Основы машинного обучения

Лекция 11

Линейная классификация. Многоклассовая классификация.

Евгений Соколов

esokolov@hse.ru

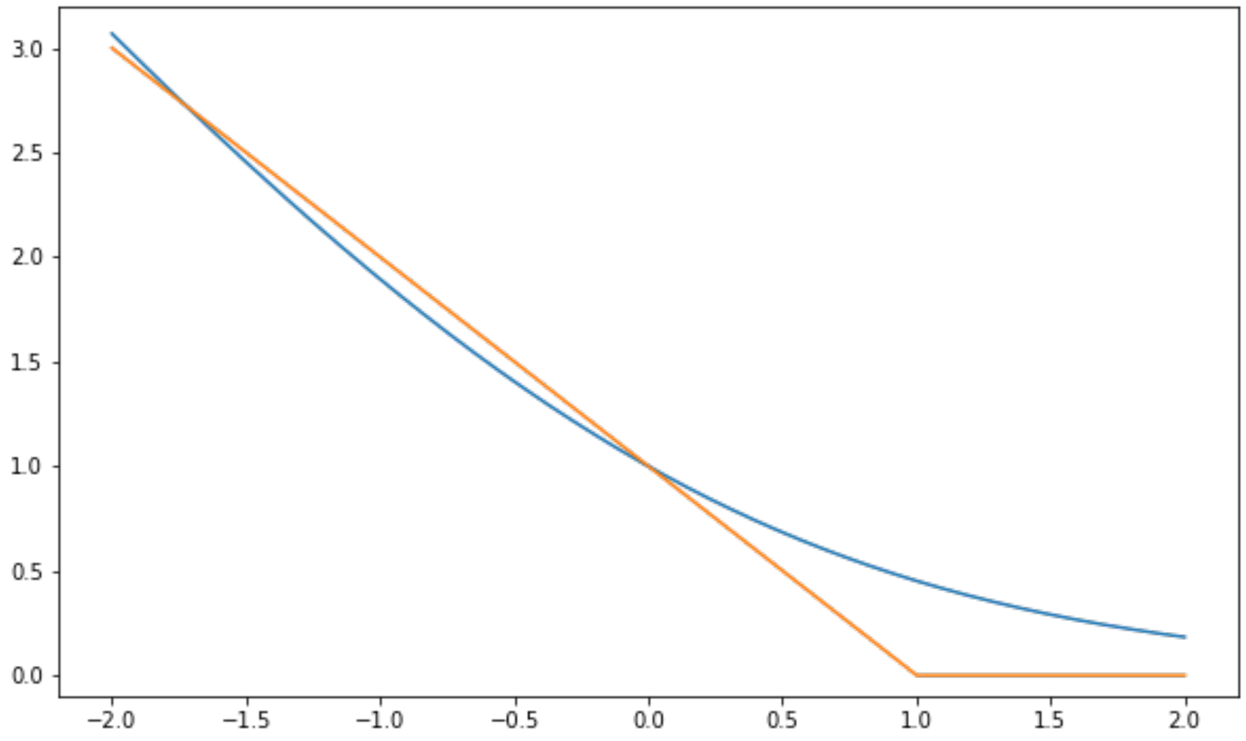
НИУ ВШЭ, 2024

Метод опорных векторов

$$C\sum_{i=1}^{\ell}\max\big(0, 1 - y_i(\langle w, x_i\rangle + w_0)\big) \;\; + \|w\|^2 \to \min_{w, w_0}$$

- Функция потерь (hinge loss) + регуляризация

Сравнение логистической регрессии и SVM



Резюме

- Логистическая регрессия — обучение модели так, что на объектах с близкими прогнозами эти прогнозы стремятся к доле положительных объектов

