Математика для машинного обучения

Линейная алгебра

Вектором размера n — набор из n вещественных чисел:

$x = (x_1, x_2, \ldots, x_n) \in R^n$,

где R — множество вещественных чисел. Элемент (число) x_i вектора x будем называеть i**-ой компонентой вектора** x.

Обратите внимание: векторы $(1, 2, 3)$ и $(3, 2, 1)$ — разные векторы, то есть порядок чисел в векторе имеет значение. Поэтому про вектор обычно говорят "упорядоченный набор чисел".

Скаляр — вектора размера 1 (вещественное число).

Скалярное произведение векторов — определяется по формуле:

$$a = (a_1, a_2, \ldots, a_n) \in R^n$$
$$b = (b_1, b_2, \ldots, b_n) \in R^n$$
$$\text{dot\_product}(a, b) = a \cdot b = \sum_{i=1}^{n} a_i * b_i$$

Вектора $x_1, x_2, \ldots, x_m$ называются **линейно зависимыми**, если существует такой набор коэффициентов (вещественных чисел) $\alpha_1, \alpha_2, \ldots, \alpha_m$, не равных нулю одновременно (говорят "ненулевой набор коэффициентов"), что выполнено:

$$\alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_m x_m = 0$$

Важно: под нулем здесь понимается именно **нулевой вектор**, равный $(0, 0, \ldots, 0)$ (n нулей).

Линейное или **векторное пространство** V над полем R действительных чисел — это упорядоченная четвёрка $(V, R, +, *)$, где V — непустое множество элементов произвольной природы, которые называются **векторами**; R — множество вещественных чисел, элементы которого называются **скалярами**; определена операция **сложения векторов** $+$ и **операция умножения векторов на скаляры** $*$. В нашем случае объекты произвольной природы — это упорядоченные наборы чисел, их мы и называем векторами.

Чтобы называться линейным (векторным) пространством, это множество должно также удовлетворять следующим свойствам:

1. $a + b = b + a, \quad a, b \in V$
2. $(a + b) + c = a + (b + c)$ — складывать можно в любой последовательности
3. Существует нулевой вектор (нейтральный по сложению элемент): $0 + a = a + 0 = a, \quad a \in V$
4. Для каждого вектора x существует обратный к нему по сложению элемент $-x$, такой что: $x + (-x) = 0$
5. $\alpha(\beta a) = (\alpha\beta)a, \quad a \in V, \alpha, \beta \in R$ — перемножать можно в любой последовательности
6. Существует единичный вектор (нейтральный по умножению элемент): $1 * a = a * 1 = \quad a \in V$
7. $(\alpha + \beta)a = \alpha a + \beta a, \quad a \in V, \alpha, \beta \in R$
8. $\alpha(a + b) = \alpha a + \alpha b, \quad a, b \in V, \alpha \in R$

Ба́зис (др.-греч. βασις «основа») — упорядоченный (конечный или бесконечный) набор векторов в векторном пространстве, такой, что любой вектор этого пространства может быть единственным образом представлен в виде линейной комбинации векторов из этого набора (базиса). Векторы базиса называются **базисными векторами**.

Стандартный базис:

$$
\begin{aligned}
e_1 &= (1, 0, \ldots, 0, 0, 0, \ldots, 0) \\
e_2 &= (0, 1, \ldots, 0, 0, 0, \ldots, 0), \\
&\ldots, \\
e_i &= (0, 0, \ldots, 0, 1, 0, \ldots, 0), \\
&\ldots \\
e_n &= (0, 0, \ldots, 0, 0, 0, \ldots, 1)
\end{aligned}
$$

где в векторе e_i единица стоит только на i-ом месте, а остальные компоненты — нули.

Модуль или **евклидова норма** вектора размера n -- вещественное число, равное:

$$||x|| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

Косинус угла между двумя векторами:

$$\text{cosine\_similarity}(a, b) = cos(a, b) = \frac{a \cdot b}{||a|| * ||b||}$$

где $a \cdot b$ — скалярное произведение векторов a и b, $||a||$ и $||b||$ — евклидовы нормы векторов. Эту величину еще называют **косинусной мерой похожести** между векторами (**cosine similarity**)

Координаты вектора — коэффициенты в линейной комбинации стандартных базисных векторов, образующей этот вектор.

Матрицей размера $n \times m$ мы будем называть набор из $n \times m$ вещественных чисел, где все эти числа **записаны в строки и стобцы матрицы**:

$$A = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \ldots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \ldots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \ldots & x_{nm} \end{bmatrix} \in R^{n \times m}$$

где R — множество вещественных чисел. В данном случае матрица имеет n строк и m столбцов. Элементами матрицы x_{ij} являются вещественные числа. Если взять одну строку матрицы — получим числовой вектор размера m, если взять один столбец — тоже получим числовой вектор, но уже размера n.

