Полиномиальная и логистическая регрессия

Гончаров Павел kaliostrogoblin3@gmail.com
Нестереня Игорь nesterione@gmail.com

План занятия

- Повторение
- Полиномиальная регрессия
- Переобучение и регуляризация
- Задача классификации/ Логистическая регрессия
- Оценка качества модели (основы)
- Категориальные признаки

Повторение

Основные термины

Регрессия [*h(x)*] – это математическое выражение, отражающее зависимость целевой переменной *y* от независимых переменных *x* при условии статистической значимости.

Функция потерь (Lost function) [*L(y*,y)*]– функция определяющая величину ошибки предсказания и настоящего значения (одни пример)

Целевая функция (Cost function)[*J(w)*] – агрегированная оценка ошибки на всей обучающей выборке (обычно представляется собой как усреднённое значение функций потерь).

Математическая постановка задачи регрессии – найти такие параметры w гипотезы $h_w(x)$, при которых значение целевой функции минимально.

$$\boldsymbol{J(w) \to min}$$

Линейная регрессия

Гипотеза:

$$h(x) = w_0 + w_1 x_1 + \ldots + w_n x_n = w^T x$$

Функция потерь:

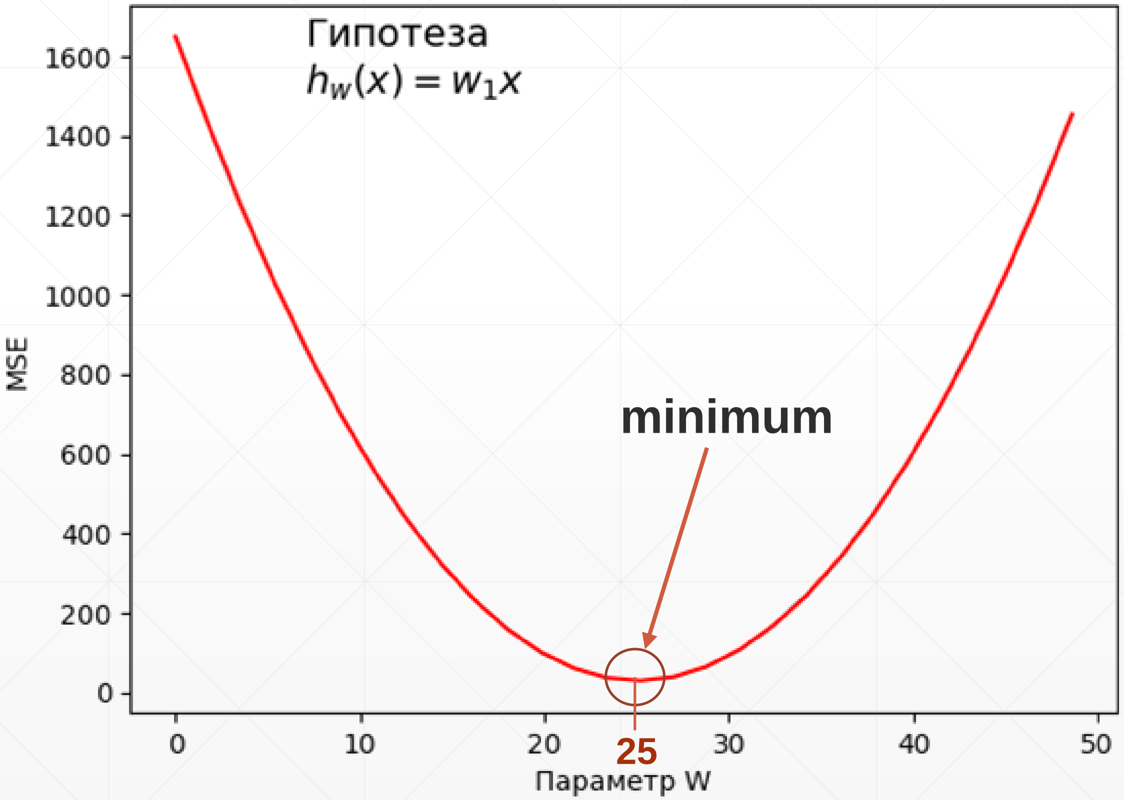
$$L(y*, y) = \frac{1}{2}(y* - y)^2 = \frac{1}{2}(h(x) - y)^2$$

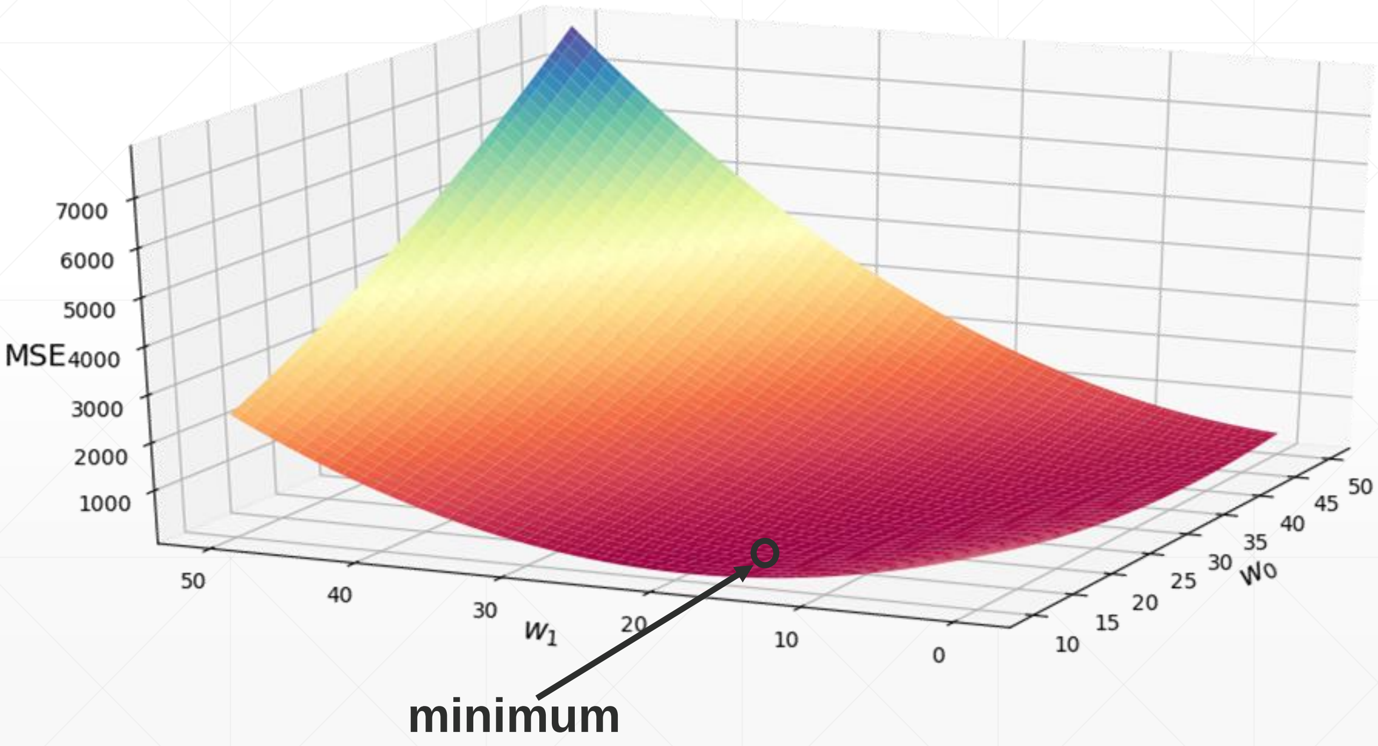
Целевая функция:

