НЕТОЛОГИЯ


# Юстина Иванова

## Специалист по анализу данных


Математика для data science:

Спикер

**Юстина Иванова,**

- Специалист по анализу данных «ОЦРВ», Сочи
- Инженер-программист МГТУ им. Баумана,
- Магистр по программе «Искуственный интеллект» Университета Саутгемптон

# Дифференциальное уравнение n-го порядка

Обыкновенное дифференциальное уравнение n-го порядка называется уравнение

$$F(x, y(x), y'(x), y''(x), \ldots, y^n(x)) = 0,$$

где F-известная функция (n+2)-х переменных, x – независимая переменная из интервала (a,b), y(x) – неизвестная функция. Число n – порядок уравнения.

Функция y(x) называется решением (или интегралом) дифференциального уравнения на промежутке (a,b), если она n раз дифференцируема на (a,b) и при подстановке в уравнение обращает его в тождество.

Диф. Уравнения имеют бесконечно много решений. Чтобы выделить единственное решение, задают дополнительные ограничения.

# Оптимизационная задача

Какие параметры модели дают минимальную ошибку?

Модель линейной регрессии – это прямая вида $y=kx+b$

# Частная производная

Частная производная – это предел отношения приращения функции по выбранной переменной к приращению этой переменной, при стремлении этого приращения к нулю

$$\frac{\partial f}{\partial x_k}(a_1, \ldots, a_n) = \lim_{\Delta x \to 0} \frac{f(a_1, \ldots, a_k + \Delta x, \ldots, a_n) - f(a_1, \ldots, a_k, \ldots, a_n)}{\Delta x}$$

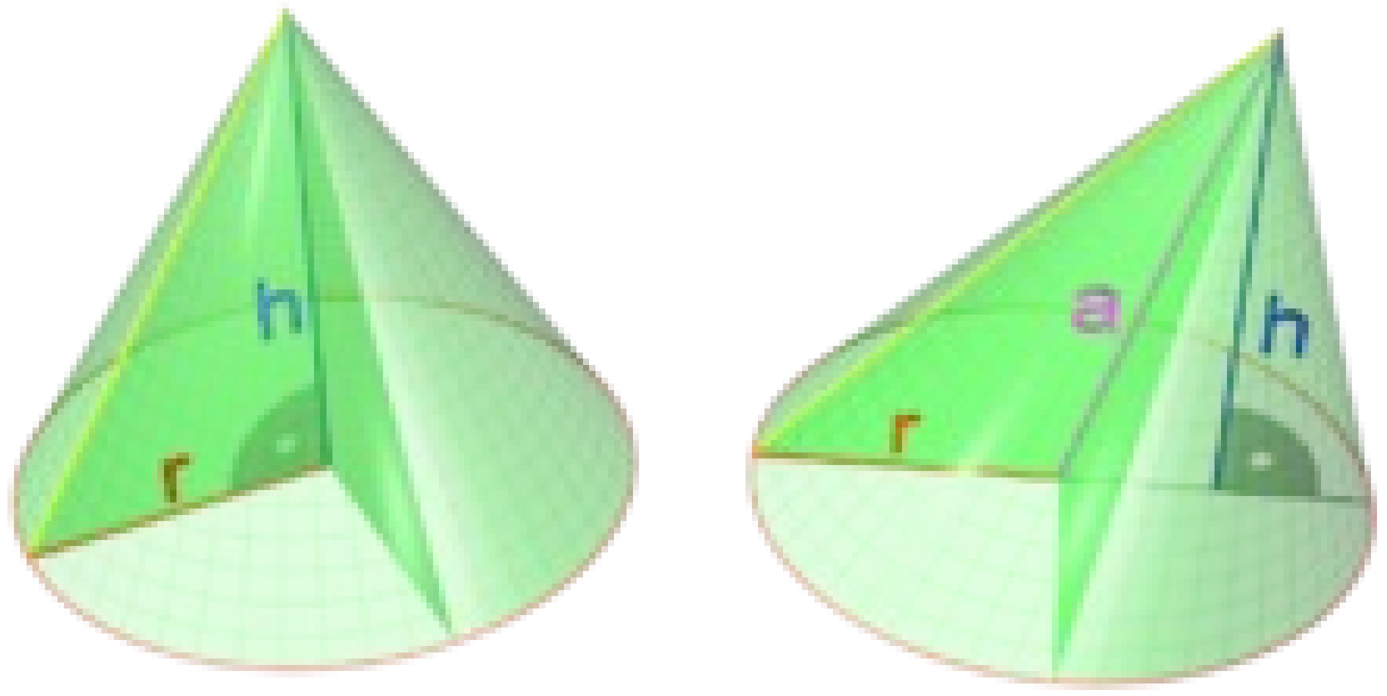
Геометрически, частная производная даёт производную по направлению одной из координатных осей.