Количество строк матрицы, стоящее в $n \times m$ первым множителем, будем называть **первой размерностью**, второй множитель (количество столбцов) будем называть **второй размерностью** матрицы. Их еще часто называют **осями** (***axis*** по-английски)

Матрицу, у которой количество строк равно количеству столбцов ($n = m$) называют **квадратной**. Все остальные матрицы называют **прямоугольными**.

Транспонирование матрицы — операция, для выполнения котороый строки записываются на место столбцов, а столбцы — на место строк:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nm} \end{bmatrix} \in R^{n \times m}$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ a_{13} & a_{23} & \dots & a_{n3} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{bmatrix} \in R^{m \times n}$$

Пусть есть две матрицы (обратите внимание на размеры матриц):

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nk} \end{bmatrix} \in R^{n \times k}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1m} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & b_{k3} & \dots & b_{km} \end{bmatrix} \in R^{k \times m}$$

Тогда **матричным произведением** этих матриц является матрица:

$$A \times B = C = \begin{bmatrix} A[1,:] \cdot B[:,1] & A[1,:] \cdot B[:,2] & A[1,:] \cdot B[:,3] & \dots & A[1, \\ A[2,:] \cdot B[:,1] & A[2,:] \cdot B[:,2] & A[2,:] \cdot B[:,3] & \dots & A[2, \\ \vdots & \vdots & \vdots & \ddots & \\ A[n,:] \cdot B[:,1] & A[n,:] \cdot B[:,2] & A[n,:] \cdot B[:,3] & \dots & A[n, \end{bmatrix}$$

то есть один элемент c_{ij} матрицы C равняется: $c_{ij} = A[i,:] \cdot B[:,j]$, где:
$A[i,:]$ — i-ая строка матрицы A,
$B[:,j]$ — j-ый столбец матрицы B,
$A[i,:] \cdot B[:,j]$ — скалярное произведение векторов $A[i,:]$ и $B[:,j]$

Простыми словами: чтобы получить элемент матрицы, являющейся результатом матричного произведение, нам надо **в правильном порядке скалярно умножать строки первой матрицы на столбцы второй матрицы**.

Матрица называется **вектор-столбец**, если она имеет размер $n \times 1$, и **вектор-строка**, если она имеет размер $1 \times n$. Понятно, что это тоже матрицы, просто по сути они являются векторами.

Обратной матрицей к квадратной матрице $A \in R^{n \times n}$ называется такая матрица $A^{-1} \in R^{n \times n}$, что:

$$A \times A^{-1} = A^{-1} \times A = E_n$$

где $E_n \in R^{n \times n}$ — **единичная матрица** размера n:

$$E_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Элементы матрицы a_{ij}, стоящие на позициях $i = j$ (то есть элементы a_{11}, a_{22}, ..., a_{nn}) называют **главной диагональю** матрицы A. Главная диагональ есть не только у квадратных матриц — просто у прямоугольных последний элемент диагонали будет на позиции $a_{min(n,m),min(n,m)}$.

Математический анализ

Пусть X и Y — два множества.

Функция, заданная на X со значениями в Y – закон F, согласно которому каждому элементу $x \in X$ поставлен в соответствие единственный элемент $y \in Y$.

y является **образом** элемента x при функции F, x является **прообразом** элемента y при функции F.

x называется **переменной** или **аргументом функции F**.

X называется **областью определения** функции F, Y называется **областью значений** функции F.

y_{min} называется **глобальным минимумом функции F**, x_{min} — **аргминимумом** (минимальным аргументом), если не существует такого $x \neq x_{min}$, что $F(x) < F(x_{min})$

Точка x_{min} — **локальный минимум** функции F, если существует такая величина $\Delta x > 0$, что значение функции F в любой точке $x \in [x_{min} - \Delta x, x_{min} + \Delta x]$ не меньше значения функции в точке x_{min}.

Точки локальных минимумом и максимумов также называются **точками экстремума**.

Функция F **непрерывна в точке** x, если:

1. F определена в точке x
2. Для любого числа $\delta > 0$ можно найти такое $\Delta x > 0$, что для любой точки $x' \in [x - \Delta x, x + \Delta x]$ выполняется: $|F(x') - F(x)| < \delta$ ($|F|$ означает модуль F)

В точках, где функция не непрерывна, говорят, что функция имеет **разрыв**.

Рассмотрим некоторую бесконечную последовательность чисел $x_1, x_2, \ldots, x_n, \ldots$

Число x является **пределом последовательности**, если для любого числа $\delta > 0$ сушествует такой номер n, что для любого числа $N > n$ выполняется: $|x - x_N| < \delta$
Если x является **пределом последовательности**, то говорят, что последовательность **стремится** к x.