$$J(w) = \frac{1}{m}\sum_i \left(h_w(x^{(i)}) - y^{(i)}\right)^2$$

Одномерная и многомерная оптимизация

Задача оптимизации сводится к задаче поиска экстремума (максимума или минимума) функции ошибки (*cost function*). Cost для регрессии – это, как правило, среднеквадратичное отклонение (MSE).





Алгоритм градиентного спуска

Цель: изменять значение параметров w модели $h_w(x)$ «шагая» в направлении к локальному минимуму функции ошибки ($J(w)$). В случае регрессии $J(w)$ = MSE.

Алгоритм:

- задать начальное значение параметров w, например $w_0 = w_1 = \cdots = w_n = 0$
- определить точность ε, например $\varepsilon = 0{,}001$
- задать ***скорость обучения*** α
- повторять до сходимости $J(w)_i - J(w)_{i-1} < \varepsilon$:
  - для всех параметров w найти смещение:

$$temp0 = w_0 - \alpha \frac{\partial}{\partial w_0} J(w_0, w_1, \dots, w_n);$$

$$\dots$$

$$tempN = w_n - \boxed{\alpha \frac{\partial}{\partial w_n} J(w_0, w_1, \dots, w_n)}.$$

  - обновить все веса w_i, где $i = 0{,}1, \dots, n$:

$$w_n = tempN$$

До тех пор, пока ошибка уменьшается больше, чем на epsilon

Частная производная cost function по параметру w_n

Обновление параметров модели

ОСТОРОЖНО МНОГО ФОРМУЛ!

Для того, чтобы обновить параметр, нужно от текущего значения параметра w_i отнять производную функции ошибки $J(w)$ по параметру w_i – $\frac{\partial J(w)}{\partial w_i}$, умноженную на скорость обучения α:

$$w_i = w_i - \alpha \frac{\partial}{\partial w_i} J(w)$$

Например:

$$h_w(x) = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n = w^T x, \text{где } x_0 = 1 \quad (1)$$

1) **Модель линейной регрессии**
2) **Функция ошибки, *cost function***
3) **Частная производная функции ошибки по параметру w_i**

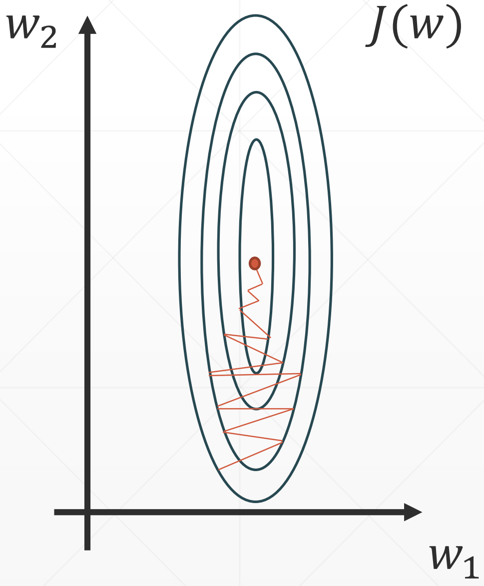
$$J(w) = \frac{1}{2m} \sum_{j=1}^{m} \left(h_w(x^{(j)}) - y^{(j)}\right)^2 \quad (2)$$

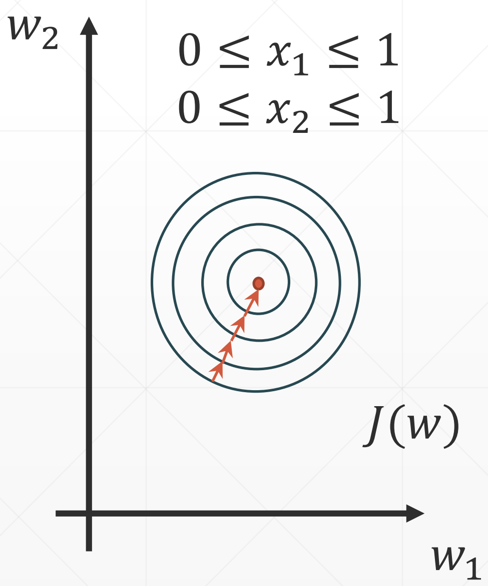
$$\frac{\partial}{\partial w_i} J(w) = \frac{1}{m} \sum_{j=1}^{m} \left(h_w\left(x^{(j)}\right) - y^{(j)}\right) x_i^{(j)} \quad (3)$$

