификация

- Конечное число ответов: $|\mathbb{Y}| < \infty$
- Бинарная классификация: $\mathbb{Y} = \{-1, +1\}$



Классификация

- Многоклассовая классификация: $\mathbb{Y} = \{1, 2, \dots, K\}$



Алгоритмы машинного обучения с учителем:

- [Ближайшие соседи \(k-Nearest Neighbors\)](#) - Подходит для небольших наборов данных, хорош в качестве базовой модели, прост в объяснении.
- [Линейные модели \(Linear Models\)](#) - Считается первым алгоритмом, который нужно попробовать, хорош для очень больших наборов данных, подходит для данных с очень высокой размерностью.
- [Наивный байесовский классификатор \(Naive Bayes Classifiers\)](#) - Подходит только для классификации. Работает даже быстрее, чем линейные модели, хорош для очень больших наборов данных и высокоразмерных данных. Часто менее точен, чем линейные модели.
- Деревья решений (Decision trees)- Очень быстрый метод, не нужно масштабировать данные, результаты можно визуализировать и легко объяснить.
- Случайные леса (Random forests) - Почти всегда работают лучше, чем одно дерево решений, очень устойчивый и мощный метод. Не нужно масштабировать данные. Плохо работает с данными очень высокой размерности и разреженными данными.
- Градиентный бустинг деревьев решений (Gradient Boosted Regression Trees (Gradient Boosting Machines)) - Как правило, немного более точен, чем случайный лес. В отличие от случайного леса медленнее обучается, но быстрее

предсказывает и требует меньше памяти. По сравнению со случайным лесом требует настройки большего числа параметров.

- Машины опорных векторов (Kernelized Support Vector Machines, SVM) - Мощный метод для работы с наборами данных среднего размера и признаками, измеренными в едином масштабе. Требует масштабирования данных, чувствителен к изменению параметров.
- Нейронные сети (Neural Networks (Deep Learning)) - Можно построить очень сложные модели, особенно для больших наборов данных. Чувствительны к масштабированию данных и выбору параметров. Большим моделям требуется много времени для обучения.

- Медицинская диагностика
Симптомы → заболевание
- Фильтрация спама
Письмо → спам/не спам
- Рекомендательные системы
Прошлые покупки → рекомендация
- Компьютерное зрение
Изображение → что изображено
- Распознавание текста
Рукописный текст → текст в машинном коде
- Компьютерная лингвистика
Предложение на русском языке → Дерево синтаксического разбора
- Машинный перевод
Текст на русском языке → перевод на английский
- Распознавание речи
Аудиозапись речи → текст

•

Распознавание изображений

Например, распознавание рукописного символа (цифры) по его изображению.

Данные optdigit <http://www.ics.uci.edu/~mllearn/MLRepository.html> содержат 1934 размеченных черно-белых изображений цифр 32×32 .

1 — пиксел черный, 0 — пиксел белый.

Признаковое описание — бинарный вектор x длины $32^2 = 1024$.

$$\mathcal{X} = \{0, 1\}^{1024}.$$

$$\mathcal{Y} = \{0, 1, 2, \dots, 9\}$$

•

Некоторые объекты из обучающей выборки

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
6	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

Обучение без учителя (самообучение, спонтанное обучение, [англ. *Unsupervised learning*](#)) — один из способов [машинного обучения](#), при котором испытуемая система спонтанно обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора. С точки зрения [кибернетики](#), это является одним из видов [кибернетического эксперимента](#). Как правило, это пригодно только для задач, в которых известны описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами.

Обучение без учителя часто противопоставляется [обучению с учителем](#), когда для каждого обучающего объекта принудительно задаётся «правильный ответ», и требуется найти зависимость между стимулами и реакциями системы.

Цели обучения без учителя

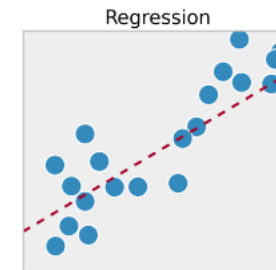
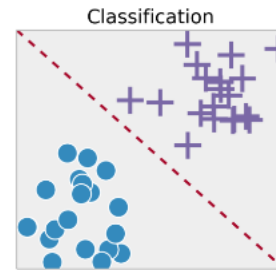
- в *Data Mining*:
выявлять структуру в данных для лучшего их понимания;
- в *Machine Learning*:
как предварительный этап при решении задачи обучения с учителем (например, сокращение размерности (РСА и др.) или решаем задачу кластеризации, а потом в каждом кластере — свою задачу классификации и т. п.).

