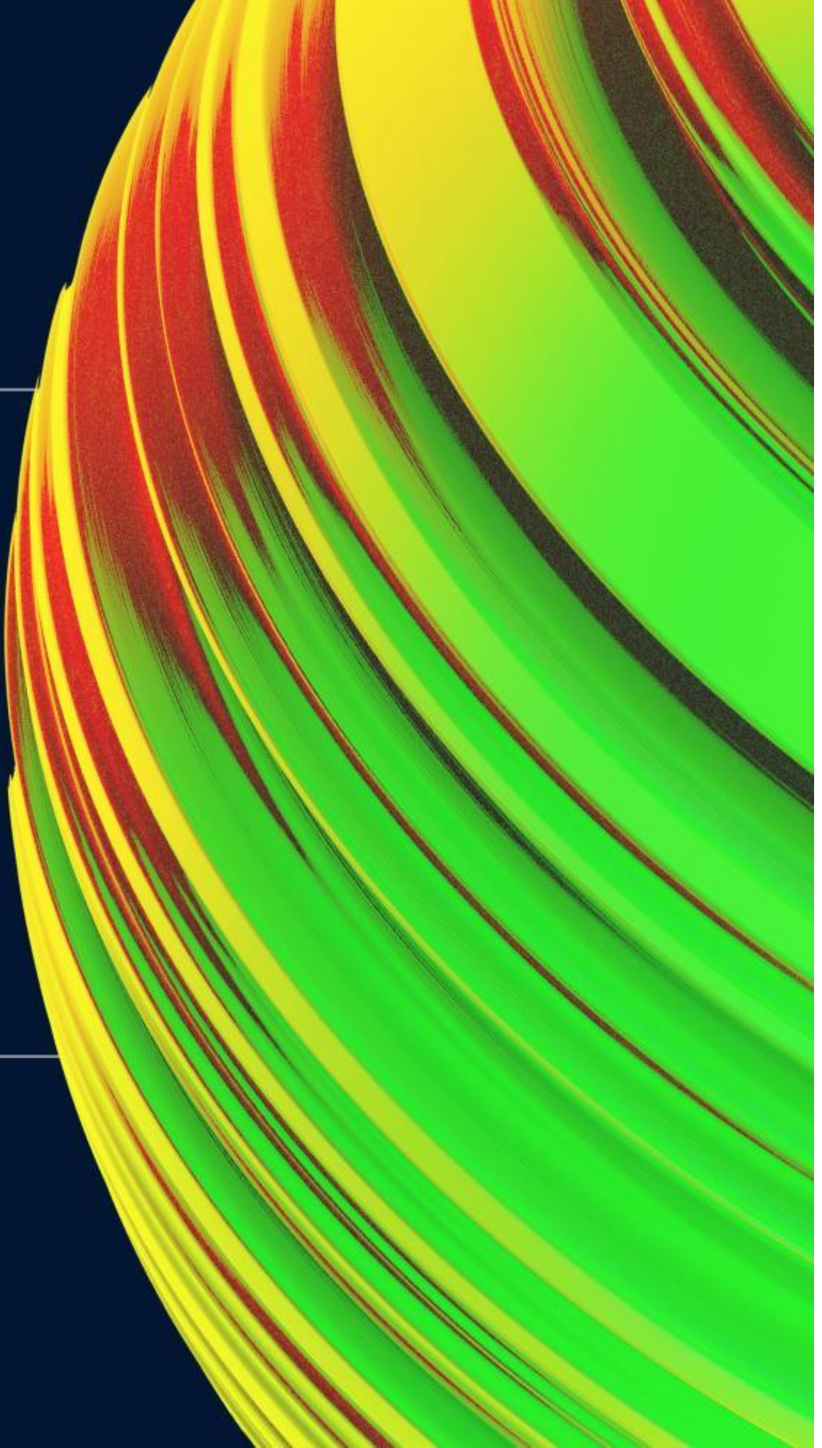
УНИВЕРСИТЕТ
ИННОПОЛИС

Логистическая регрессия. Полиномиальная регрессия

Воробьёва Мария

- maria.vorobyova.ser@gmail.com
- @SparrowMaria



Логистическая регрессия

ID клиента	1	2	3	4	5	6	7	...	n	достиг состояния дефолта
1										1
2										0
3										0
4										0
5										1
6										1
7										1
8										0
9										0
10										0
...										...
k										1

У нас есть задача:

на основании исторических данных построить модель,

которая разделяет клиентов, которые **достигли состояние дефолта** (1)

и

которые **не достигли состояние дефолта** (0, иногда еще помечают как -1)

Какие данные у нас есть?
Одна строка - заявка на кредит. Столбцы - это все возможные признаки.
Зеленый столбец - целевая переменная

Логистическая регрессия

Еще одна задача: есть множество пользователей, которым мы показывали нашу рекламу. По каждому пользователю известно кто установил приложение (1), кто нет(0/-1). Мы хотим построить модель, которая предскажет нам пользователя, который установит наше приложение

ID пользователя	1	2	3	4	5	6	7	...	n	установил наше приложение?
1										1
2										-1
3										1
4										-1
5										1
6										1
7										1
8										-1
9										-1
10										-1
...										...
k										1

Логистическая регрессия

Еще одна задача: есть множество пользователей, которым мы показывали нашу рекламу. По каждому пользователю известно кто установил приложение (1), кто нет(0/-1). Мы хотим построить модель, которая предскажет нам пользователя, который установит наше приложение

ID пользователя	1	2	3	4	5	6	7	...	n	установил наше приложение?
1										1
2										-1
3										1
4										-1
5										1
6										1
7										1
8										-1
9										-1
10										-1
...										...
k										1

Как это можно сделать?