Частная производная функции f в точке $\vec{x} = (x_1^0, \ldots, x_n^0)$ по координате $x_k$ равна производной $\frac{\partial f}{\partial \vec{e}}$ по направлению $\vec{e} = \vec{e}^k = (0, \ldots, 0, 1, 0, \ldots, 0)$, где единица стоит на k-м месте.

ru.wikipedia.org/wiki/Частная_производная

# Пример частной производной

Объём конуса V зависит от высоты h и радиуса r:




$$V = \frac{\pi r^2 h}{3}$$

Частная производная объема V относительно радиуса r:

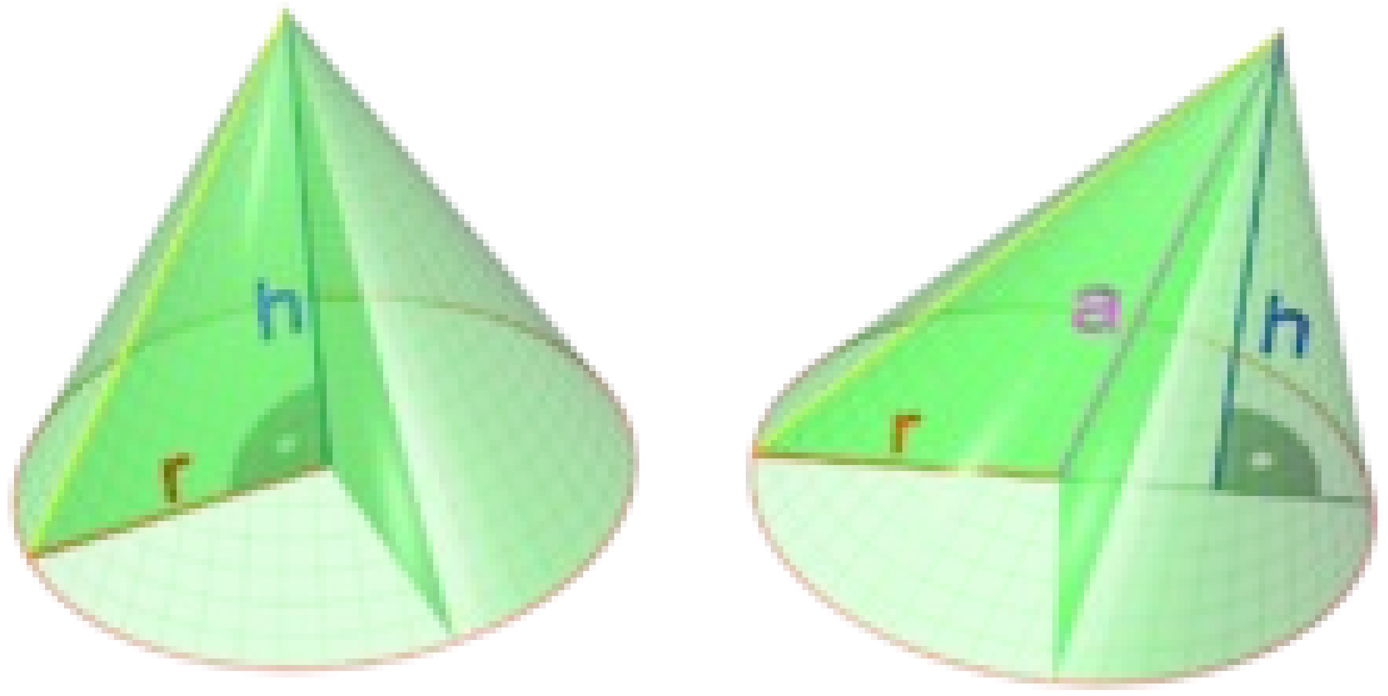
$$\frac{\partial V}{\partial r} = \frac{2\pi r h}{3}$$

Показывает скорость изменения объёма, если радиус конуса меняется, а высота неизменна.

ru.wikipedia.org/wiki/Частная_производная

# Полная производная

Полная производная V относительно r и h:




$$\frac{dV}{dr} = \frac{\partial V}{\partial r} + \frac{\partial V}{\partial h}\frac{dh}{dr} = \frac{2\pi r h}{3} + \frac{\pi r^2}{3}\frac{dh}{dr}$$

$$\frac{dV}{dh} = \frac{\partial V}{\partial h} + \frac{\partial V}{\partial r}\frac{dr}{dh} = \frac{2\pi r^2}{3} + \frac{2\pi r h}{3}\frac{dr}{dh}$$

ru.wikipedia.org/wiki/Частная_производная

# Логистическая регрессия.

Задача логистической регрессии – определить вероятность принадлежности к классу.