- 1. Кластеризация — задача разделения объектов на группы, обладающие некоторыми свойствами. Примером может служить кластеризация документов из электронной библиотеки или кластеризация абонентов мобильного оператора.
- 2. Оценивание плотности — задача приближения распределения объектов. Примером может служить задача обнаружения аномалий, в которой на этапе обучения известны лишь примеры «правильного» поведения оборудования (или, скажем, игроков на бирже), а в дальнейшем требуется обнаруживать случаи некорректной работы (соответственно, незаконного поведения игроков). В таких задачах сначала оценивается распределение «правильных» объектов, а затем аномальными объявляются все объекты, которых в рамках этого распределения получают слишком низкую вероятность.
- 3. Визуализация — задача изображения многомерных объектов в двумерном или трехмерном пространстве таким образом, что сохранялось как можно больше зависимостей и отношений между ними.
- 4. Понижение размерности — задача генерации таких новых признаков, что их меньше, чем исходных, но при этом с их помощью задача решается не хуже (или с небольшими потерями качества, или лучше — зависит от постановки). К этой же категории относится задача построения латентных моделей, где требуется описать процесс генерации данных с помощью некоторого (как правило, небольшого) набора скрытых переменных. Примерами являются задачи тематического моделирования и построения рекомендаций, которым будет посвящена часть курса.

Типология задач машинного обучения

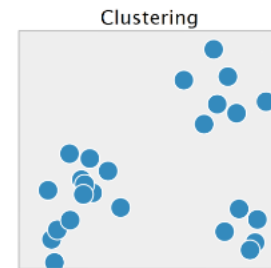
Обучение с учителем (supervised learning)

- классификация (classification)
- регрессия (regression)
- ранжирование (learning to rank)
- прогнозирование (forecasting)



Обучение без учителя (unsupervised learning)

- кластеризация (clustering)
- поиск ассоциативных правил (association rule learning)
- восстановление плотности (density estimation)
- одноклассовая классификация (anomaly detection)



Частичное обучение (semi-supervised learning)

- обучение с положительными примерами (PU-learning)

Что предсказываем?

Два типа обучения:

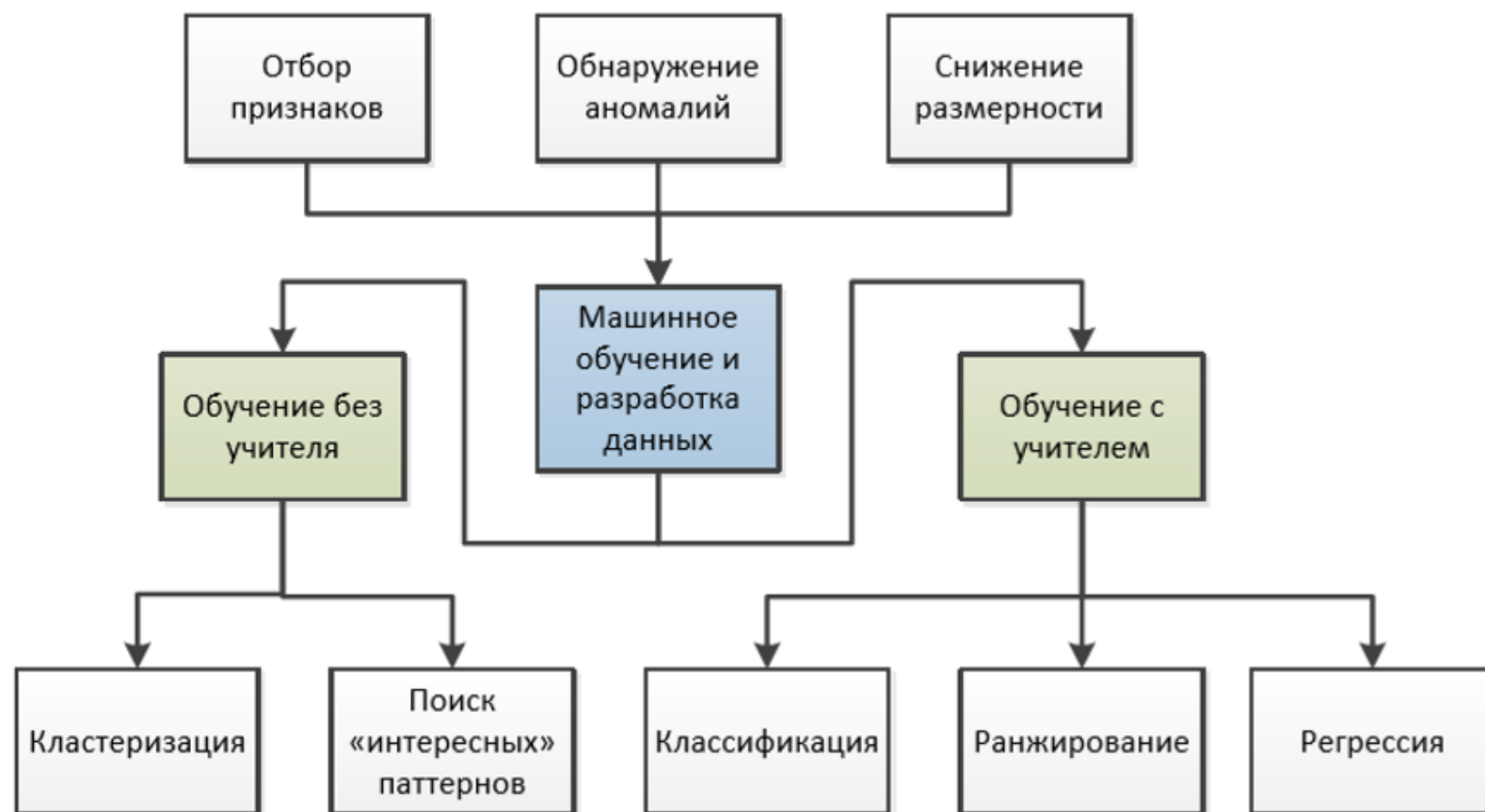
- Обучение с учителем (пытаемся понять, как зависят ответы, известные на объектах обучающей выборки, от входных данных):
 - Классификация (бинарная, multiclass, multilabel)
 - Регрессия
 - Прогнозирование временных рядов
 - Рекомендации
 - ...
- Обучение без учителя (как можем формализуем, что хотим найти в данных, и ищем).
 - Кластеризация
 - Понижение размерности
 - Визуализация
 - ...