Используем метод обучения с учителем…

Что мы уже знаем?
KNN и линейную регрессию

Линейная регрессия

Линейная регрессия: $a(x) = w_0 + w_1*x_1 + w_2*x_2 + \ldots w_d*x_d$

В компактном виде $$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j.$$

w_i - веса/коэффициенты

w_0 - свободным коэффициентом/сдвиг

можно еще в **более компактном виде** $$a(x) = w_0 + \langle w, x \rangle,$$

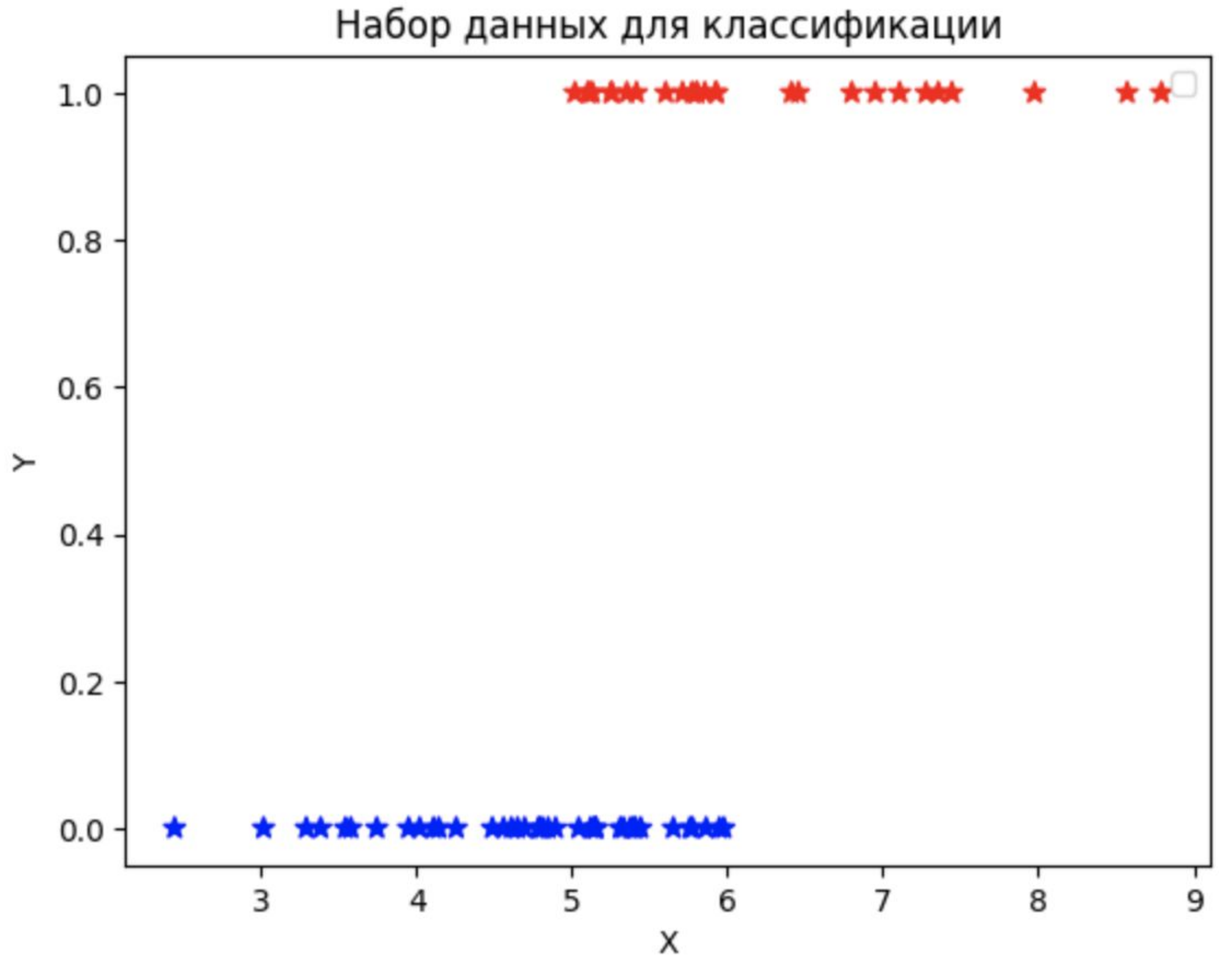
а если предположить, что существует еще один столбец со всеми 1, то можно записать еще **компактнее** $$a(x) = \langle w, x \rangle.$$

http://wiki.cs.hse.ru/Машинное_обучение_1

Логистическая регрессия

Рассмотрим случай с 1 независимым признаком X и бинарной целевой переменной y, которая принимает значения 0 или 1

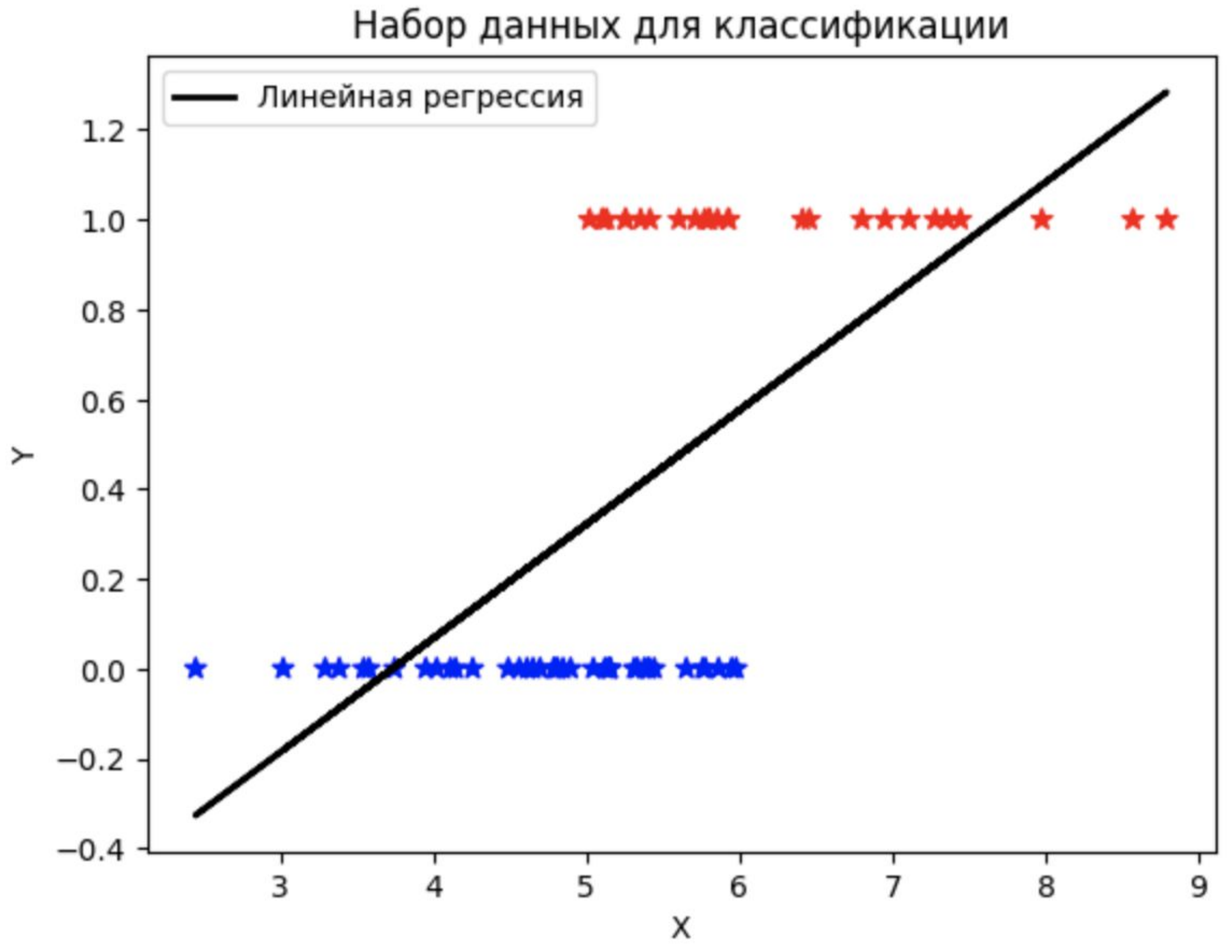
	x	y
0	5.400157	0.0
1	5.978738	0.0
2	4.022722	0.0
3	5.950088	0.0
4	4.848643	0.0
...	...	...
61	6.413146	1.0
62	5.021000	1.0
63	8.571741	1.0
64	5.253824	1.0
65	5.803979	1.0