Число y является **пределом функции** F **в точке** x, если для любого числа \delta > 0 можно найти такое $\Delta x > 0$, что для любой точки $x' \in [x - \Delta x, x + \Delta x]$ выполняется: $|F(x') - y| < \delta$

Производная функции F в точке x

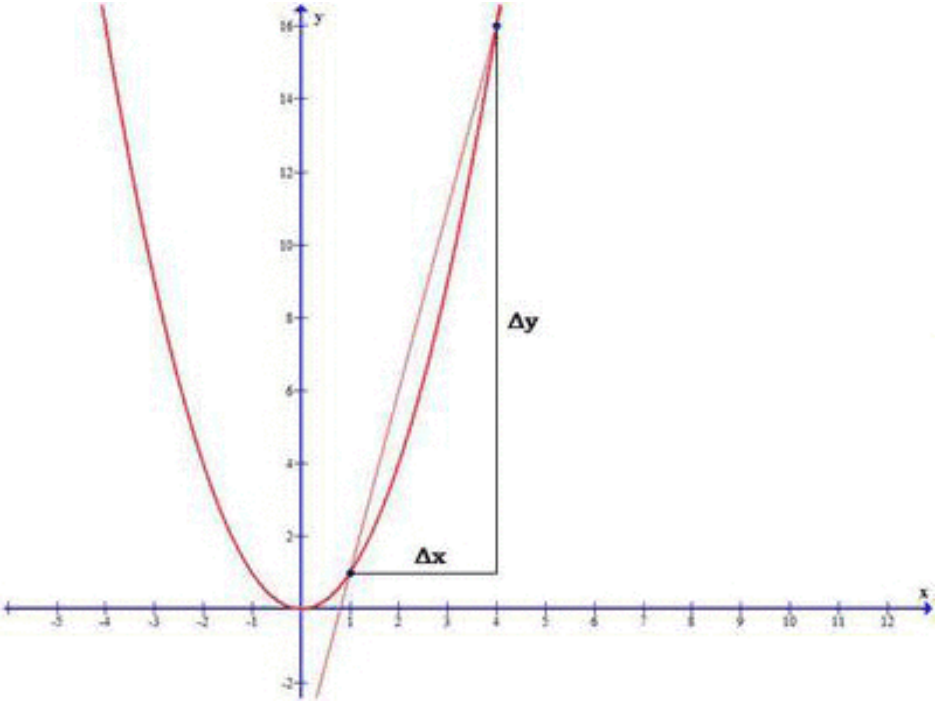
$$F'(x) = \lim_{\Delta x \to 0} \frac{F(x + \Delta x) - F(x)}{\Delta x}$$

Производная функции в точке показывает характер изменения функции в точке, а именно:

Модуль значения производной говорит о скорости убывания/ возрастания функции.

Знак производной показывает характер изменения функции в точке.

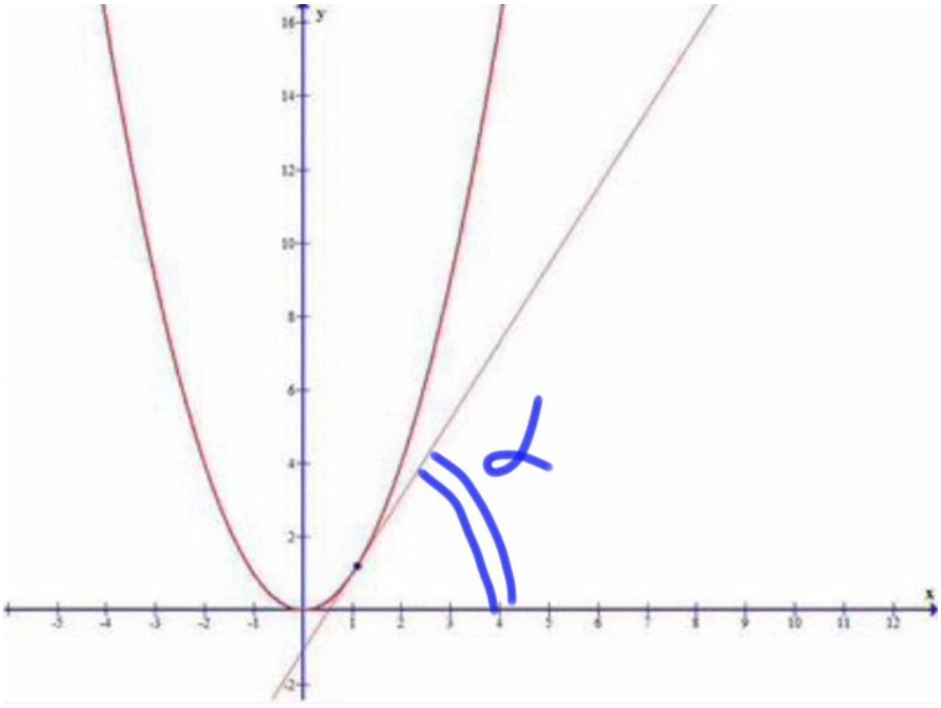
Производная функции есть тоже функция.



(https://imgbb.com/)

Прямая, проходящая через две точки графика функции, называется **секущей** (см рисунок сверху).