Нормализация и стандартизация данных

Как правило, каждый признак имеет свой диапазон значений. Например, если мы говорим об оценке стоимости жилья по каким-то критериям, то параметр «число комнат» будет иметь значение от 1-10, а параметр «размер жилой площади» может измеряться сотнями квадратных метров. Важно привести все параметры к виду:

$$-1 \le x \le 1$$





1) MinMax масштабирование:

$$\frac{x - x_{min}}{x_{max} - x_{min}} \Rightarrow 0 \le x \le 1$$

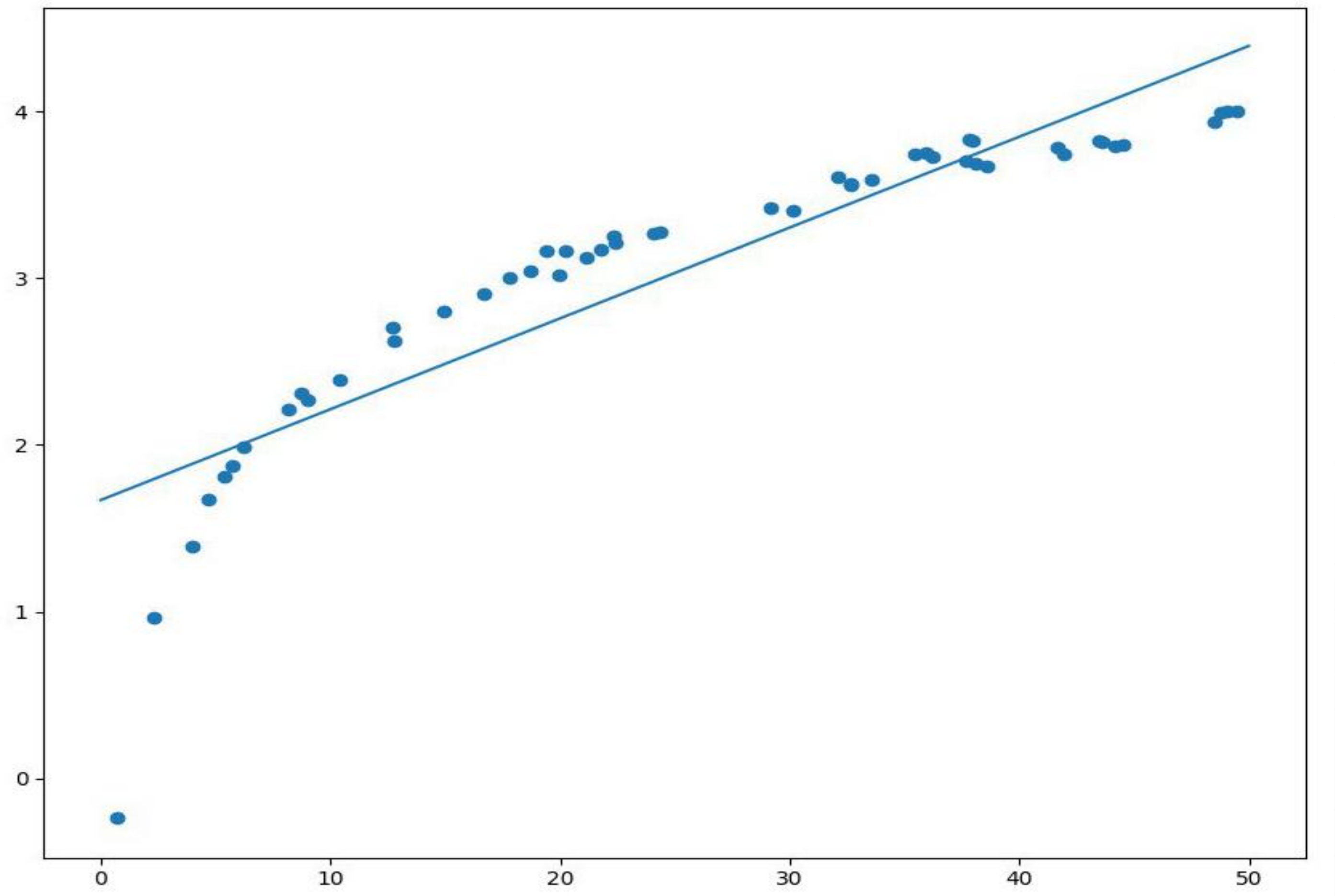
2) Z-score стандартизация:

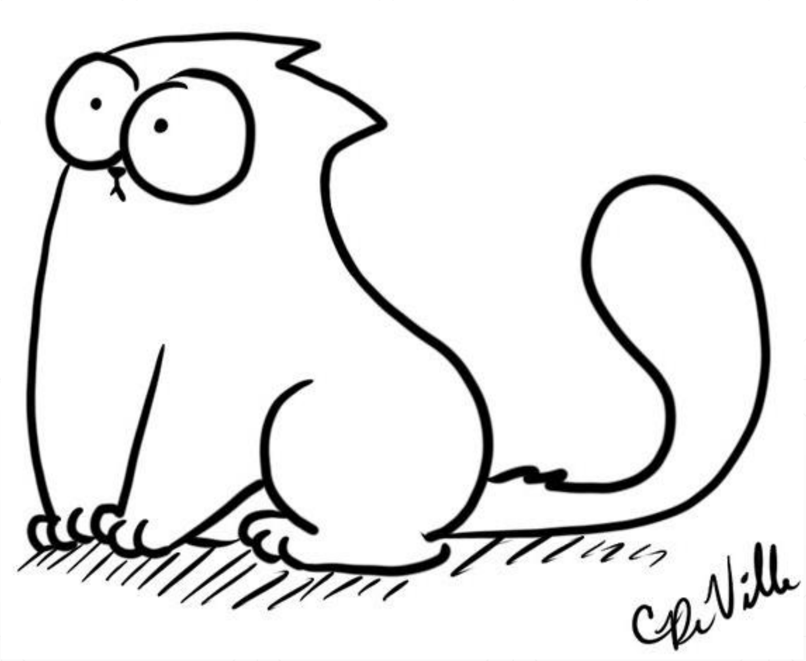
$$\frac{x - \mu}{\sigma} \Rightarrow -1 \le x \le 1$$

$$\mu - \text{среднее}, \sigma = x_{max} - x_{min}$$

Полиномиальная регрессия

- Что делать если признаки зависят нелинейно?