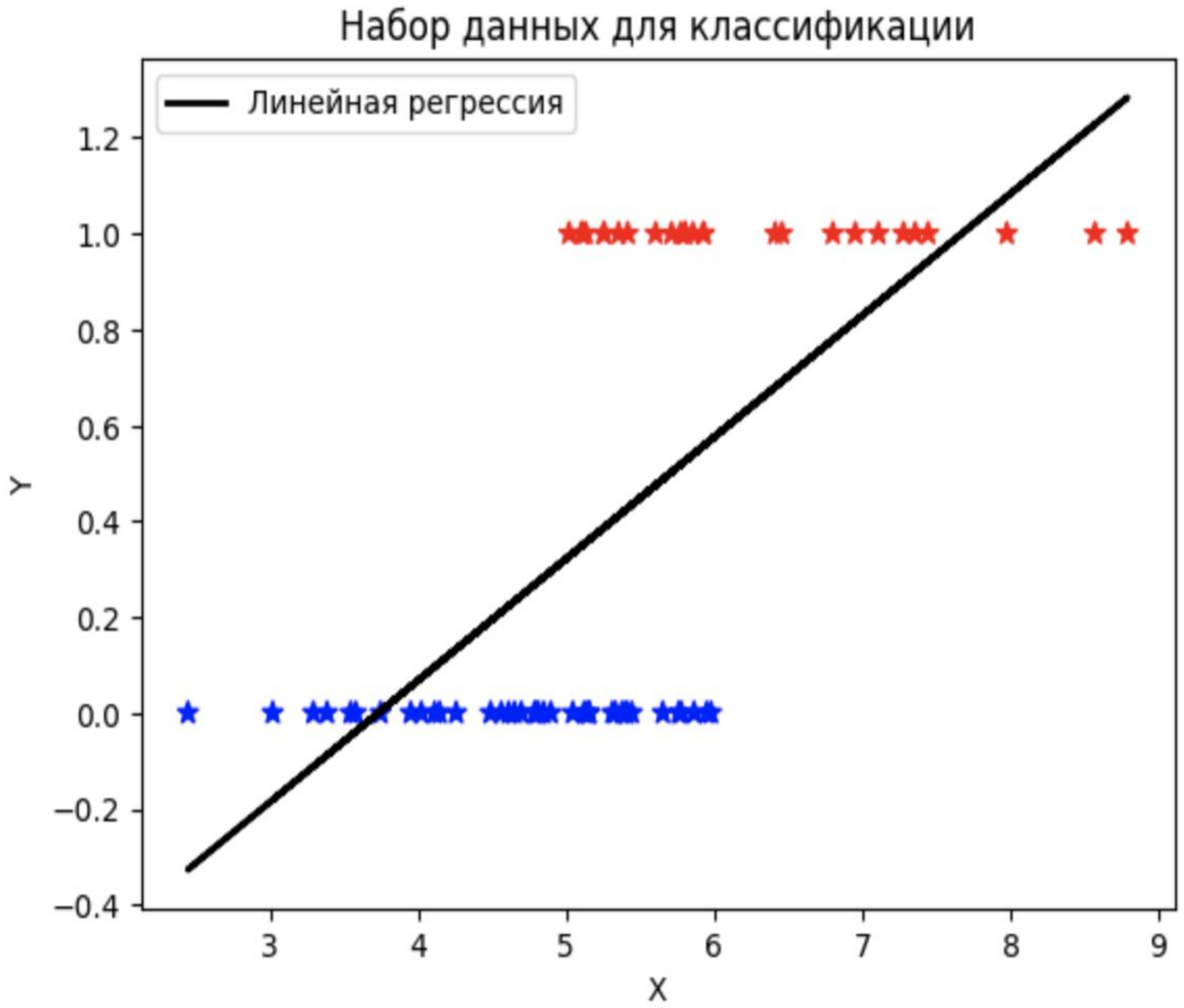
Чем же линейная регрессия плохо?

Построим линейную регрессию с loss функцией MSE

	x	y
0	5.400157	0.0
1	5.978738	0.0
2	4.022722	0.0
3	5.950088	0.0
4	4.848643	0.0
...	...	...
61	6.413146	1.0
62	5.021000	1.0
63	8.571741	1.0
64	5.253824	1.0
65	5.803979	1.0



Чем же линейная регрессия плохо?



Линейная регрессия будет стремиться предсказывать значения внутри непрерывного диапазона, что может привести к неправильным результатам.