Что предсказываем?

Два типа обучения:

- Обучение с учителем (пытаемся понять, как зависят ответы, известные на объектах обучающей выборки, от входных данных):
 - Классификация (бинарная, multiclass, multilabel)
 - Регрессия
 - Прогнозирование временных рядов
 - Рекомендации
 - ...
- Обучение без учителя (как можем формализуем, что хотим найти в данных, и ищем).
 - Кластеризация
 - Понижение размерности
 - Визуализация
 - ...

Таксономия методов DM & ML



Задача машинного обучения с учителем

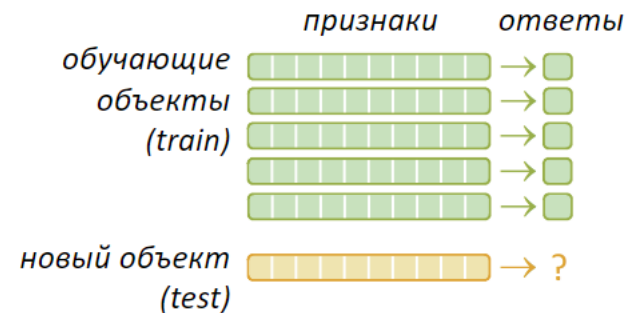
Этап №1 – обучение с учителем

- **На входе:**
данные – выборка прецедентов «объект → ответ»,
каждый объект описывается набором признаков
- **На выходе:**
модель, предсказывающая ответ по объекту

Если нет данных,
то нет
и машинного
обучения

Этап №2 – применение

- **На входе:**
данные – новый объект
- **На выходе:**
предсказание ответа на новом объекте



Классическое Обучение



Модель алгоритмов — параметрическое семейство отображений

$$A = \{g(x, \theta) \mid \theta \in \Theta\},$$

где $g: X \times \Theta \rightarrow Y$ — фиксированная функция,
 Θ — множество допустимых значений параметра θ .

Пример.

Линейная модель с вектором параметров $\theta = (\theta_1, \dots, \theta_n)$, $\Theta = \mathbb{R}^n$:

$$g(x, \theta) = \sum_{j=1}^n \theta_j f_j(x) \quad \text{— для регрессии, } Y = \mathbb{R};$$

$$g(x, \theta) = \text{sign} \sum_{j=1}^n \theta_j f_j(x) \quad \text{— для классификации, } Y = \{-1, +1\}.$$

Метод обучения (learning algorithm) — это отображение вида

$$\mu: (X \times Y)^\ell \rightarrow A,$$

которое произвольной выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ ставит в соответствие некоторый алгоритм $a \in A$.

В задачах обучения по прецедентам всегда есть два этапа:

- Этап обучения (training):
метод μ по выборке X^ℓ строит алгоритм $a = \mu(X^\ell)$.
- Этап применения (testing):
алгоритм a для новых объектов x выдаёт ответы $a(x)$.