Касательная к графику функции F в точке (x, y) — это предельное положение секщей в этой точке.



(https://ibb.co/bdt4tyX)

Производная функции в точке (x, y) численно равна тангенсу угла наклона α касательной в это точке.

Таблица производных часто встречающихся функций:

$C' = 0$	$(\arcsin x)' = \dfrac{1}{\sqrt{1-x^2}}$
$x' = 1$	$(\arccos x)' = -\dfrac{1}{\sqrt{1-x^2}}$
$(x^n)' = n \cdot x^{n-1}$	$(\operatorname{arctg} x)' = \dfrac{1}{1+x^2}$
$(\sqrt{x})' = \dfrac{1}{2\sqrt{x}}$	$(\operatorname{arcctg} x)' = -\dfrac{1}{1+x^2}$
$(e^x)' = e^x$	$(\operatorname{sh} x)' = \operatorname{ch} x$
$(a^x)' = a^x \ln a$	$(\operatorname{ch} x)' = \operatorname{sh} x$
$(\ln x)' = \dfrac{1}{x}$	$(\operatorname{th} x)' = \dfrac{1}{\operatorname{ch}^2 x}$
$(\log_a x)' = \dfrac{1}{x \ln a}$	$(\operatorname{cth} x)' = -\dfrac{1}{\operatorname{sh}^2 x}$
$(\sin x)' = \cos x$	$(\operatorname{arcsh} x)' = \dfrac{1}{\sqrt{x^2+1}}$
$(\cos x)' = -\sin x$	$(\operatorname{arcch} x)' = \dfrac{1}{\sqrt{x^2-1}}$
$(\operatorname{tg} x)' = \dfrac{1}{\cos^2 x}$	$(\operatorname{arcth} x)' = \dfrac{1}{1-x^2}$
$(\operatorname{ctg} x)' = -\dfrac{1}{\sin^2 x}$	$(\operatorname{arccth} x)' = \dfrac{1}{1-x^2}$

Правила нахождения производной композиции функций:

1. Производная суммы двух функций есть сумма производных этих функций (аналогично с вычитанием)
$$(u + v)' = u' + v'$$

2. Производная произведения двух функций:
$$(u \times v)' = u'v + uv'$$

3. Производная отношения двух функций:
$$\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$$

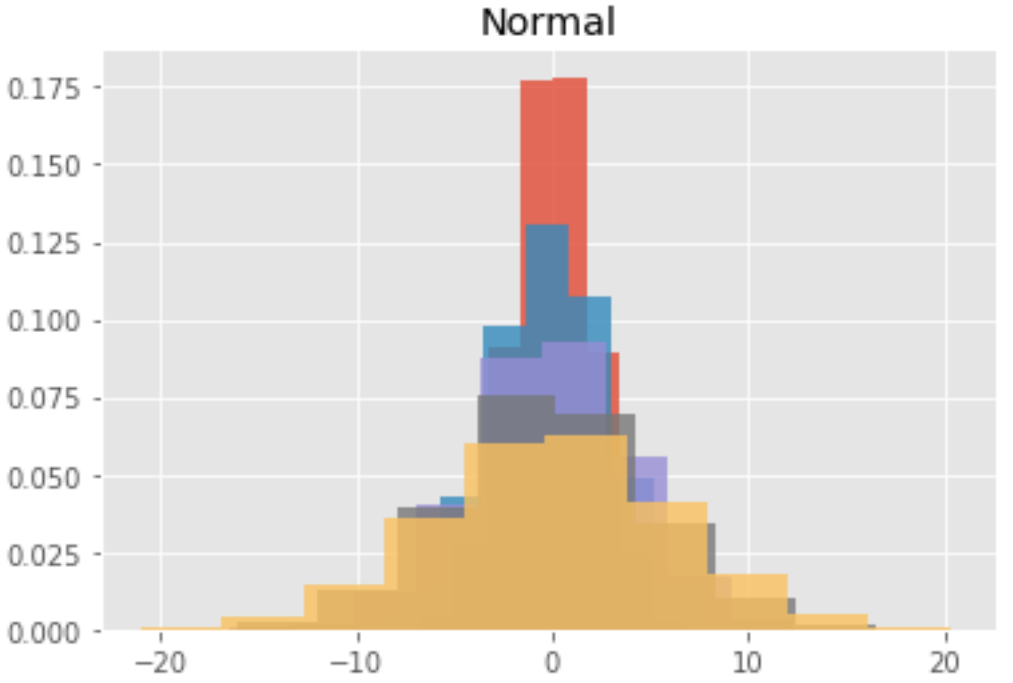
Нахождение производной сложной функции:
$$(u(v(x)))' = u'(v) * v'(x)$$

Градиент функции многих переменных — это вектор частных производных функции.