Построена на основе линейной функции.

$$h(x) = \theta^T x$$

К линейной функции применяется функция активации:

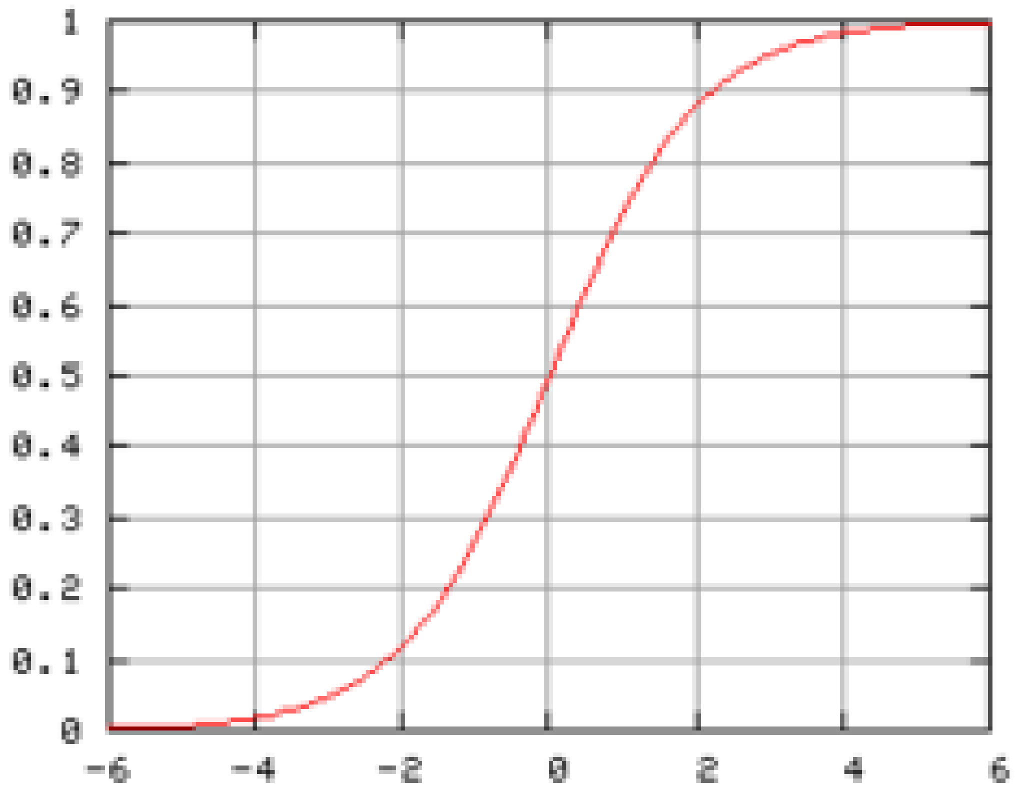
$$h(x) = \sigma(\theta^T x)$$

Функция активации:

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24

# Сигмоида.




$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Производная сигмоиды:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