Исходные признаки можно дополнять

- Например для предыдущего пример можно добавить ещё один признак
- $h_w(x) = w_0x_0 + w_1x_1 + w_1x_1^2$

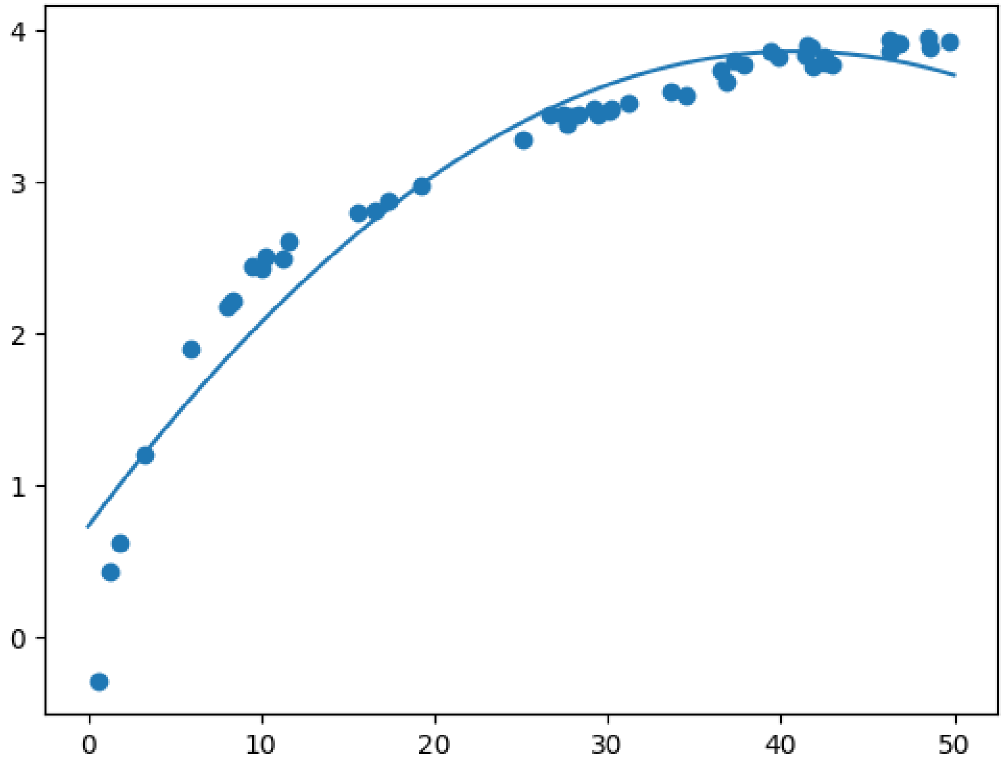
  или

- $h_w(x) = w_0x_0 + w_1x_1 + w_1x_1^2 + w_1x_1^3$