Этап обучения (training):

метод μ по выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ строит алгоритм $a = \mu(X^\ell)$:

$$\boxed{\begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}} \xrightarrow{y} \begin{pmatrix} y_1 \\ \dots \\ y_\ell \end{pmatrix} \xrightarrow{\mu} a$$

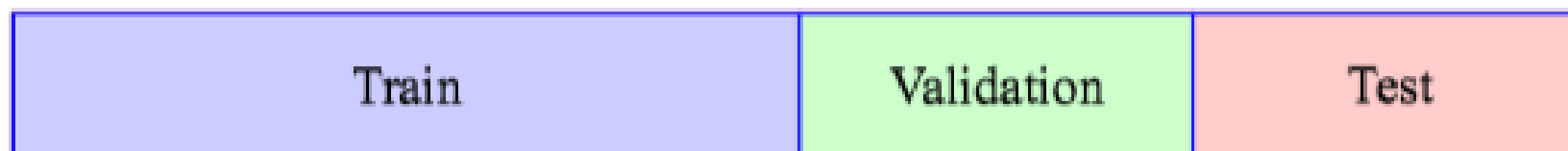
Этап применения (testing):

алгоритм a для новых объектов x'_i выдаёт ответы $a(x'_i)$.

$$\begin{pmatrix} f_1(x'_1) & \dots & f_n(x'_1) \\ \dots & \dots & \dots \\ f_1(x'_k) & \dots & f_n(x'_k) \end{pmatrix} \xrightarrow{a} \begin{pmatrix} a(x'_1) \\ \dots \\ a(x'_k) \end{pmatrix}$$

Если данных достаточно, то их делят

- на обучающую (train) выборку,
- на проверочную (validation) выборку,
- на тестовую (test) выборку.



Обучающая выборка используется для построения моделей $f(\cdot, \alpha) \in \mathcal{F}$ для разных α .

Проверочная — для оценки среднего риска каждой из построенной модели и выбора наилучшей модели в \mathcal{F} .

Тестовая — для оценки ошибки предсказания выбранной модели.

ПРОБЛЕМА 1

Вернемся к ресторанам

- Не все алгоритмы полезны
- $a(x) = 0$ — не принесет никакой выгоды
- Функция потерь — мера корректности ответа алгоритма
- Предсказали \$10000 прибыли, на самом деле \$5000 — хорошо или плохо?
- Квадратичное отклонение: $(a(x) - y)^2$

Функционал качества

- Функционал качества, метрика качества — мера качества работы алгоритма на выборке
- Среднеквадратичная ошибка (Mean Squared Error, MSE):

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2$$

- Чем меньше, тем лучше

Функционал качества

- Должен соответствовать бизнес-требованиям
- Одна из самых важных составляющих анализа данных

Обучение алгоритма

- Есть обучающая выборка и функционал качества
- Семейство алгоритмов \mathcal{A}
 - Из чего выбираем алгоритм
 - Пример: все линейные модели
 - $\mathcal{A} = \{w_1x^1 + \dots + w_dx^d \mid w_1, \dots, w_d \in \mathbb{R}\}$
- Обучение: поиск оптимального алгоритма с точки зрения функционала качества

$\mathcal{L}(a, x)$ — функция потерь (loss function) — величина ошибки алгоритма $a \in A$ на объекте $x \in X$.

Функции потерь для задач классификации:

- $\mathcal{L}(a, x) = [a(x) \neq y(x)]$ — индикатор ошибки;

Функции потерь для задач регрессии:

- $\mathcal{L}(a, x) = |a(x) - y(x)|$ — абсолютное значение ошибки;
- $\mathcal{L}(a, x) = (a(x) - y(x))^2$ — квадратичная ошибка.

Эмпирический риск — функционал качества алгоритма a на X^ℓ :

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(a, x_i).$$

ПРОБЛЕМА 1

Необходимость предобработки данных Некоторые модели хорошо работают только при выполнении определенных требований. Так, для линейных моделей крайне важно, чтобы признаки были нормированными, то есть измерялись в одной шкале. Примером способа нормировки данных является вычитание среднего и деление на дисперсию каждого столбца в матрице «объекты-признаки».

- Бывает, что в выборку попадают выбросы — объекты, которые не являются корректными примерами из-за неправильно посчитанных признаков, ошибки сбора данных или чего-то еще. Их наличие может сильно испортить модель.

Некоторые признаки могут оказаться шумовыми, то есть не имеющими никакого отношения к целевой переменной и к решаемой задаче. Примером, скорее всего, может служить признак «фаза луны в день первого экзамена» в задаче предсказания успешности прохождения сессии студентом.