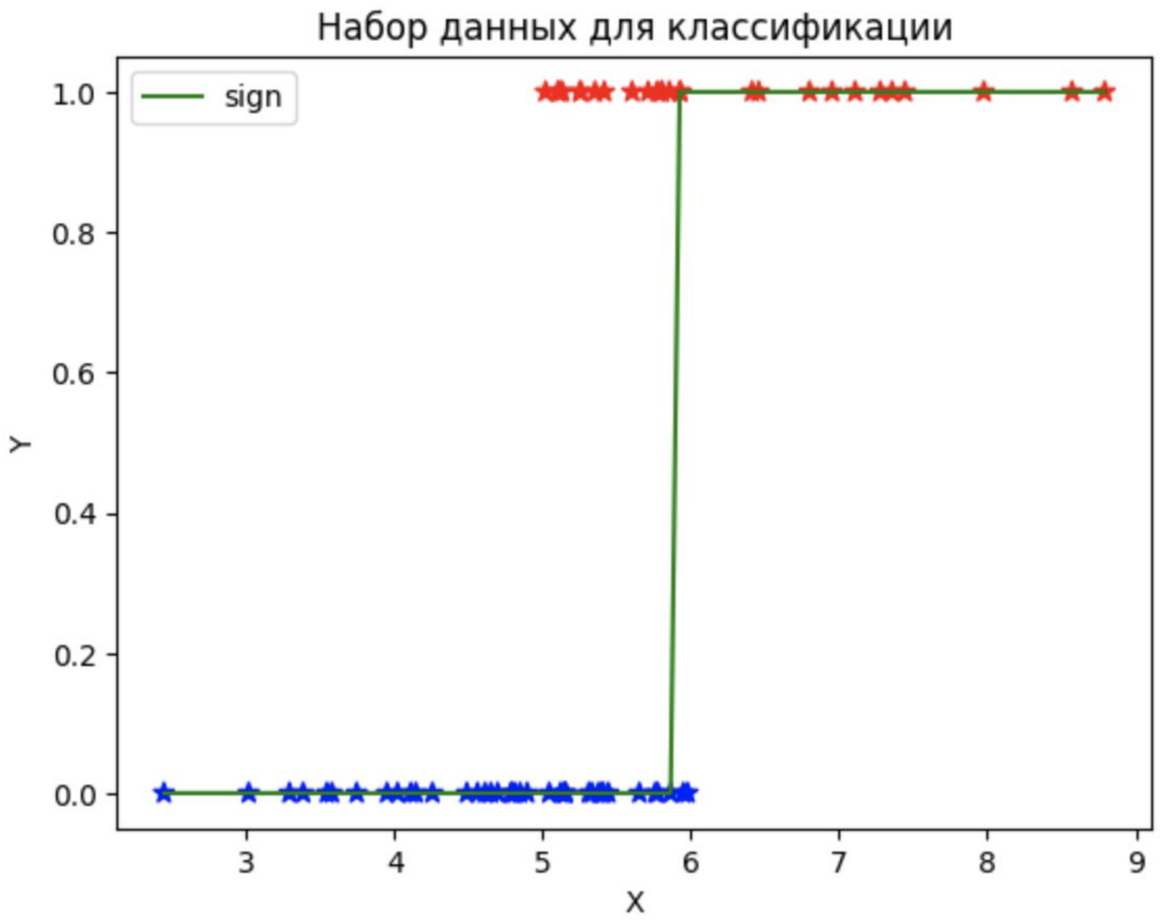
Например, предсказания модели могут оказаться за пределами диапазона [0, 1], что не соответствует бинарной классификации.

Во-первых, регрессия почти не штрафует за ошибки на объектах, которые лежат близко к *разделяющей плоскости*, но не с той стороны.

Во вторых, ошибкой будет считаться предсказание, например, 5 вместо 1, хотя нам-то на самом деле не важно, какой у числа модуль, лишь бы знак был правильным

https://academy.yandex.ru/handbook/ml/article/linear-models

Чем же линейная регрессия плохо?



Иногда используют $a(x) = \text{sign}\,\langle w, x\rangle$

Если sign (знак скалярного произведения больше 0, то 1, если меньше 0, то 0)

Плюсы:

1) ориентируемся только на модуль, поэтому исчезла ситуация, когда ошибкой считается предсказание, например, 5 вместо 1
2) простая функция

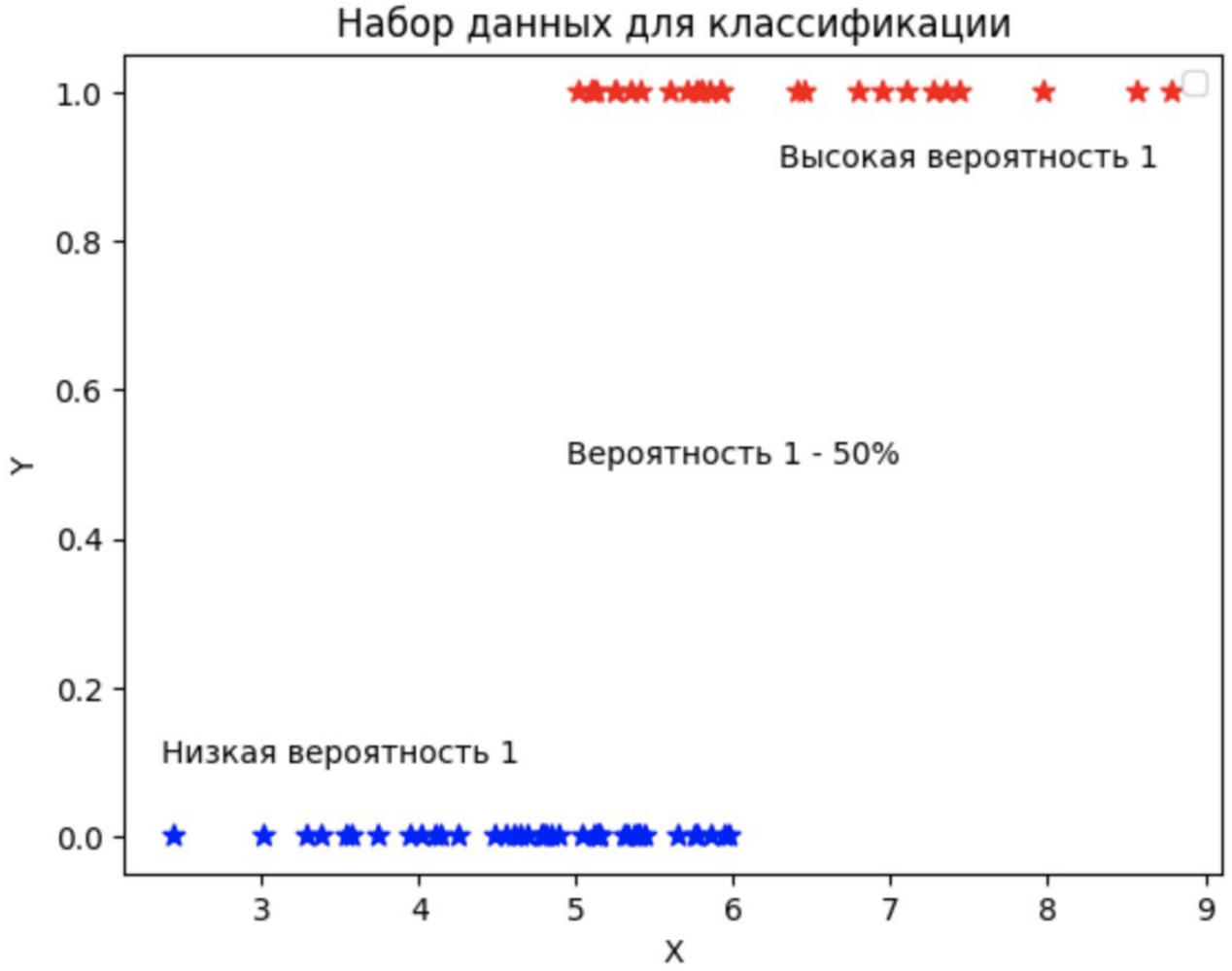
Минусы:

1) sign является разрывной и не дифференцируемой в точках, где значение равно нулю. Это может быть проблематичным при использовании методов оптимизации, которые требуют градиентов функции.

2) нет вероятностной интерпретации, есть только два решения 0 или 1

https://academy.yandex.ru/handbook/ml/article/linear-models

Чем же линейная регрессия плохо?



вероятность, по определению, величина от 0 до 1,

простого способа обучить линейную модель так, чтобы это ограничение соблюдалось, нет.

Из этой ситуации можно выйти так: научить линейную модель правильно предсказывать какой-то объект, связанный с вероятностью, но с диапазоном значений $(-\infty,\infty)$, а дальше преобразовать ответы модели в вероятность!!!

https://academy.yandex.ru/handbook/ml/article/linear-models

Логистическая регрессия

Таким объектом является **logit** или **log odds** – логарифм отношения вероятности положительного события к отрицательному $\log\left(\frac{p}{1-p}\right)$

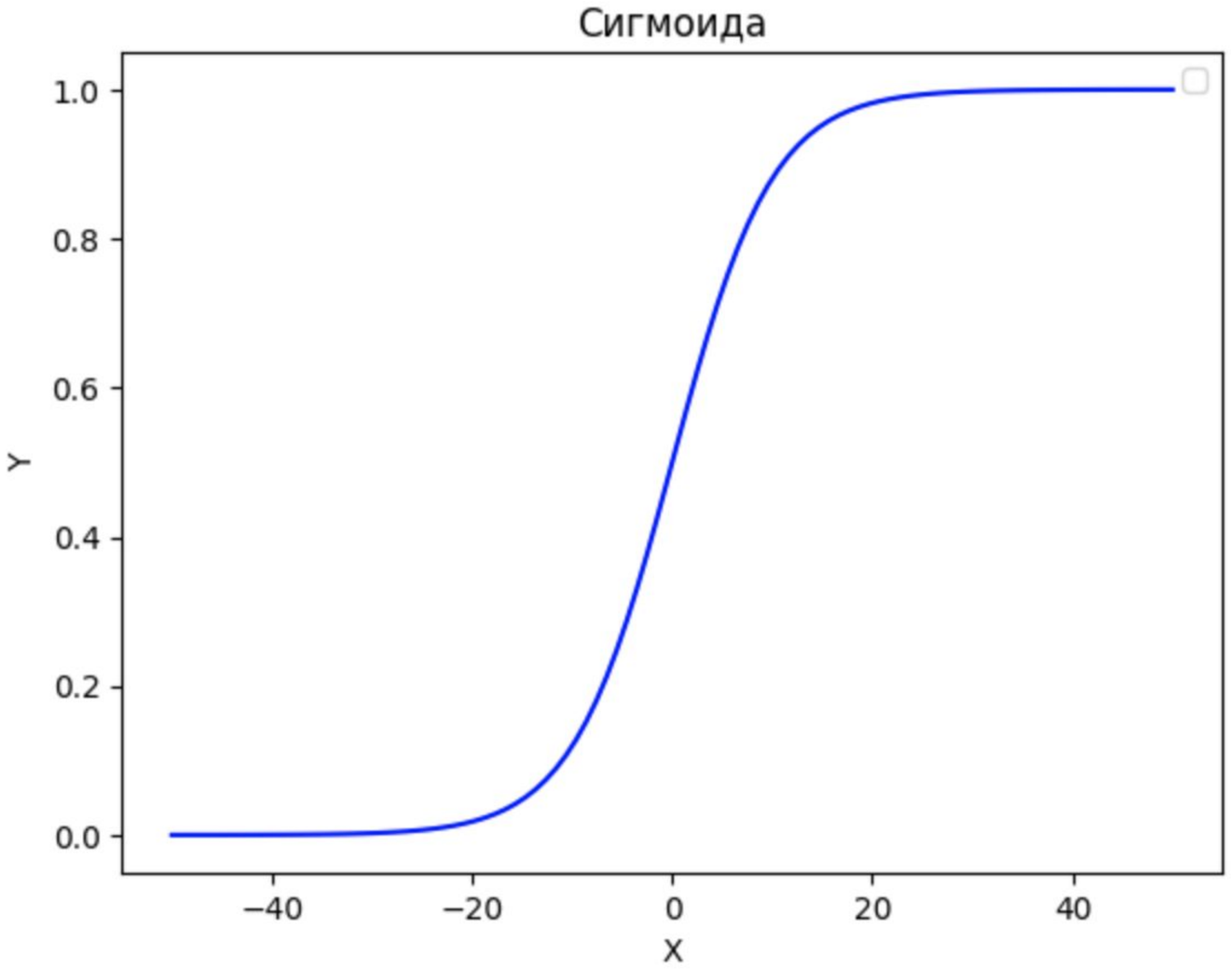
$$\langle w, x_i \rangle = \log\left(\frac{p}{1-p}\right)$$