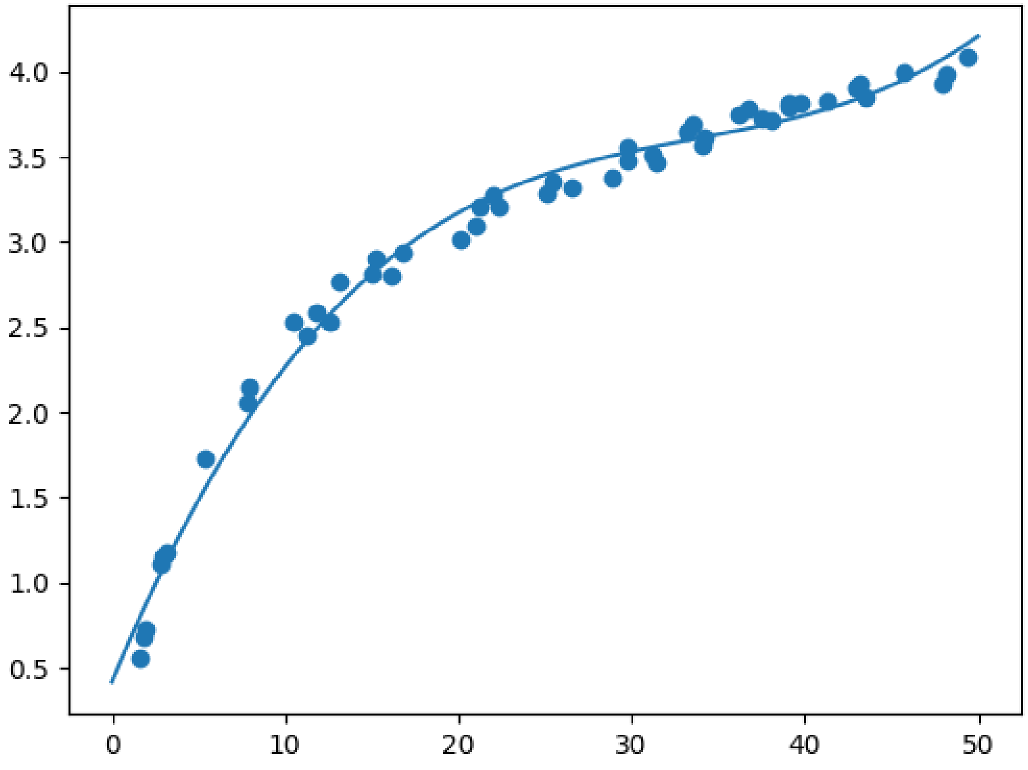
  или

- $h_w(x) = w_0x_0 + w_1x_1 + w_1\sqrt{x_1}$

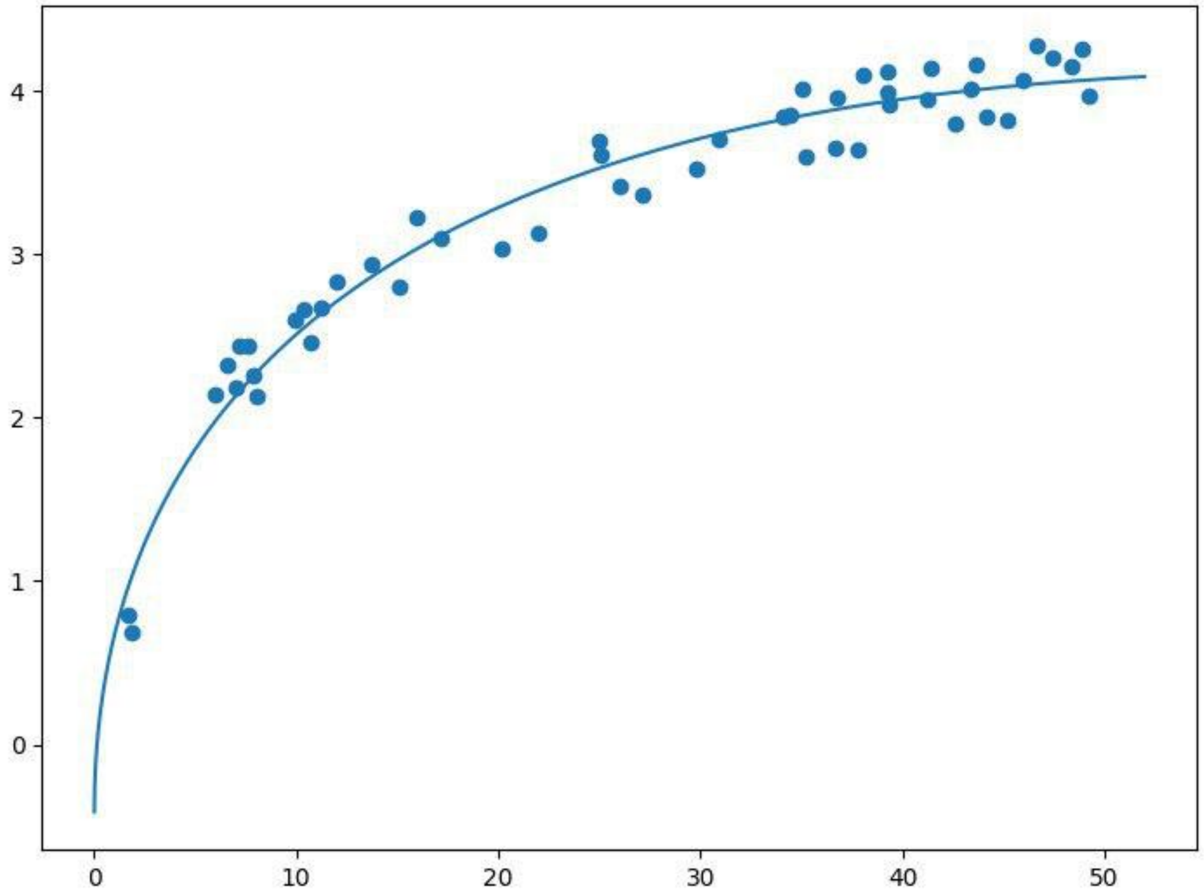
Алгоритм регрессии останется неизменным, вводится только дополнительная обработка параметров.