- Метод опорных векторов основан на идее максимизации отступа классификатора

Логистическая регрессия: сложное объяснение

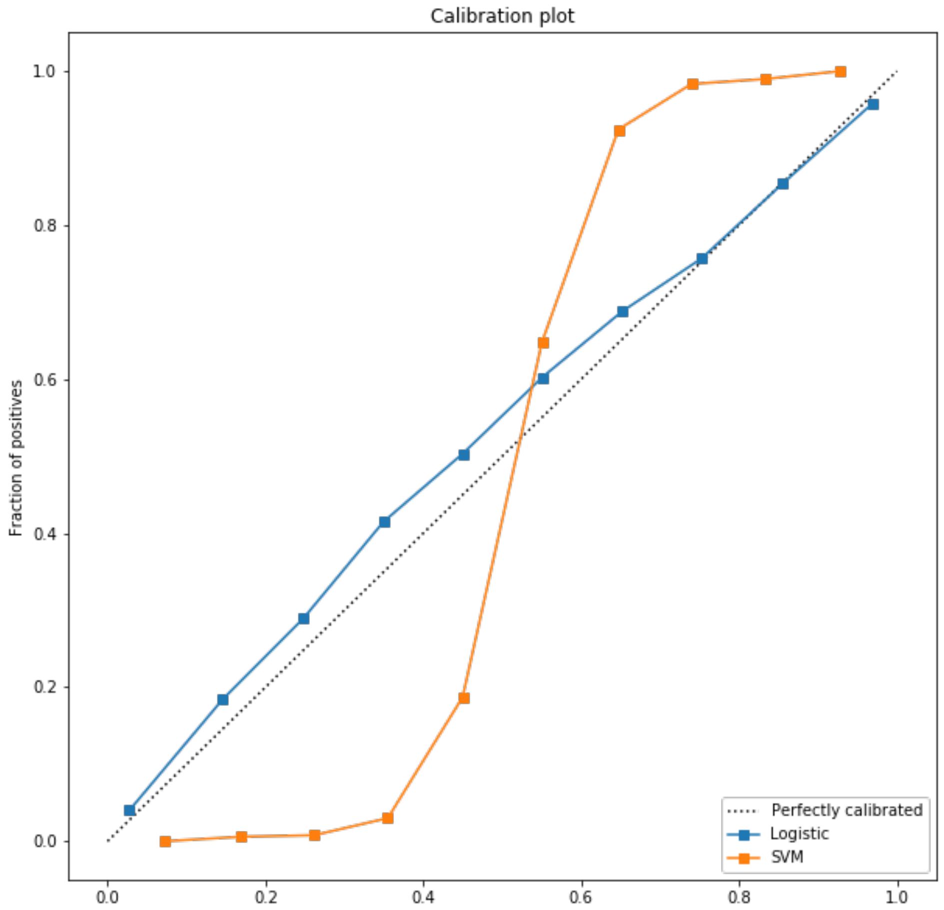
Предсказание вероятностей

Будем говорить, что модель $b(x)$ предсказывает вероятности, если среди объектов с $b(x) = p$ доля положительных равна p.

Калибровочная кривая

- Разобьём отрезок $[0, 1]$ на n корзинок $[0, t_1], [t_1, t_2], \dots, [t_{n-1}, 1]$ — это ось X
- Для каждого отрезка $[t_i, t_{i+1}]$ берём объекты, для которых $b(x) \in [t_i, t_{i+1}]$
- Считаем среди объектов долю положительных, откладываем её на оси Y

Калибровочная кривая



Предсказание вероятностей

- Функционал ошибки:

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} L\big(y_i, b(x_i)\big) \to \min_a$$

Предсказание вероятностей

- Функционал ошибки:

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} L\big(y_i, b(x_i)\big) \to \min_a$$

- Рассмотрим ошибку только на объектах $x_1, \dots, x_n$, где модель $b(x)$ выдаёт вероятность около p:

$$\sum_{i=1}^{n} L\big(y_i, b(x_i)\big) = \sum_{i=1}^{n} L(y_i, p)$$

- А что было бы оптимально выдать на этих объектах?