Как показывает практика, простейшая предобработка данных может радикально улучшить качество итоговой модели.

ПРОБЛЕМА 2

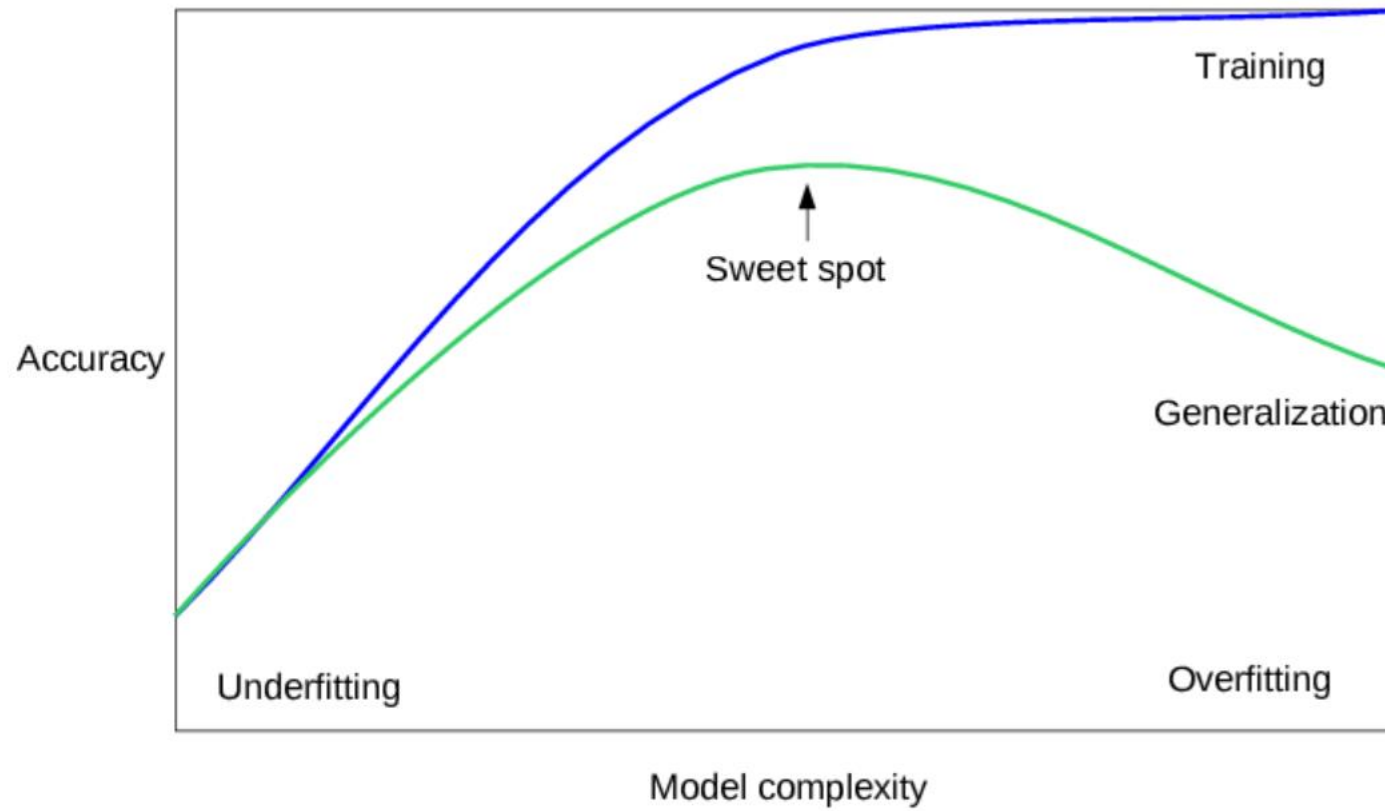
Переобучение (overfitting) - модель слишком точно подстраивается под особенности обучающего набора. В результате чего, хорошо работает на обучающем наборе, но не умеет обобщать результат на новые данные.

Недообучение (underfitting) - недостаточный охват многообразия и изменчивости данных. Модель плохо работает даже на обучающем наборе.

Регуляризация (regularization) - явное ограничение модели для предотвращения переобучения.

Classification and Regression

Generalization, Overfitting, and Underfitting



Основные этапы решения задачи анализа данных:

- 1. Постановка задачи;**
- 2. Выделение признаков;**
- 3. Формирование выборки;**
- 4. Выбор метрики качества;**
- 5. Предобработка данных;**
- 6. Построение модели;**
- 7. Оценивание качества модели.**