$$e^{\langle w, x_i \rangle} = \frac{p}{1-p}$$

$$p = \frac{1}{1 + e^{-\langle w, x_i \rangle}}$$

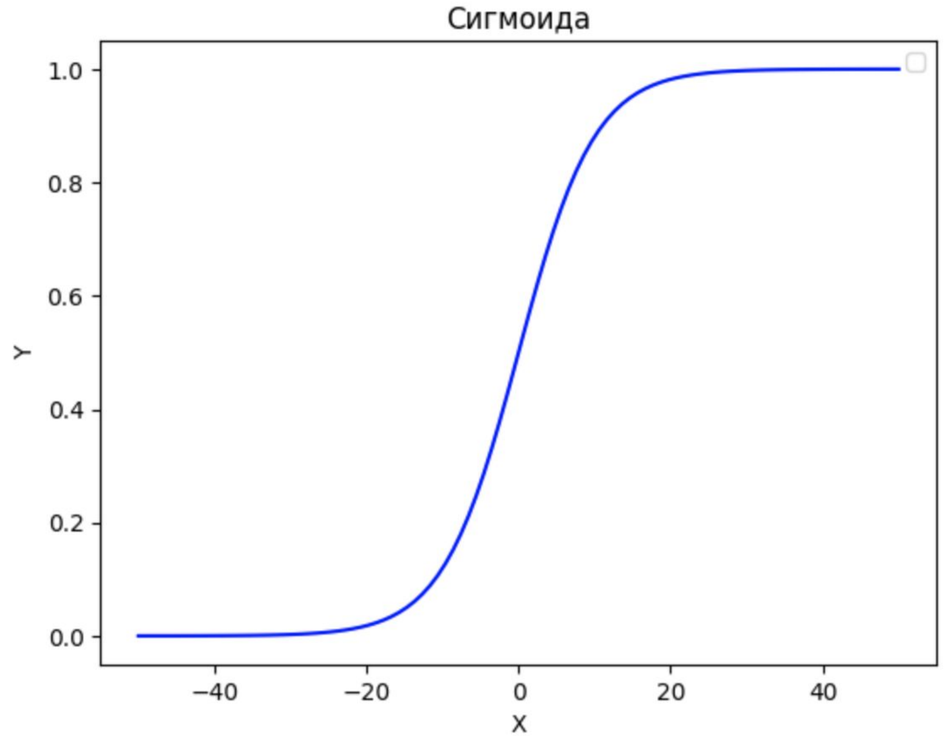
https://academy.yandex.ru/handbook/ml/article/linear-models

Логистическая регрессия



$$\sigma(\langle w, x \rangle) = \frac{1}{1 + \exp(-\langle w, x \rangle)}$$

Логистическая регрессия



$$\sigma(\langle w, x \rangle) = \frac{1}{1 + \exp(-\langle w, x \rangle)}$$

Как теперь научиться оптимизировать w так, чтобы модель как можно лучше предсказывала логиты?

так как у нас 2 события, то применим распределение Бернулли

Распределение Бернулли

Распределение Бернулли - это дискретное распределение с двумя возможными исходами: успех (обычно обозначается как 1) и неудача (обычно обозначается как 0).

Оно моделирует случайную переменную, которая принимает только эти два значения с некоторой вероятностью успеха (p) и вероятностью неудачи (1-p).

Это распределение применяется для моделирования бинарных данных, таких как броски монеты, клики на веб-странице или прогнозирование успеха или неудачи определенного события.

Распределение числа успехов в серии из n испытаний Бернулли называют биномиальным распределением. Вероятность того, что в серии из n испытаний Бернулли произойдет ровно k успехов равна: $P(X = k) = C_n^k \cdot p^k \cdot (1-p)^{n-k}, k = 0, 1, 2, \ldots, n.$

Распределение Бернулли

Распределение Бернулли описывает случайную переменную, которая может принимать одно из двух возможных значений: 1 с вероятностью p и 0 с вероятностью 1−p.

$$P(y_i = 0|\mathbf{x}_i) = 1 - \hat{y}_i = 1 - \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}_i + b)}}$$

$$\hat{y}_i = P(y_i = 1|\mathbf{x}_i) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}_i + b)}}$$

Для оценки параметров воспользуемся максимальным правдоподобием

Максимальное правдоподобие

Максимальное правдоподобие (Maximum Likelihood Estimation, MLE) — это метод статистической оценки параметров модели, при котором выбираются такие значения параметров, которые максимизируют вероятность наблюдения имеющихся данных. Проще говоря, **MLE ищет параметры, при которых наблюдаемые данные наиболее вероятны.**

Пусть $X=(X1,X2,\ldots,Xn)$ — выборка из некоторого распределения, зависящего от неизвестного параметра θ .

Функция плотности или функция вероятности наблюдений будет зависеть от θ: $f(X_i|\theta)$

функции правдоподобия: $$L(\theta|X) = \prod_{i=1}^{n} f(X_i|\theta)$$

Логарифм функции правдоподобия $$\ell(\theta|X) = \log L(\theta|X) = \sum_{i=1}^{n} \log f(X_i|\theta).$$

Оценки параметров: $$\hat{\theta} = \arg\max_{\theta} \ell(\theta|X).$$

Логистическая регрессия

Запишем теперь логарифм функции правдоподобия для логистической регрессии

$$\ell(\mathbf{w}, b) = \sum_{i=1}^{N} [y_i \log P(y_i = 1|\mathbf{x}_i) + (1 - y_i) \log(1 - P(y_i = 1|\mathbf{x}_i))]$$

где $P(y_i = 1|\mathbf{x}_i)$ — вероятность, вычисленная по модели для примера $\mathbf{x}_i$.