Предсказание вероятностей

- Рассмотрим ошибку только на объектах $x_1, \dots, x_n$, где модель $b(x)$ выдаёт вероятность около p:

$$\sum_{i=1}^{n} L\big(y_i, b(x_i)\big) = \sum_{i=1}^{n} L(y_i, p)$$

- А что было бы оптимально выдать на этих объектах?

$$p_* = \arg\min \sum_{i=1}^{n} L(y_i, p)$$

- Мы ожидаем, что $p_* = \frac{1}{n}\sum_{i=1}^{n}[y_i = +1]$

Log-loss

- Рассмотрим ошибку только на объектах, где модель $b(x)$ выдаёт вероятность около p:

$$\sum_{i=1}^{n} L\big(y_i, b(x_i)\big) = \sum_{i=1}^{n} L(y_i, p)$$

- А что было бы оптимально выдать на этих объектах?

$$p_* = \arg\min \sum_i \{-[y_i = +1] \log p - [y_i = -1] \log(1 - p)\}$$

Log-loss

$$p_* = \arg\min \sum_i \{-[y_i = +1] \log p - [y_i = -1] \log(1-p)\}$$

- Посчитаем производную по p и приравняем к нулю:

$$\sum_i \left\{-\frac{[y_i = +1]}{p} + \frac{[y_i = -1]}{1-p}\right\} = -\frac{n_+}{p} + \frac{n_-}{1-p} = 0$$

$$p_* = \frac{n_+}{n_+ + n_-} = \frac{1}{n}\sum_{i=1}^{n}[y_i = +1]$$

Предсказание вероятностей

- Считаем, что модель корректно оценивает вероятности, если для любых $y_1, \dots, y_n \in \mathbb{Y}$

$$\arg\min \sum_{i=1}^{n} L(y_i, p) = \frac{1}{n} \sum_{i=1}^{n} [y_i = +1]$$

- Это условие на функцию потерь
- Оно выполнено для log-loss, то есть логистическая регрессия корректно оценивает вероятности
- Значит, для объектов с близкими вероятностями она будет пытаться выдать число, близкое к доле положительных объектов

MSE

$$p_* = \arg\min \sum_{i=1}^{n} (p - [y_i = +1])^2$$

- Посчитаем производную по p и приравняем к нулю:

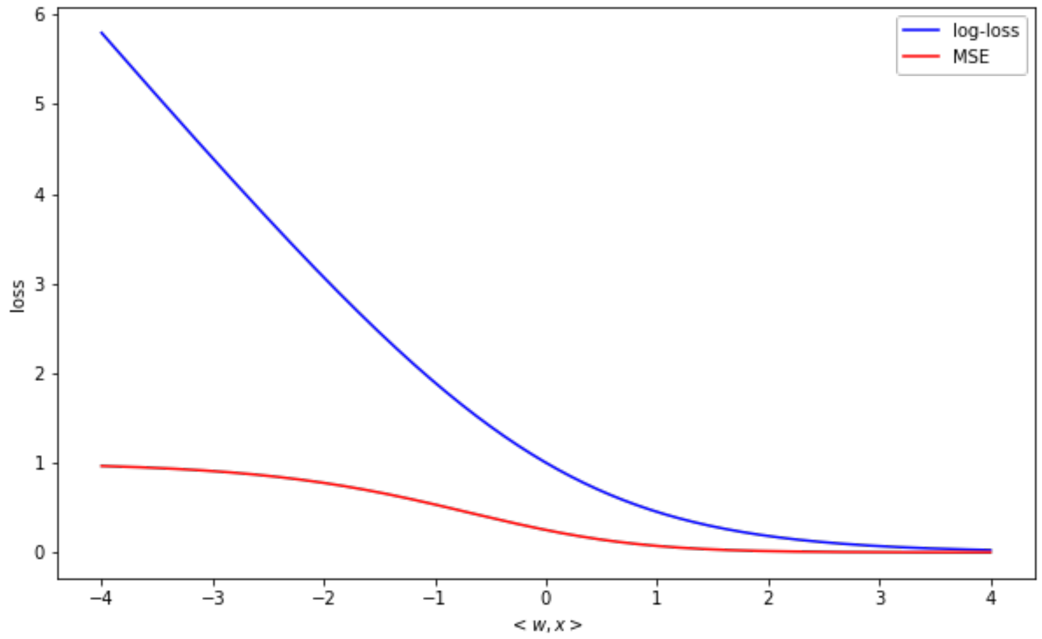
$$2 \sum_{i=1}^{n} (p - [y_i = +1]) \quad = 0$$