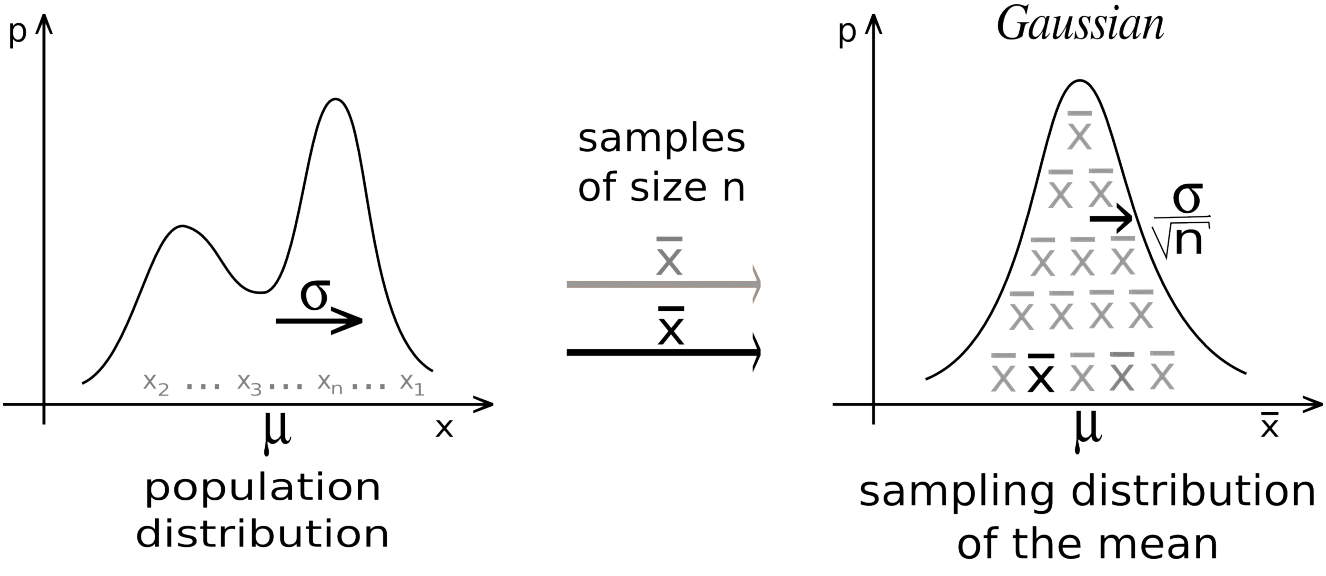
Теория вероятностей и статистика

1. Элементы $\omega \in \Omega$ называют элементарными исходами некоторого эксперимента, а Ω — пространством элементарных исходов
2. Подмножество $A \subset \Omega$ назовем **событием**, произошедшим, если эксперимент закончился одним из элементарных исходов ω, входящих в множество A.
3. Алгебра множеств — это непустая система подмножеств, замкнутая относительно операций дополнения (разности) и объединения (суммы). алгебра множеств, замкнутая относительно операции счётного объединения, называется сигма-алгеброй
4. Функция $P : \mathscr{F} \to \mathbb{R}$ называется *вероятностной мерой* или *вероятностью*, если она принимает значения от нуля до единицы и аддитивна по $\mathscr{F}$.
5. Тройка: вероятностное пространство, сигма-алгебра и вероятность образуют **вероятностное пространство**
6. Условной вероятностью A при условии B, причем $P(B) > 0$, называется $P(A|B) = \frac{P(AB)}{P(B)}$. Здесь $P(AB)$ — вероятность одновременного наступления событий A и B.

7. **Формула полной вероятности**: Назовем *разбиением* систему событий $\{A_1, \ldots, A_n\}$ разбиением, если никакие два события из разбиения не пересекаются и покрывают все множество элементарных исходов Ω, причем $P(A_i) > 0, \forall i \in 0..n$. Тогда для любого события B верна формула: $P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$
8. **(формула Байеса:** $P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$, $P(B) > 0$ Заметим, что если использовать формулу полной вероятности, то можно записать формулу Байеса в виде: $P(A_i|B) = \frac{P(B|A_i)\ P(A_i)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)}$
9. События A и B назовем **независимыми**, если выполнено утверждение: $P(AB) = P(A)P(B)$
10. Вероятностная мера P, заданная на $(\mathbb{R}, \mathscr{B}(\mathbb{R}))$, называется распределением вероятностей. Здесь $\mathscr{B}(\mathbb{R})$ — борелевская сигма-алгебра — система из отрезков, (полу)интервалов на множестве действительных чисел $\mathbb{R}$.
11. Функция $F : \mathbb{R} \to [0, 1]$ $F(x) = P((-\infty, x])$ называется функцией распределения, соответствующей распределению вероятностей P.
12. Функция $p(t), t \in \mathbb{R}$, такая, что $\int_{\mathbb{R}} p(t)dt = 1$ и $F(x) = \int_{-\infty}^{x} p(t)dt$, называется *плотностью абсолютно непрерывного распределения* P, соответсвующего функции распределения F.
13. **Абсолютно непрерывными** называют распределения, имеющие плотность вероятности.
14. **Дискретными** называют распределения величин, принимающих не более чем счетное число значений.
15. **Случайная величина** — переменная, ее значения представляют собой исходы какого-нибудь случайного эксперимента. Другими словами, это численное выражение результата случайного события. Часто обозначается $\xi(x)$. В основной теории вероятностей(не уровня ликбеза) значительная часть теорем связана именно с этим понятием. В частности, закон больших чисел или центральная предельная теорема.