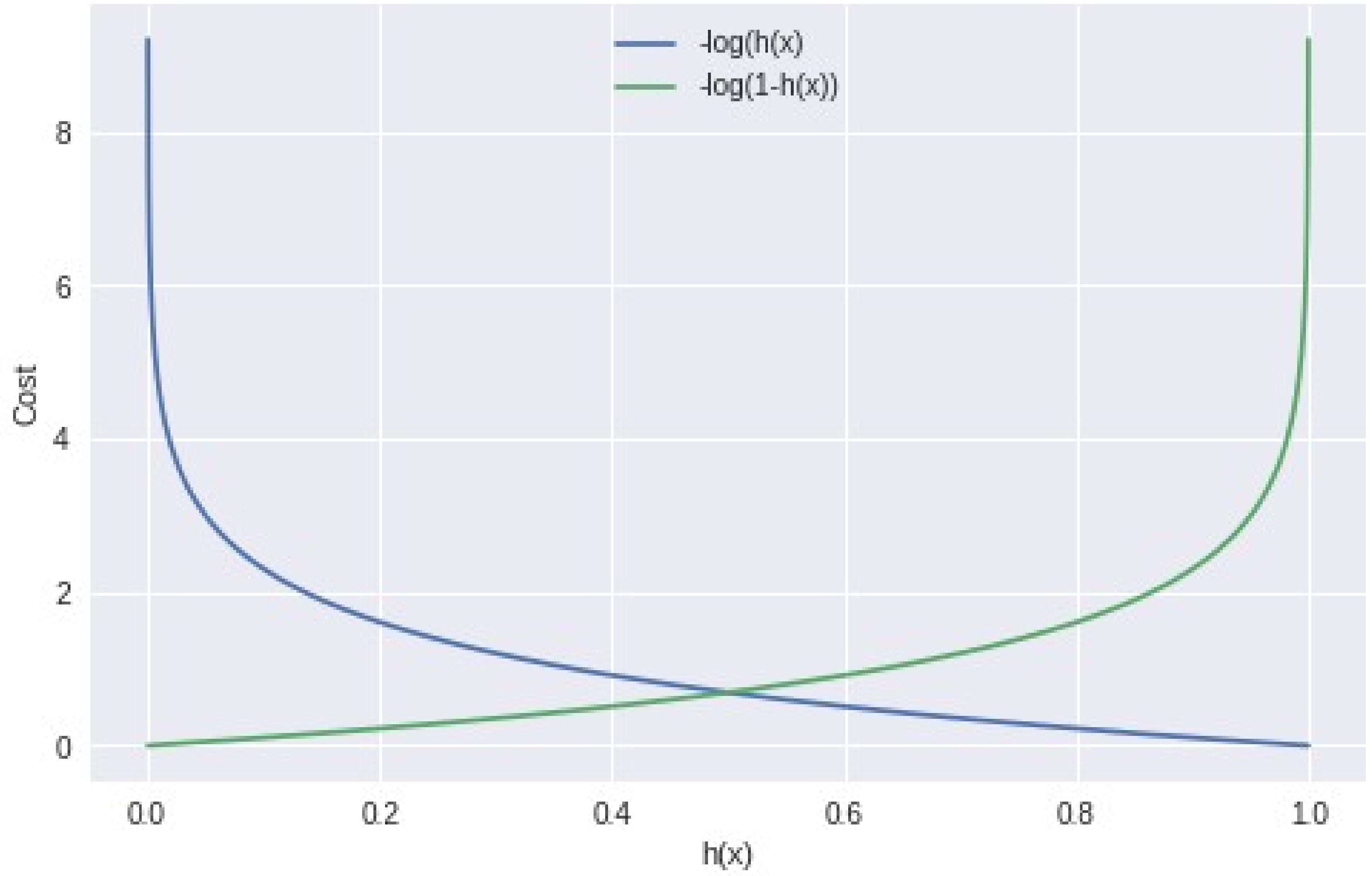
https://ru.wikipedia.org/wiki/Сигмоида

# Функция ошибки в логистической регрессии.

Модель ищет параметры, которые минимизируют функцию ошибки:

$$cost = \begin{cases} -log(h(x)), & if \quad y = 1 \\ -log(1 - h(x)), & if \quad y = 0 \end{cases}$$

Чем выше вероятность определения класса 1 при верном классе 0, тем выше стоимость ошибки.




https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24

# Функция ошибки в логистической регрессии.

Общий вид функции ошибки для модели:

$$cost(h(x), y) = -y \cdot log(h(x)) - (1-y)log(1-h(x))$$

Ошибка для всех данных датасета:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^i log(h(x^i)) + (1-y^i)log(1-h(x^i))\right]$$

Где m – количество элеметов.

https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24

# Градиентный спуск.

Будем искать минимум функции относительно параметров:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h(x^i) - y^i) x_j^i$$