$$p_* = \frac{1}{n} \sum_{i=1}^{n} [y_i = +1]$$

MSE

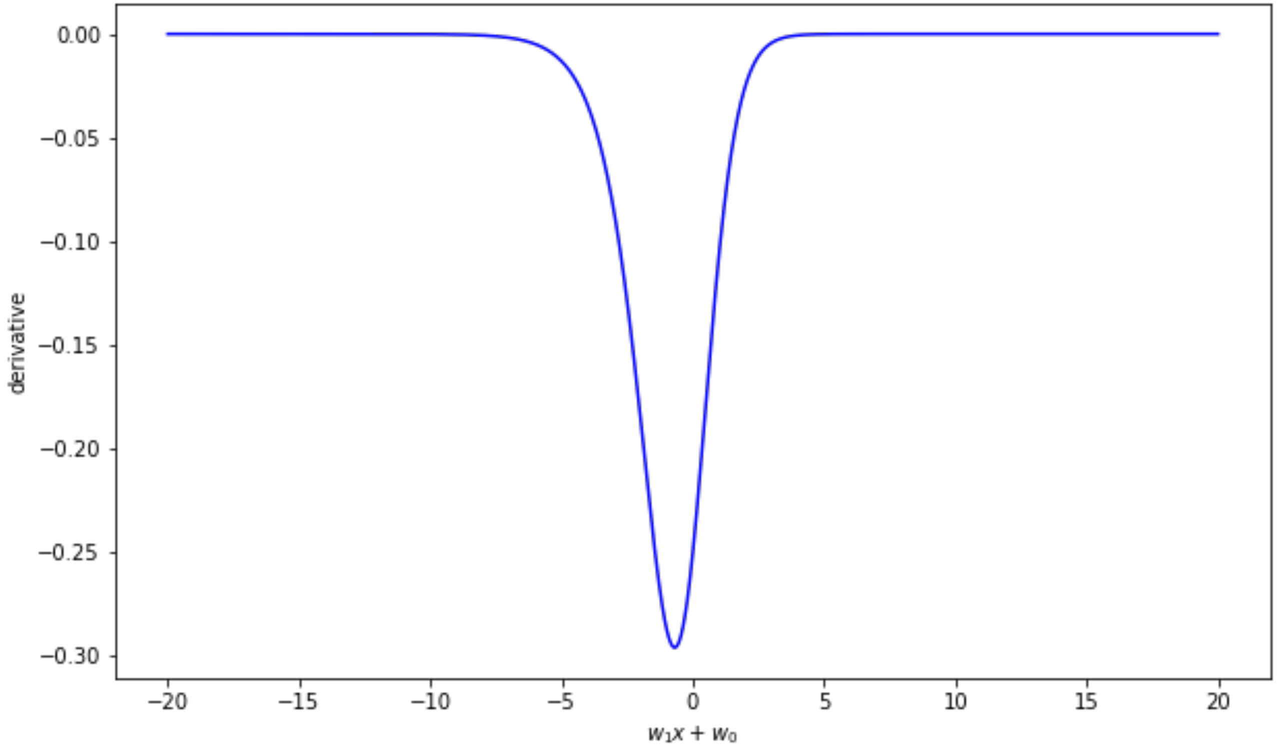
- Почему бы не обучать классификаторы на MSE?