$$h_w(x) = w_0x_0 + w_1x_1 + w_1x_1^2$$

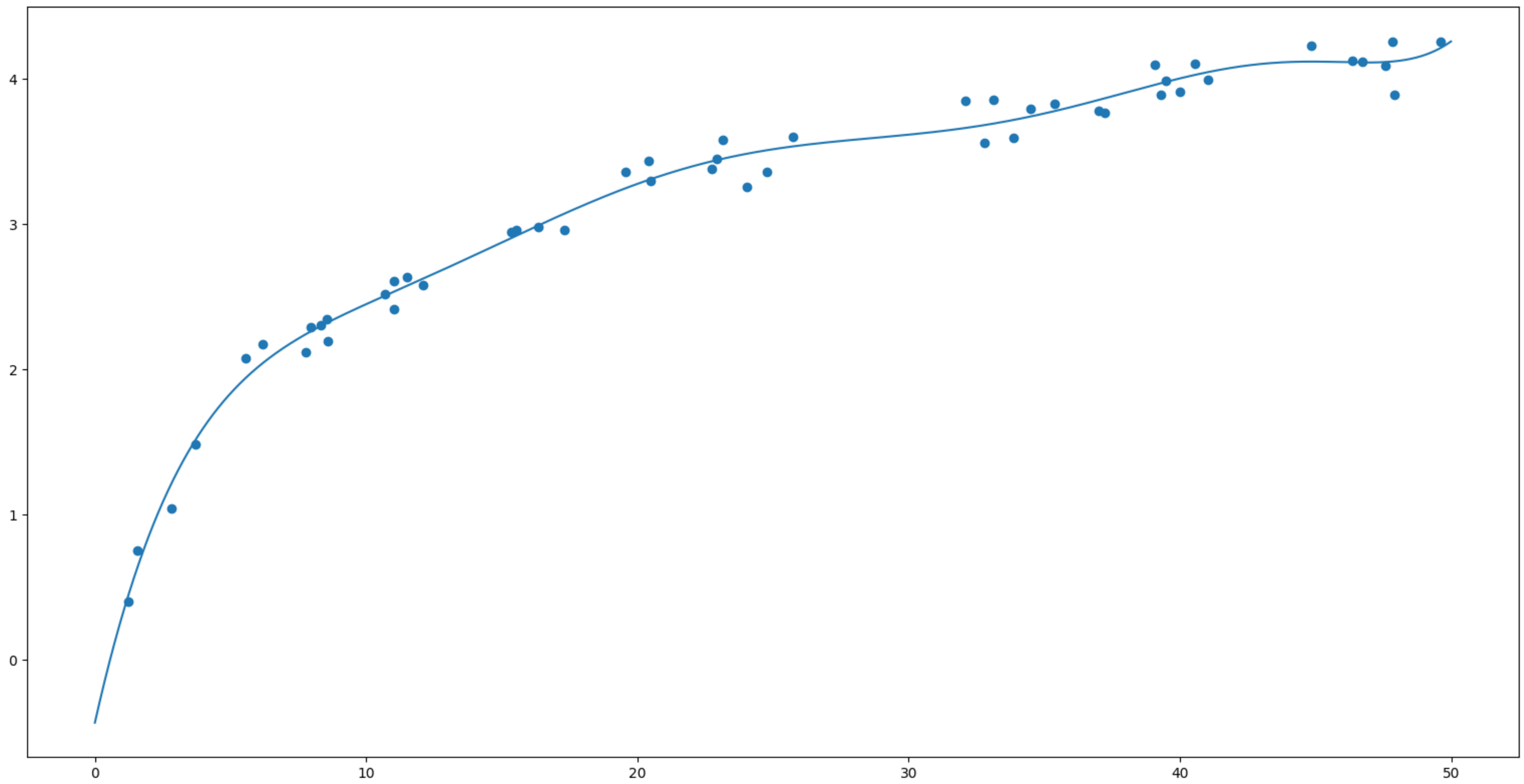


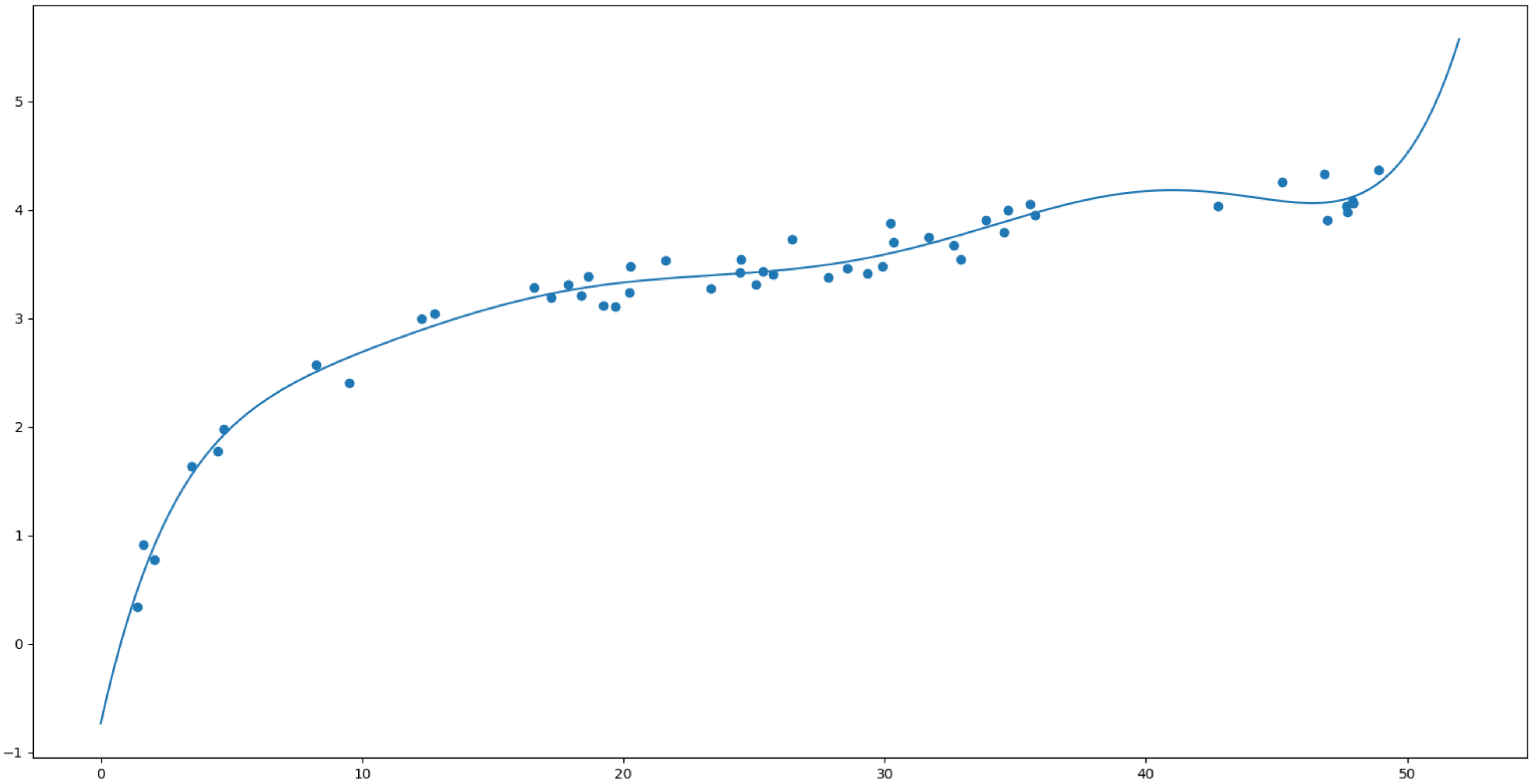
$$h_w(x) = w_0x_0 + w_1x_1 + w_1x_1^2 + w_1x_1^3$$



$$h_w(x) = w_0x_0 + w_1x_1 + w_1\sqrt{x_1}$$

Переобучение





5
4
3
2
1
0
−1
0
10
20
30
40
50

Регуляризация

$$J(\boldsymbol{w}) = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{m}}\sum_{\boldsymbol{i}}\left(\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}^{(\boldsymbol{i})}) - \boldsymbol{y}^{(\boldsymbol{i})}\right)^{\mathbf{2}} - C\sum \boldsymbol{w}_{\boldsymbol{j}}{}^{\mathbf{2}}$$

C – параметр регуляризации

$$\frac{\partial}{\partial w_i}J(w) = \frac{1}{m}\sum_{j=1}^{m}\left(h_w\left(x^{(j)}\right) - y^{(j)}\right)x_i^{(j)} + \frac{C}{m}w_i$$

$$w_i = w_i\left(1 - \alpha\frac{C}{m}\right) - \alpha\frac{\partial}{\partial w_i}J(w)$$