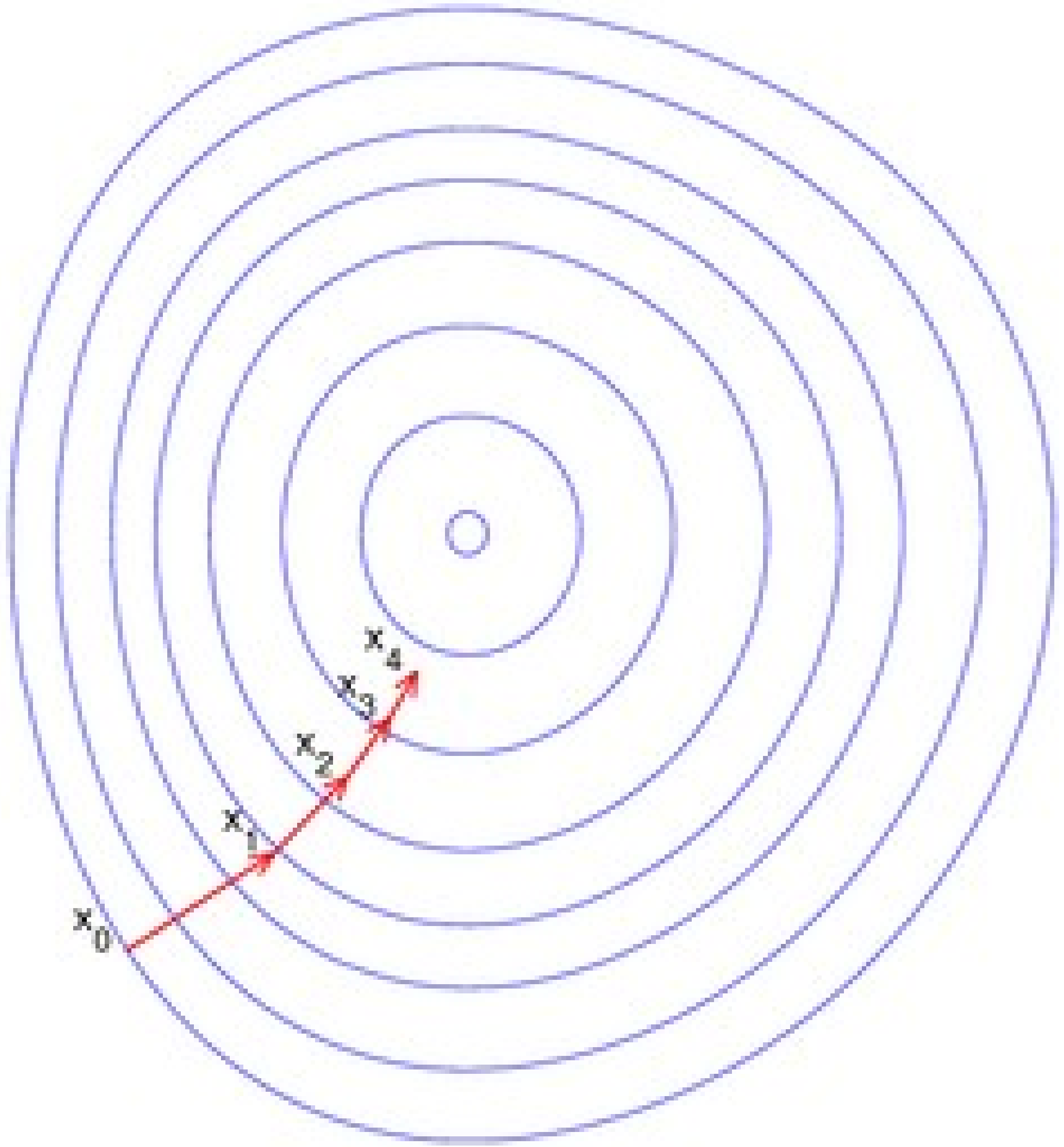
Для поиска минимума используется градиентный спуск.

Это метод нахождения локального минимума (максимума) функции с помощью движения вдоль градиента.

https://ru.wikipedia.org/wiki/Градиентный_спуск

# Градиентный спуск.

Основная идея – идти в направлении наискорейшего спуска, оно задается антиградиентом.