$$\sum_{i=1}^{n} (\sigma(\langle w, x_i \rangle) - [y_i = +1])^2 \to \min_w$$



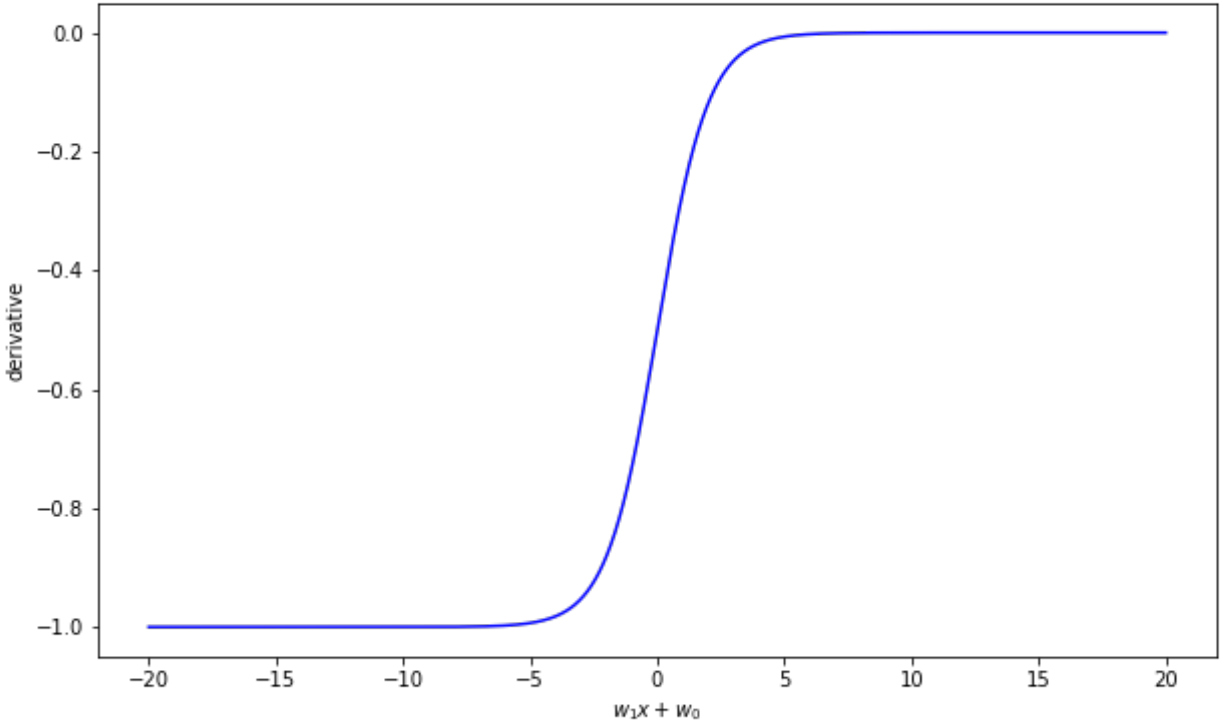
MSE

$$\frac{\partial}{\partial w_1}\left(\frac{1}{1+e^{-w_1 x - w_0}} - 1\right)^2 = -\frac{2xe^{w_1 x + w_0}}{(1+e^{w_1 x + w_0})^3}$$



Log-loss

$$\frac{\partial}{\partial w_1}\left(\log \frac{1}{1+e^{-w_1 x - w_0}}\right) = \frac{x}{1+e^{w_1 x + w_0}}$$