16. **Математическое ожидание** (взвешенное по вероятностям возможных значений) значение — наиболее ожидаемый исход
17. Дисперсия — мера разброса результатов эксперимента относительно её математического ожидания
18. **Закон больших чисел (ЗБЧ)** — принцип, описывающий результат выполнения одного и того же эксперимента много раз. Согласно закону, среднее значение конечной выборки из фиксированного распределения близко к математическому ожиданию этого распределения.
19. **Центральная предельная теорема** утверждает, что сумма достаточно большого количества слабо зависимых случайных величин без явно доминирующих слагаемых имеет распределение, близкое к нормальному.



Основы машинного обучения

Данные — информация, хранящаяся на электронном устройстве в некотором объектном виде (чаще всего в виде файла определенного расширения). Когда говорят про какой-либо набор данных в машинном обучении, он всегда имеет **конкретную природу**:

- табличные данные (структурированная информация)
- изображения / видео (визуальная информация)
- аудиозаписи (аудиоинформация)
- тексты (текстовая информация)

Выборка (Sample) — набор данных, наблюдаемых в эксперименте.

Обучающая выборка (Train set, Training Sample) — данные, используемые для обучения.

Валидационная (контрольная) выборка (Validation set, Val Sample) — данные используемые, для проверки обобщающей способности выбранного алгоритма.

Тестовая выборка (Test set, Test Sample) — данные, используемые для выбора алгоритма

Объекты (Objects) — сущности, для которых строятся предсказания. Обозначается X.

Множество допустимых ответов — множество значений, которые можно получить на выходе модели. Например если выполняем бинарную классификацию, то это будут 0 и 1. Обозначается Y.

Признаки (Фичи, Features) — результат измерения некоторой характеристики объекта. Обозначается f.

Истинные ответы (Ground Truth, GT) — результаты эксперимента, которые используем в обучающей, тестовой и валидационной выборках.

Предсказания модели (Predictions) — элементы из Y, которые появляются в результате отработки алгоритма для очередного элемента из X.

Задано множество объектов X, множество допустимых ответов Y , и существует целевая функция (target function) $y^* : X \to Y$, значения которой $y_i = y^*(x_i)$ известны только на конечном подмножестве объектов

$x_1, \ldots, x_l \subset X$. Пары «объект – ответ» (x_i, y_i) называются прецедентами. Совокупность пар $X_l = (x_i, y_i)_{i=1}^{l}$ называется обучающей выборкой (training sample).

Задача обучения по прецедентам заключается в том, чтобы по выборке $\{X_l\}$ восстановить зависимость y^* , то есть построить решающую функцию (decision function) $a : X \to Y$, которая приближала бы целевую функцию $y^*(x)$, причём не только на объектах обучающей выборки, но и на всём множестве X. Решающая функция a должна допускать эффективную компьютерную реализацию; по этой причине будем называть её алгоритмом.

Машинное обучение (Machine Learning) — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

Модель (алгоритм) машинного обучения — в задачах обучения по прецедентам — алгоритм μ, который принимает на входе обучающую выборку данных D, строит и выдаёт на выходе функцию $a(x) : X \to Y$ из заданной модели F, реализующую отображение из множества объектов X во множество ответов Y.

Логические модели (алгоритмы)— модели, активно использующие условие информативности.

Метрические модели (алгоритмы) — модели, предполагающие что геометрически близким объектам соответсвуют одинаковые метки классов.

Линейные модели (алгоритмы) — модели, линейные по параметрам.

Композиция моделей (алгоритмов) (Ансамбль, Ensemble) — комбинация нескольких слабых алгоритмов (например путем голосования) для получения одного сильного. **Параметры модели (алгоритма)** — характеристики, которые настраиваются в процессе обучения. Например — коэффициенты линейной регрессии.

Гиперпараметры (Hyperparameters) — характеристики, которые настраиваются до начало обучения. Например — число слоев в нейронной сети.

Скорость обучения (Learning Rate) — гиперпараметр, который характеризует скорость сходимости оптимизации. При больших значениях алгоритм сойдется быстрее, но может оказаться менее точным.

Функция потерь — $L(a(x), y^*(x))$ характеризует величину нестыковок между результатами предсказания модели и реальными значениями.

Оптимизатор (алгоритм оптимизации) — алгоритм, осуществляющий минимизацию функции потерь.