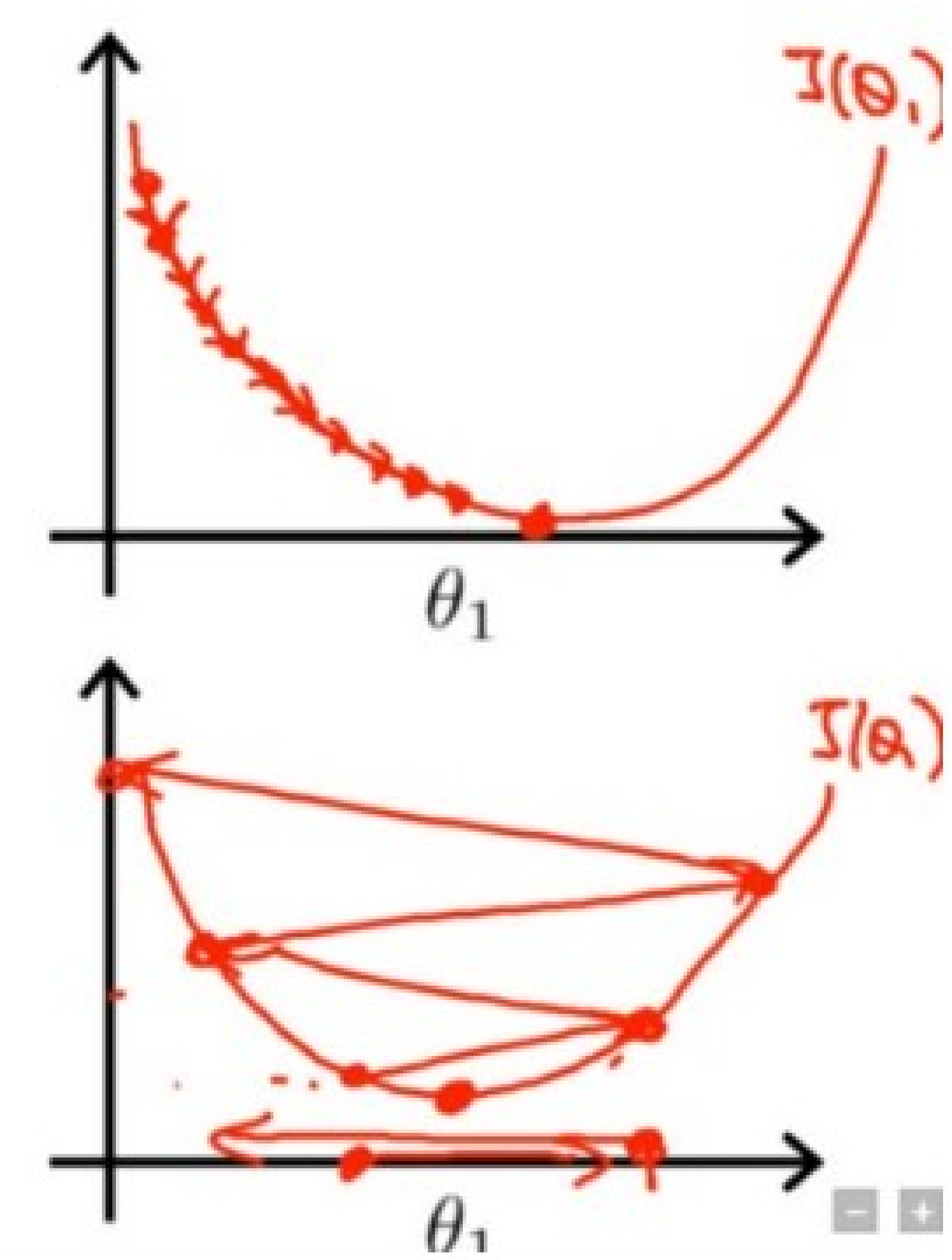

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]})$$

Где $\lambda^{[j]}$- скорость градиентного спуска,
F – целевая функция

https://ru.wikipedia.org/wiki/Градиентный_спуск

# Скорость градиентного спуска

Проблемы градиентного спуска.



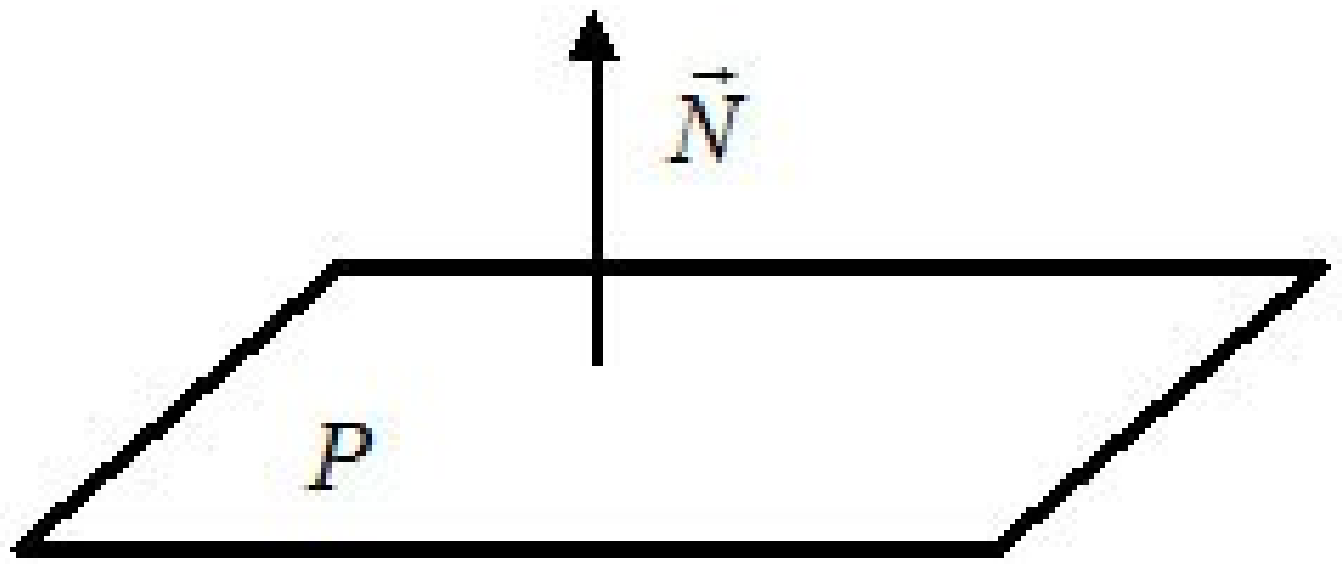

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \bigtriangledown F(\vec{x}^{[j]})$$