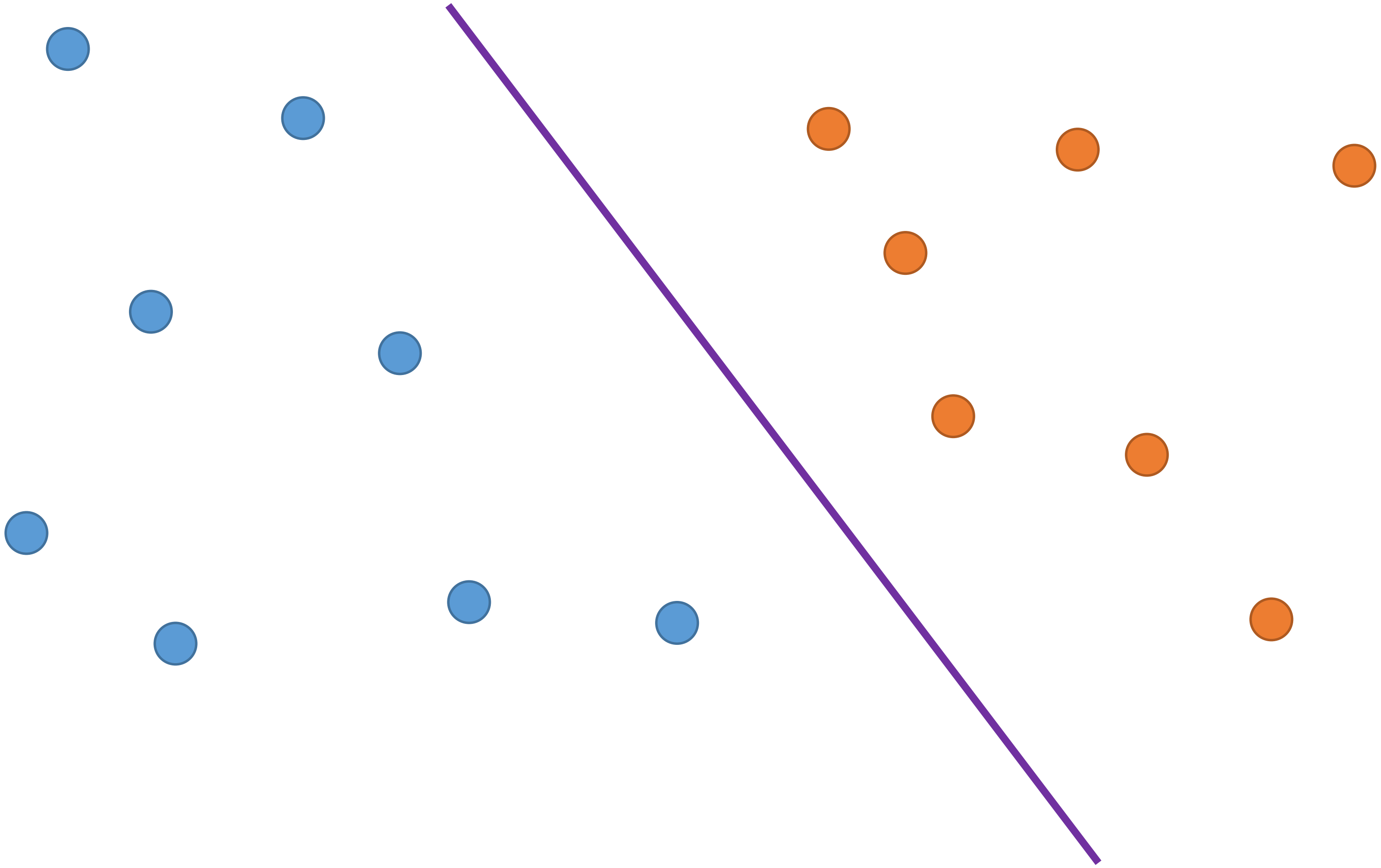
MAE

$$p_* = \arg\min \sum_{i=1}^{n} |p - [y_i = +1]|$$