Далее будем минимизировать логарифм функции правдоподобия со знаком минус

$$\min_{\mathbf{w},b} -\ell(\mathbf{w}, b)$$

Далее градиентный спуск, используя градиенты:

$$\frac{\partial \ell}{\partial \mathbf{w}} = \sum_{i=1}^{N} (y_i - P(y_i = 1|\mathbf{x}_i))\mathbf{x}_i$$

$$\frac{\partial \ell}{\partial b} = \sum_{i=1}^{N} (y_i - P(y_i = 1|\mathbf{x}_i))$$

Логистическая регрессия

ID пользователя	возраст	установил наше приложение?	прогноз модели
1	20	1	0.8
2	30	-1	0.2
3	40	1	0.7
4	54	-1	0.6
5	20	1	0.4
6	30	1	0.5
7	40	1	0.3
8	54	-1	0.2
9	48	-1	0.34
10	50	-1	0.1
...		...	
k	43	1	0.78

Предсказание модели будет вычисляться по следующей формуле: $p = \sigma(\langle w, x_i \rangle)$

То есть мы будем получать в ответе модели в виде вероятности и это вероятность положительного класса, а как от неё перейти к предсказанию самого класса?

Логистическая регрессия

ID пользователя	возраст	установил наше приложение?	прогноз модели	прогноз модели бинарный
1	20	1	0.8	1
2	30	-1	0.2	0
3	40	1	0.7	1
4	54	-1	0.6	1
5	20	1	0.4	0
6	30	1	0.5	0
7	40	1	0.3	0
8	54	-1	0.2	0
9	48	-1	0.34	0
10	50	-1	0.1	0
...		...		0
k	43	1	0.78	1

Введем некоторый **cutoff (threshold)** a_c, далее вводим правило :

если p>a_c, то 1, иначе 0

cutoff (threshold) подбирается исходя из минимизации нужного параметра:

- бизнес метрики (сколько денег мы потеряем/заработаем, доля одобренных заявок и так далее)
- статистические метрики точности модели (precision/recall , F1 и тд)

Логистическая регрессия. История

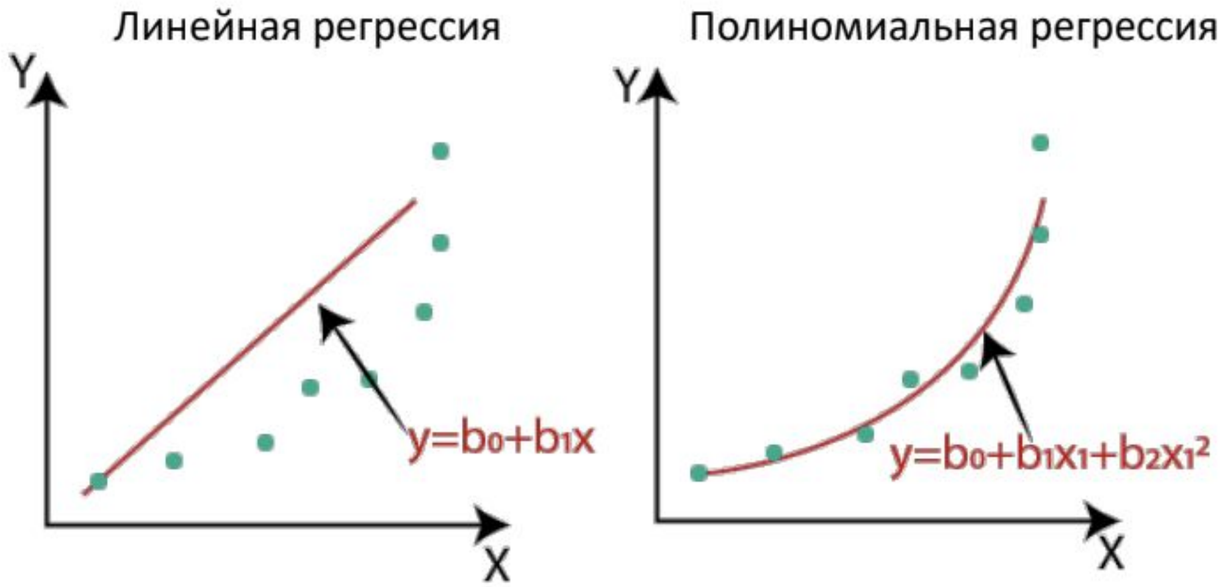
Метод называется **логистической регрессией,** а не логистической классификацией именно потому, что предсказываем мы не классы, а вещественные числа – логиты.

Термин «логистическая» не значит, что кривая связана с поставками товаров. Оно происходит от греческого *λογιστικός, «искусный в расчётах»*.

Логистическое уравнение придумал бельгийский математик Франсуа Ферхюльст, который на основе пяти переписей населения построил кривую изменения численности, предсказавшую значение на 100 лет вперёд с точностью до 1%.

Если название и «мимикрирует» под задачу регрессии, всё-таки это алгоритм классификации.

Полиномиальная регрессия



$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \ldots + \beta_n x_n^n$$

ДЕМО

Многомерный случай

Пусть у нас k классов, как в таком случае строить классификацию?

Существуют два самых популярных способа:

- one-vs-all
- all-vs-all

https://github.com/esokolov/ml-course-hse/blob/master/2020-fall/lecture-notes/lecture06-linclass.pdf

Многомерный случай. One-vs-all

Разделение задачи: Для задачи с k классами создается k бинарных классификаторов. Каждый классификатор обучается отличать один класс (положительный класс) от всех остальных классов (отрицательные классы).

Обучение модели: Для каждого класса i создается набор обучающих данных, где все примеры класса i обозначаются как положительные, а все остальные примеры — как отрицательные. Таким образом, обучается k бинарных моделей.

Предсказание: Во время предсказания для нового примера каждая из k моделей выдает вероятность или оценку принадлежности к своему классу. Класс с наивысшей вероятностью или оценкой выбирается как окончательный предсказанный класс.

Многомерный случай. All-versus-all

Разделение задачи: Для задачи с k классами создается бинарных классификаторов.

C_K^2 классификаторов $a_{ij}(x), i, j = 1, \ldots, K, i \neq j$

Каждый классификатор обучается отличать один класс от одного другого класса.

Обучение модели: Для каждого классификатора используются только примеры двух соответствующих классов, остальные классы игнорируются. Таким образом, каждый классификатор обучается на подмножестве данных.