Где $\lambda^{[j]}$- скорость градиентного спуска,
F – целевая функция

Source: Andrew Ng’s course on Coursera

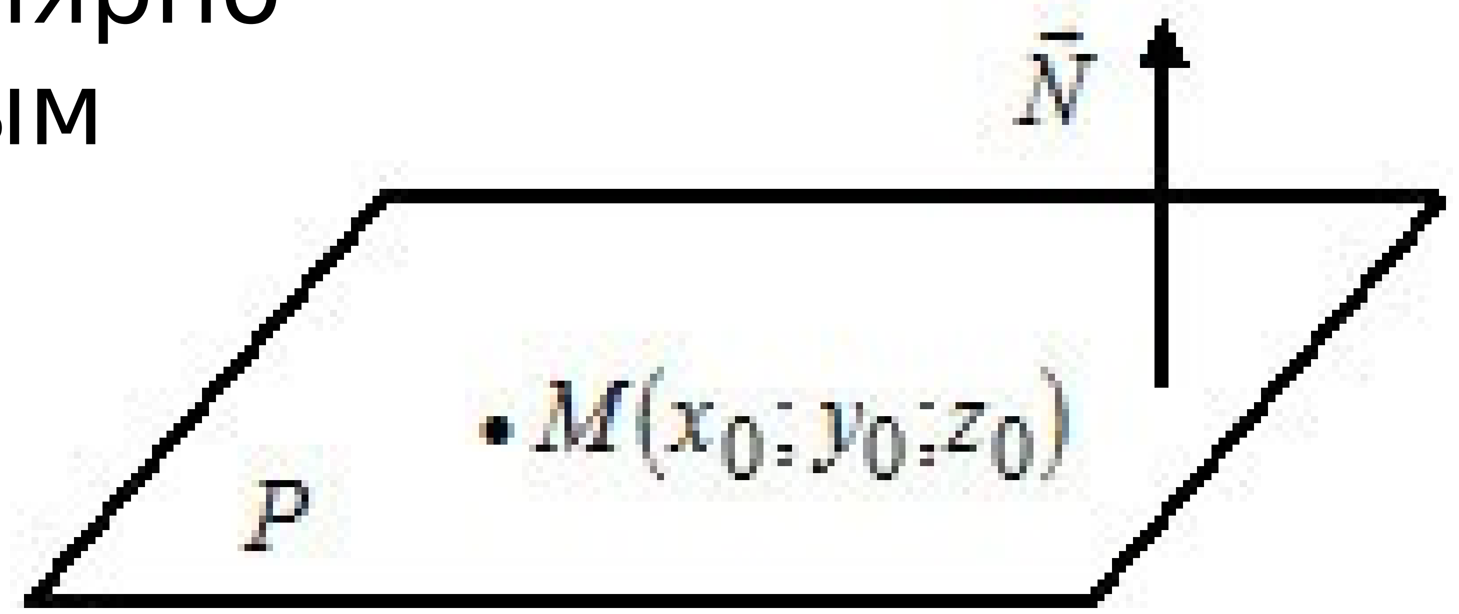
# Уравнение плоскости в пространстве.

Плоскость в пространстве задается следующими способами:

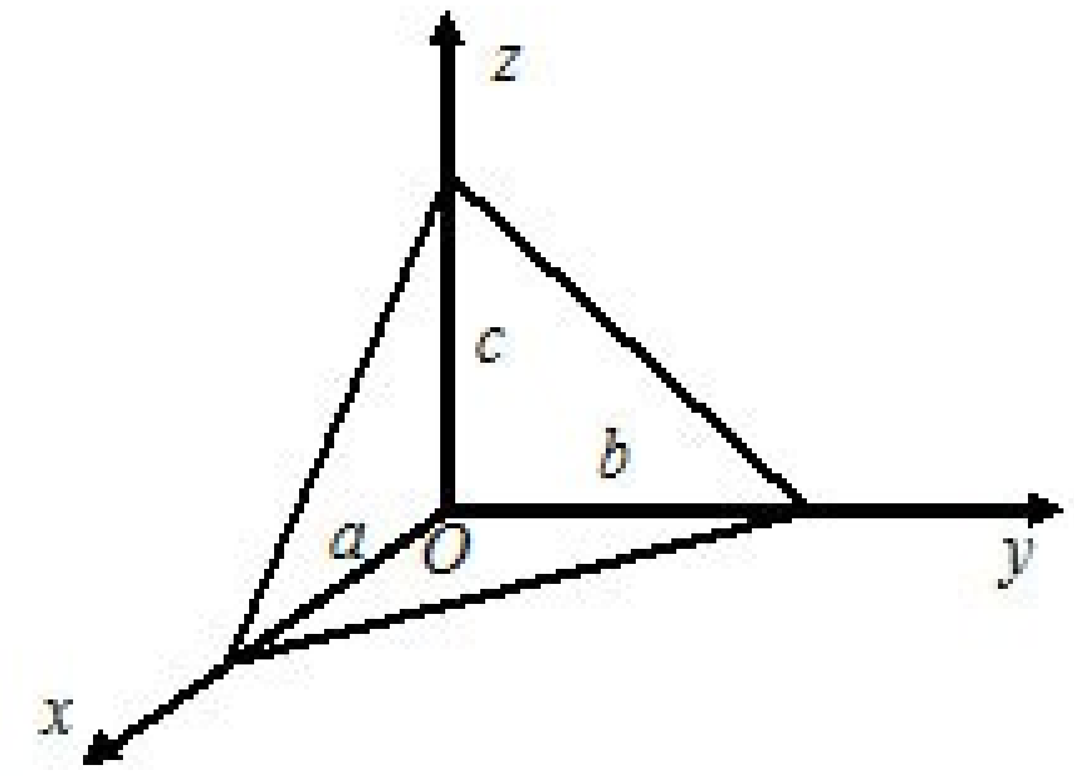



$Ax+By+Cz+D=0-$ общее уравнение плоскости P, где **N**=(A,B,C)− нормальный вектор плоскости P.

$A(x-x0)+B(y-y0)+C(z-z0)=0$ - уравнение плоскости P, которая проходит через точку M(x0,y0,z0) перпендикулярно вектору **N**=(A,B,C). Вектор **N** называется нормальным вектором плоскости.

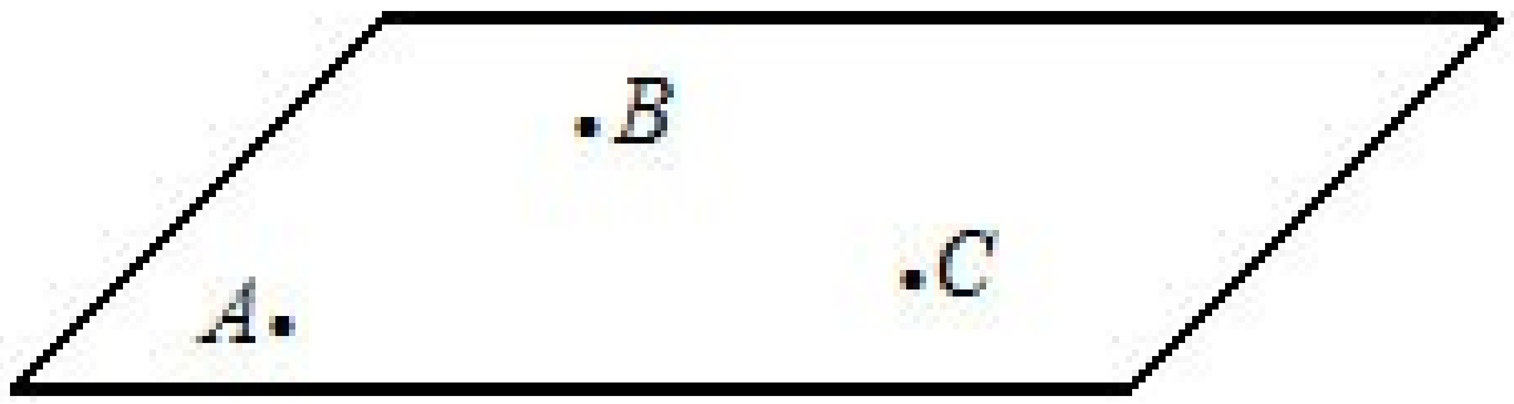


# Уравнение плоскости в пространстве.




$$\frac{x}{a} + \frac{y}{b} + \frac{z}{c} = 1$$

− уравнение плоскости в отрезках на осях, где a, b и c− величины отрезков, которые плоскость отсекает на осях координат.




$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & x_2 - x_1 & x_3 - x_1 \end{vmatrix}$$

уравнение плоскости, которая проходит через три точки A(x1,y1,z1),B(x2,y2,z2) и C(x3,y3,z3).

# Уравнение плоскости в пространстве.

$$x \cdot cos(\alpha) + y \cdot cos(\beta) + z \cdot cos(\gamma) - p = 0$$

нормальное уравнение плоскости, где cos(α),cos(β) и cos(γ) − направляющие косинусы нормального вектора **N**, направленного из начала координат в сторону плоскости, а p>0− расстояние от начала координат до плоскости.

Расстояние от точки M(x0,y0,z0) до плоскости P:Ax+By+Cz+D=0 вычисляется по формуле

$$d = \left| \frac{Ax_0 + By_0 + Cz_0 + D}{\sqrt{A^2 + B^2 + C^2}} \right|$$

# Спасибо за внимание!

**Юстина Иванова,**
Data scientist
«ОЦРВ»