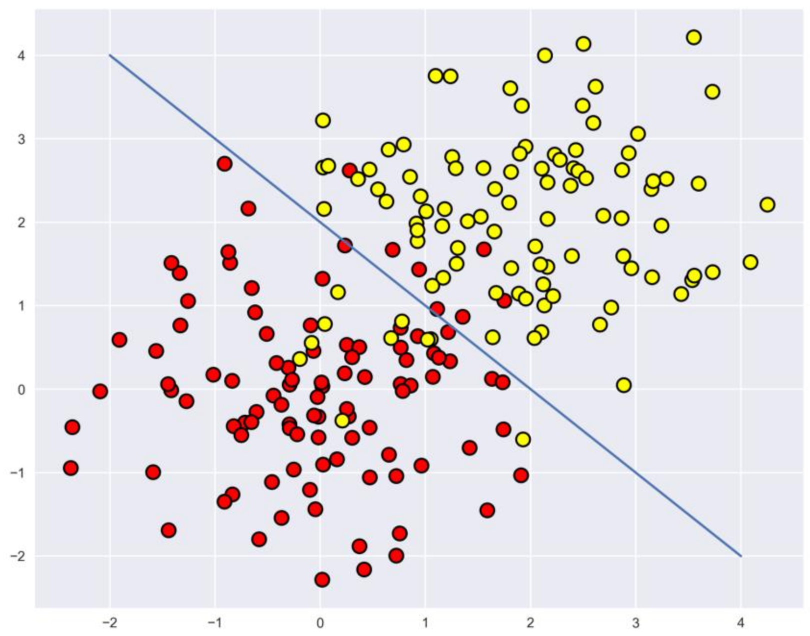
L2 = $C\sum \boldsymbol{w}_{\boldsymbol{j}}{}^{\mathbf{2}}$

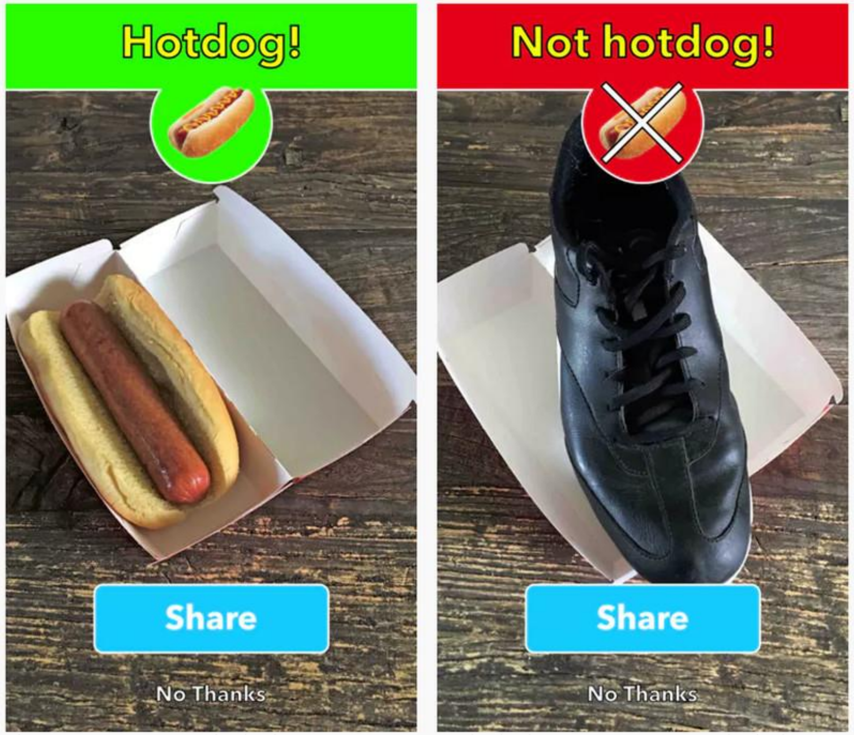
L1 = $C\sum |\boldsymbol{w}_{\boldsymbol{j}}|$

http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/

https://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization-How-does-it-solve-the-problem-of-overfitting-Which-regularizer-to-use-and-when