Переобучение — явление, когда функция потерь на обучающей выборке значительно меньше, чем на тестовой. Если алгоритм переобучен, то говорят, что у него низкая обобщающая способность

Этап обучения (Training, Train time) — процесс, состоящий из настройки параметров путем анализа прецедентов из обучающей выборки.

Этап предсказания (Inference, Test time) — процесс прохода через модель с уже настроенными параметрами новых данных и получения в результате предсказаний из множества допустимых значений Y.

Метрика качества (Metric, Quality Metric) — функция, позволяющая оценить качество алгоритма. Бывают разными в зависимости от специфики задач. Классические примеры: accuracy, precision-recall, f-мера, ROC_AUC.

Кросс-валидация (Cross-Validation, CV) — процедура эмпирического оценивания обобщающей способности алгоритмов машинного обучения. Фиксируется некоторое множество разбиений исходной выборки на две подвыборки: обучающую и валидационную. Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах валидационной подвыборки. Оценкой по кросс-валидации называется средняя по всем разбиениям величина ошибки на валидационных подвыборках.

Кривая обучения (Learning Curve) — график ошибки, который используется для отслеживания процесса обучения.

SOTA (SotA, State-of-the-Art) — модель (алгоритм), являющейся лучшей в конкретной задаче в соответствие с какой-либо фиксированной метрикой (метрикам). Например: "EfficientNet — SOTA в задаче классификации изображений по метрике Top-5 Accuracy".

Основы нейронных сетей и глубокого обучения

Функция активации (Activation Function) — функция, аргументом которой является линейная комбинация входов в нейрон/перцептрон. Чаще всего функции активации — нелинейные функции (ReLU, Tanh, Sigmoid, ELU, Swish, LeackyReLU...). Существует распространенное мнение (опирающееся исключительно на опыт экспериментов), что именно нейлинейность позволяет моделировать сложные зависимости в данных.

Перцептрон (Perceptron) — математическая или компьютерная модель восприятия информации мозгом (кибернетическая модель мозга), предложенная Фрэнком Розенблаттом в 1957 году и впервые реализованная в виде электронной машины «Марк-1» в 1960 году. Ключевая "фишка" перцептрона — взвешивание сигналов и использование **пороговой функции активации** для предсказания.

Нейрон (Neuron) — модель (алгоритм) машинного обучения, осуществляющий следующие операции над входом:

$$out = f(X \cdot w)$$

где w — вектор весов (параметров) нейрона, X — вход нейрона, f — функция активации, out — выход нейрона на данном входе X

Нейронная сеть (Neural Network) — модель (алгоритм) машинного обучения, являющий композицией (ансамблем) нейронов. Нейронная сеть состоит из **слоёв**.

Слой нейронной сети (Neural Network Layer) — набор нейронов, связанных между собой только логически. Параметры (веса) одного слоя являются матрицей, каждый столбец которой есть веса одного нейрона из набора. Так, если в слое L нейронов, а на вход поступают объекты, имеющие F признаков, матрица весов (параметров) слоя нейронной сети имеет размер $F \times L$.

Архитектура нейронной сети (Architecture of Neural Network) — определенный набор слоев конкретного типа, названный одним именем. Например (названия архитектур): ResNet, WaveNet, Megatron.

Глубокая нейронная сеть (Deep Neural Network) -- нейронная сеть, имеющая большое количество слоев. Точная цифра, после которой сеть считается глубокой, не определена, однако в целом считается, что если слоев более 12, то сеть уже не "мелкая".

Глубокое обучение (Deep learning) — раздел машинного обучения, в рамках которого исследуются глубокие нейронные сети и все, что с ними связано.

Полносвязный слой (Fully-Connected Layer, FC) — тип слоя нейронной сети, имеющий два ключевых свойства:

1. Каждый нейрон в слое осуществляет только линейное преобразование и функцию активации от этого преобразования
2. Каждый нейрон FC-слоя передает свой выход каждому нейрону следующего слоя (связь "все со всеми", поэтому и "полносвязный")

Входной слой (Input Layer) — абстрактное название для данных, поступающих на вход нейронной сети. Входной слой не имеет весов (параметров), а лишь представляет собой сами данные, которые будет обрабатывать нейронная сеть.

Скрытый слой (Hidden Layer) — слой нейронной сети, не являющийся входным или выходным. Чаще всего скрытые слои имеют веса (параметры) и осуществляют основную работу нейронной сети.

Выходной слой (Output Layer) — последний (если смотреть "слева-направо") слой нейронной сети, выход которого либо подается в функцию потерь (при обучении), либо является ответом на поставленную нейронной сети задачу (при inference).

Многослойный перцептрон (Multilayered Perceptron, MLP) — нейронная сеть, состоящая из нескольких полносвзяных слоев.

Обучение нейронной сети — процесс изменения весов (параметров) нейронной сети так, что при предсказании с помощью обновленных весов функция потерь стала меньше.