Предсказание: Во время предсказания для нового примера каждый из классификаторов выдает свое предсказание (голос). Класс, который получает наибольшее количество голосов от всех классификаторов, выбирается как окончательный предсказанный класс.

https://github.com/esokolov/ml-course-hse/blob/master/2020-fall/lecture-notes/lecture06-linclass.pdf

Многомерный случай. Многоклассовая логистическая регрессия

Допустим, что мы теперь решаем многоклассовую задачу и построили линейных моделей

$$b_k(x) = \langle w_k, x \rangle + w_{0k}$$

каждая из которых даёт оценку принадлежности объекта одному из классов. Как преобразовать вектор оценок в вероятности?

Для этого можно воспользоваться оператором softmax, который производит «нормировку» вектора:

$$\text{softmax}(z_1, \ldots, z_K) = \left(\frac{\exp(z_1)}{\sum_{k=1}^{K} \exp(z_k)}, \ldots, \frac{\exp(z_K)}{\sum_{k=1}^{K} \exp(z_k)} \right)$$

https://github.com/esokolov/ml-course-hse/blob/master/2020-fall/lecture-notes/lecture06-linclass.pdf

УНИВЕРСИТЕТ
ИННОПОЛИС

ВОПРОСЫ И ОТВЕТЫ

Основные методы преобразования

Независимые переменные

Категориальные

One hot encoding
Label Encoding
Mean Target encoding
Frequency Encoding

Непрерывные

Standart Scaling(приводим к распределению со средним 0 и среднеквадратическим отклонением 1) (x-ср_знач)/ст.откл

Min Max Scaling(растягиваем в промежуток 0-1) - (x-min)/(max-min)

Логистическая регрессия

Правдоподобие позволяет понять, насколько вероятно получить данные значения таргета y при данных X и весах w

$$p(y \mid X, w) = \prod_i p(y_i \mid x_i, w)$$

для распределения Бернулли правдоподобие можно определить следующим образом:

$$p(y \mid X, w) = \prod_i p_i^{y_i} (1 - p_i)^{1 - y_i}$$

где p_i – это вероятность, поссчитанная из ответов модели

Логистическая регрессия

Оптимизировать произведение неудобно, перейдем к сумме, то есть перейдём к логарифмическому правдоподобию и подставим формулу для вероятности, которую мы получили выше:

$$\ell(w, X, y) = \sum_i \big(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\big) =$$

$$= \sum_i \big(y_i \log(\sigma(\langle w, x_i \rangle)) + (1 - y_i) \log(1 - \sigma(\langle w, x_i \rangle))\big)$$

Логистическая регрессия

Формула с прошлого слайда:

$$\ell(w, X, y) = \sum_i \left(y_i \log(\sigma(\langle w, x_i \rangle)) + (1 - y_i) \log(1 - \sigma(\langle w, x_i \rangle))\right)$$

если заметить, что

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

можно loss функцию переписать проще

$$\ell(w, X, y) = \sum_i \left(y_i \log(\sigma(\langle w, x_i \rangle)) + (1 - y_i) \log(\sigma(-\langle w, x_i \rangle))\right)$$

Логистическая регрессия

Нас интересует w, для которого правдоподобие максимально.

Умножим на минус один, чтобы получить функцию потерь, которую будем минимизировать с помощью градиентного спуска:

$$\ell(w, X, y) = -\sum_i \left(y_i \log(\sigma(\langle w, x_i \rangle)) + (1 - y_i) \log(\sigma(-\langle w, x_i \rangle))\right)$$

Распределение Бернулли

Распределение Бернулли - это дискретное распределение с двумя возможными исходами: успех (обычно обозначается как 1) и неудача (обычно обозначается как 0).

Оно моделирует случайную переменную, которая принимает только эти два значения с некоторой вероятностью успеха (p) и вероятностью неудачи (1-p).

Это распределение применяется для моделирования бинарных данных, таких как броски монеты, клики на веб-странице или прогнозирование успеха или неудачи определенного события.

Распределение числа успехов в серии из n испытаний Бернулли называют биномиальным распределением. Вероятность того, что в серии из n испытаний Бернулли произойдет ровно k успехов равна: $P(X = k) = C_n^k \cdot p^k \cdot (1-p)^{n-k}, k = 0, 1, 2, \ldots, n.$

Maximum likelihood estimator

Пусть $Data = \{x_1, \dots, x_n\}$ – выборка из **известного** распределения вероятностей. ***Правдоподобием выборки*** называется $L(Data) = P(Data) = P(x_1 \cap x_2 \cap \dots \cap x_n)$, т.е. вероятность одновременно получить наблюдения $x_1, \dots, x_n$ (имеющуюся выборку). Поскольку выборка по определению – последовательность независимых одинаково распределенных случайных величин, то $P(x_1 \cap x_2 \cap \dots \cap x_n) = P(x_1) \times P(x_2) \times \dots \times P(x_n) = \prod_{i=1}^{n} P(x_i)$

Например, пусть дана выборка из B(n=10, p=0.8), состоящая из следующих трех наблюдений: 5, 7, 9. То есть три раза независимым образом была проведена серия из 10 испытаний Бернулли с вероятностью успеха 0.8. Первая серия из 10 испытаний привела к 5 успехам (первое наблюдение в выборке); вторая – к 7 успехам; третья – к 10. Чему равно правдоподобие этой выборки? То есть какова вероятность получить именно такую выборку?

Ответ: $L(5,7,9) = P(X = 5 \cap X = 7 \cap X = 9) = C_{10}^{5} 0.8^5 0.2^5 C_{10}^{7} 0.8^7 0.2^3 C_{10}^{9} 0.8^9 0.2 \approx 0.001428.$

https://www.hse.ru/data/2012/11/21/1248365473/%D0%9C%D0%9C%D0%9F%20%D0%B4%D0%BB%D1%8F%202%20%D0%BA%D1%83%D1%80%D1%81%D0%B0%20%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D0%BE%D0%B2%20(%D0%BF%D0%BE%D0%B2%D1%82%D0%BE%D1%80%D0%B5%D0%BD%D0%B8%D0%B5%20-%20%D1%87%D1%8C%D1%8F-%D1%82%D0%BE%20%D0%BC%D0%B0%D1%82%D1%8C).pdf