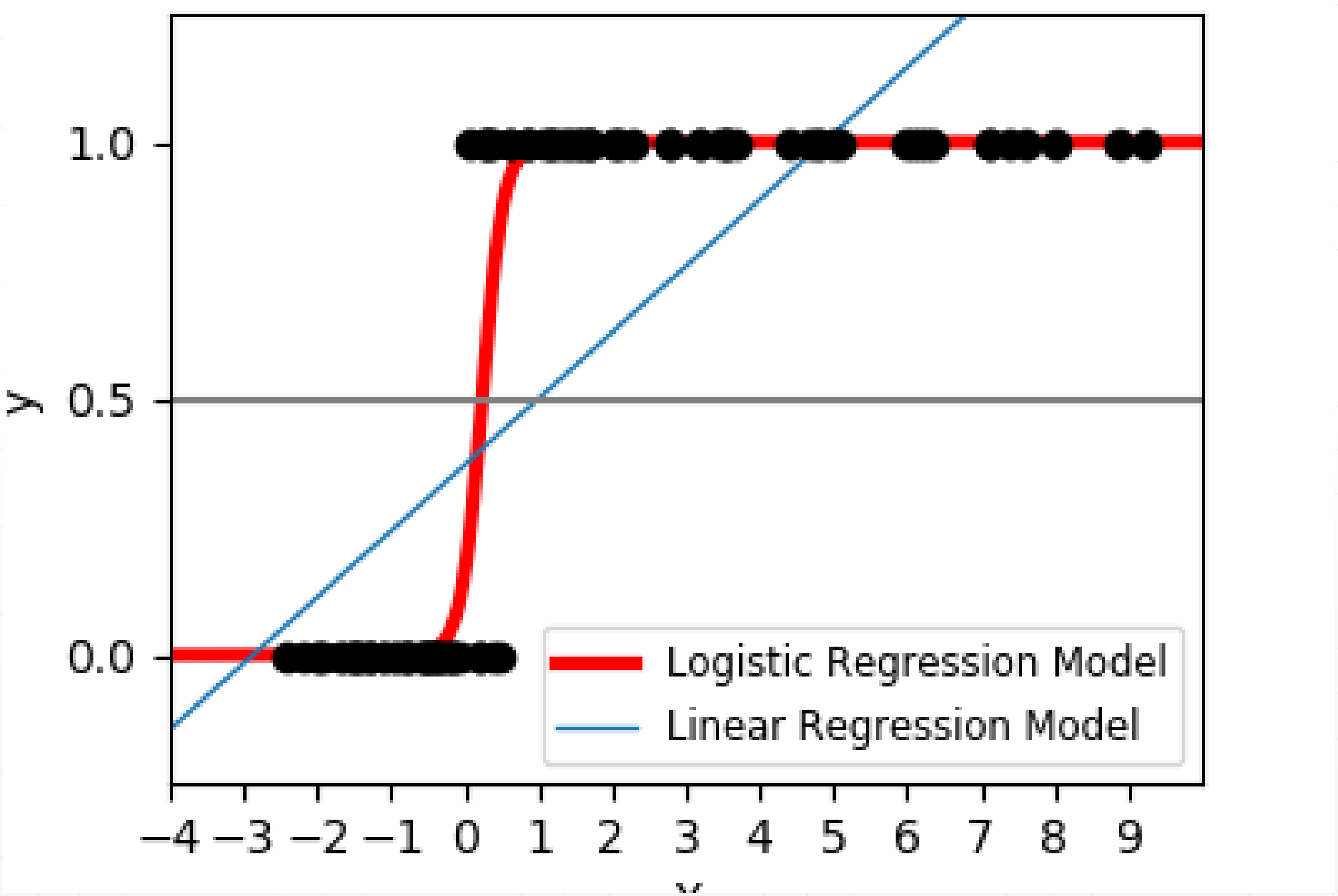
Задача классификации





Алгоритм регрессии останется неизменным, вводится только дополнительная обработка параметров.

Линейной регрессии недостаточно!



Логистическая регрессия

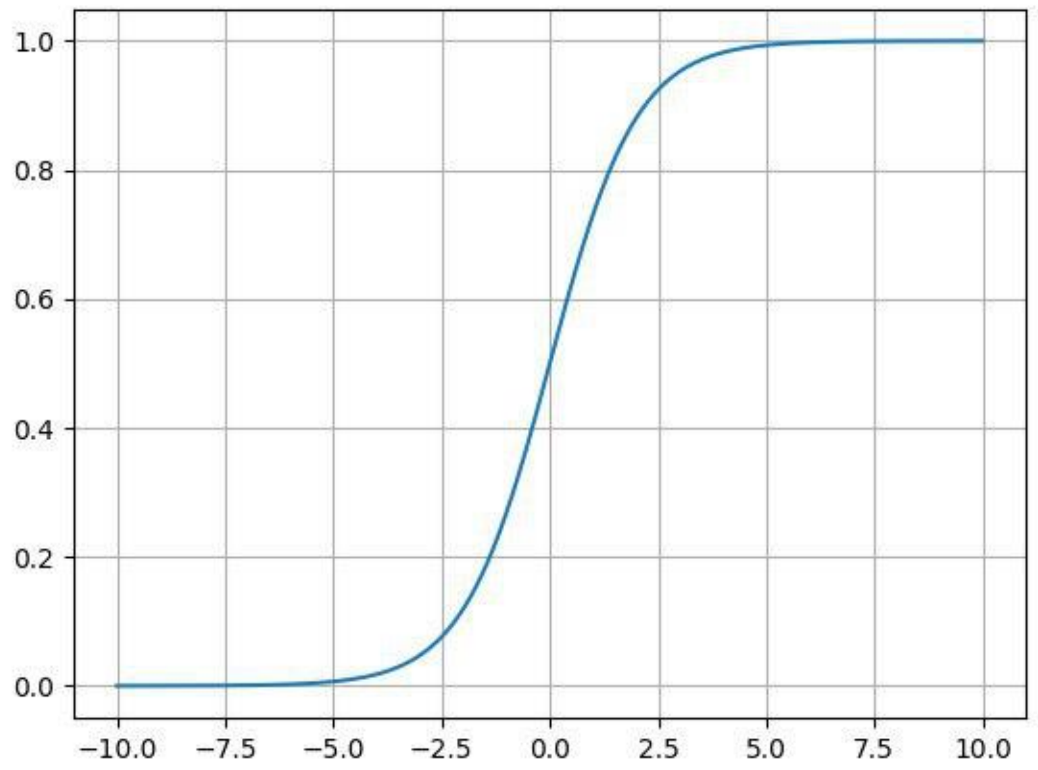
y* = P(y=1|x)

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

$$h_w(x) = a = \sigma(w_0 x_0 + w_1 x_1 + \cdots + w_n x_n) = \sigma(w^T x)$$

L*(a,y)*= – (y * log(a) + (1 – y)log(1 – a))

$$\mathrm{J}(\boldsymbol{w}) = \frac{\boldsymbol{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i}} \boldsymbol{L}(h_w(x), \boldsymbol{y})$$



Алгоритм градиентного спуска

Цель: изменять значение параметров w модели $h_w(x)$ «шагая» в направлении к локальному минимуму функции ошибки ($J(w)$). В случае логистической регрессии $J(w)$ = **log_loss**.

Алгоритм:

- задать начальное значение параметров w, например $w_0 = w_1 = \cdots = w_n = 0$
- определить точность ε, например $\varepsilon = 0{,}001$
- задать ***скорость обучения*** α
- повторять до сходимости $J(w)_i - J(w)_{i-1} < \varepsilon$:
  - для всех параметров w найти смещение:

$$temp0 = w_0 - \alpha \frac{\partial}{\partial w_0} J(w_0, w_1, \dots, w_n);$$

$$\dots$$

$$tempN = w_n - \boxed{\alpha \frac{\partial}{\partial w_n} J(w_0, w_1, \dots, w_n)}.$$

  - обновить все веса w_i, где $i = 0,1, \dots, n$:

$$w_n = tempN$$

До тех пор, пока ошибка уменьшается больше, чем на epsilon

Частная производная cost function по параметру w_n

Используем градиентный спуск

- https://www.kaggle.com/c/titanic