Алгоритм обратного распространения ошибки (Error Backpropagation Algorithm, Backprop) — алгоритм обновления весов нейронной сети, подразумевающий использование выходов более поздних слоев сети при обнолвении более ранних. Таким образом, ошибка как бы "распространяется обратно", проходя путь **от выходного слоя ко входному**, меняя веса всех скрытых слоев.

Сверточные нейронные сети

Тензор — матрица, имеющая произвольно много размерностей. Например, если взять K матриц размера $N \times M$ каждая, получим тензор размера $N \times M \times K$ (с тремя размерностями).

Фильтр (окно) свертки (Convolution Filter) — тензор, имеющий фиксированный размер (обычно $F \times F \times 3$, где F — число меньше 10). Числа внутри этого тензора называются **весами фильтра (окна) свертки**.

Операция свертки (Convolution) — операция, в ходе которой фильтр (окно) свертки применятеся ко входному тензору специальным образом: веса фильтра на каждом шаге поэлементно умножаются на подтензор этого тензора и складываются.

Операция подвыборки (Пулинга, Pooling) — операция, в ходе которой берется лишь часть элементов тензора, поданного на вход. Причем эти элементы берутся по некоторому конкретному правилу (например, максимум при Max Pooling'e, или среднее при Average Pooling'e). Чаще всего пулинг — это извлечение максимума/среднего элементов, попадающих под **окно пулинга** (область фиксированноо размера, например, окно 2x2).

Карта признаков (Feature Map) — тензор, являющийся результатом операции свертки.

Сверточный нейрон (Convolutional Neuron) — один элемент карты признаков.

Рецептивное поле (Receiptive Field) одного сверточного нейрона — размер окна, которое получится, если "сложить" (с перекрытиями) все окна сверток от входа нейронной сети до рассматриваемого сверточного нейрона.

Сверточный слой (Convolutional Layer) — набор фильтров сверток, каждый из которых имеет свой фиксированный размер и правило, по которому этот фильтр будет сворачивать тензор, подающийся на вход этому сверточному слою. **Весами сверточного слоя** является набор весов всех фильтров этого сверточного слоя.

Пулинг слой (Pooling Layer) — набор окон пулинга (подвыборки), каждое из которых имеет свой фиксированный размер. Пулинг-слой не имеет параметров, а только гиперпараметры — размеры окон пулинга.

Глобальный пулинг (Global Pooling) — операция подвыборки, результатом которой является тензор, у которого одна из размерностей равна 1. Простыми словами: глобальный пулинг берет максимум/среднее **всех** элементов в тензоре, а не только в рамках окна.

Сверточная нейронная сеть (Convolutional Neural Network, CNN) — нейронная сеть, в которой есть сверточный слои. Чаще всего в сверточных нейросетях так же есть пулинг-слои и полносвязные слои.

Компьютерное зрение (Computer Vision) — раздел Computer Science, в рамках которого изучаются подходы к обработке изображений, видео, медиаконтента и любой визуальной информации.

Изображение (Image) — информация, структурированная таким образом, чтобы электронное устройсвто могло ее отображать (выводить) на экран. Изображения чаще всего состоят из **пикселей** — маленьких цветовых элементов, из которых складывается нужная картинка. ПРи работу с изображениями в Python они чаще всего представляются в виде числового тензора с 3-мя размерностями: высота (H), ширина (W) и количество цветовых каналов (C), то есть размер тензора -- $H \times W \times C$. Пара $H \times W$ называется **разрешением** изображения.

Видео (Video) — набор изображений, упорядоченных во времени. Звук в видео считается отдельной компонентой и обычно игнорируется в компьютерном зрении.

Рекуррентные нейронные сети

NLP (Natural Language Processing) -- Задача обработки текстов/звуков/любых последовательностей на естественном языке.

RNN (Recurrent Neural Network) -- вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки..

GRU (Gated Recurrent Unit) -- Тип нейрона рекуррентной нейронной сети, чуть сложнее, чем обычный RNN Cell

LSTM (Long-Short Term Memory) -- Тип нейрона рекуррентной нейронной сети, сложнее, чем обычный RNN Cell и GRU. Является state-of-the-art рекуррентных сетей.

Bidirectional RNN (двунаправленный RNN) -- тип слоя рекуррентной нейронной сети, где нейроны разворачиваются во времени с двух сторон -- слева направо и справа налево. Это позволяет улавливать больше зависимостей в тесте.

Backpropagation Through Time (обратное распространение ошибки сквозь время) -- техника backpropagation для рекуррентных нейронных сетей. Отличается от обычного backpropagation тем, что для рекуррентных нейронов необходимо вычислить градиенты рекурсивно во все моменты времени.

Проблема затухающих градиентов -- проблема, часто встречающая при обучении рекуррентных нейронных сетей, состоящая в том, что при наличии длинных зависимостей (например, когда обрабатывается длинное предложение со сложными грамматическими зависимостями), сеть к концу обработки предложения "забывает" ее начало и слова выходят не связанными между собой.

Sentiment analysis -- задача оценивания тональности текста.