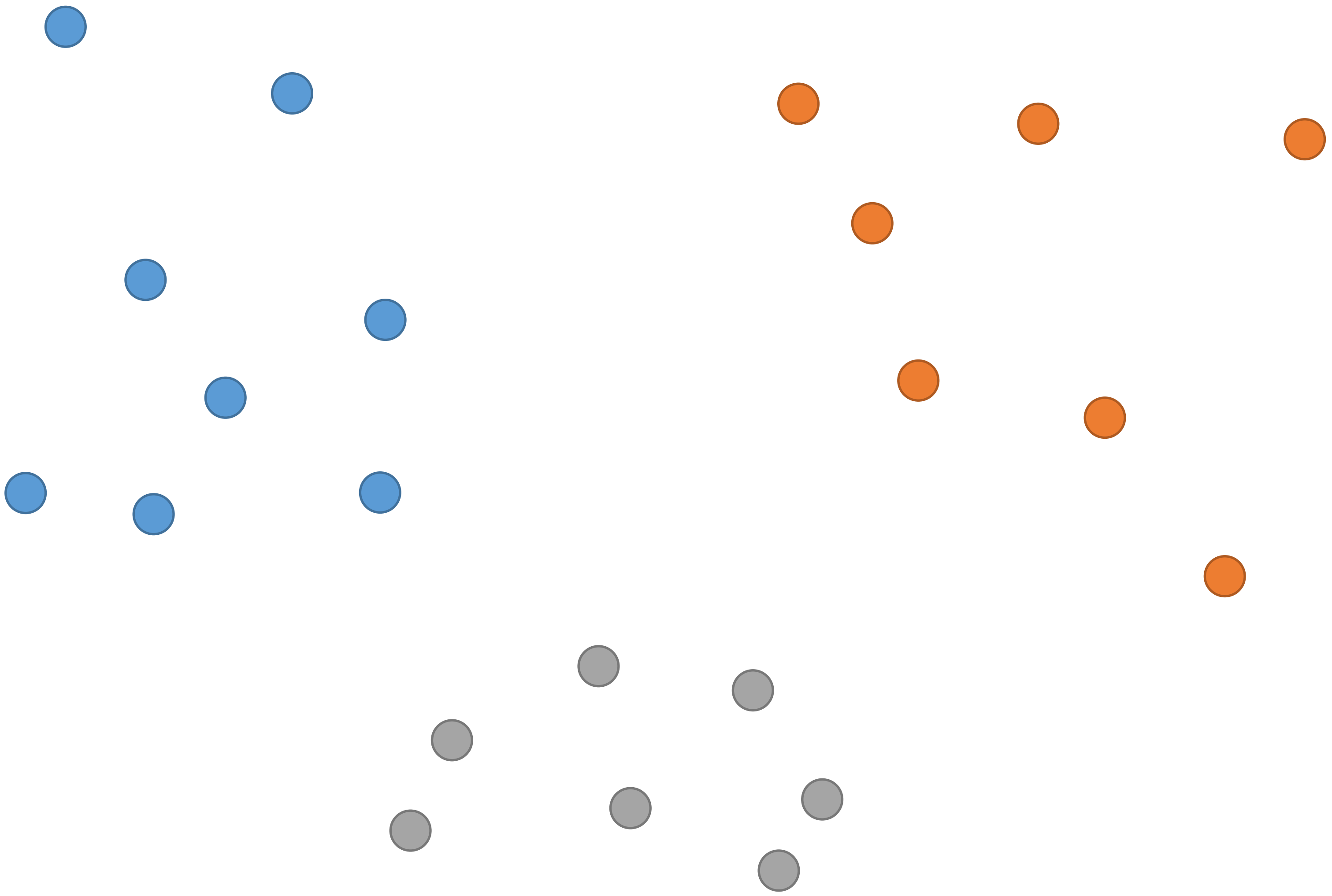
- Можно показать, что p_* равно либо 0, либо 1

Многоклассовая классификация

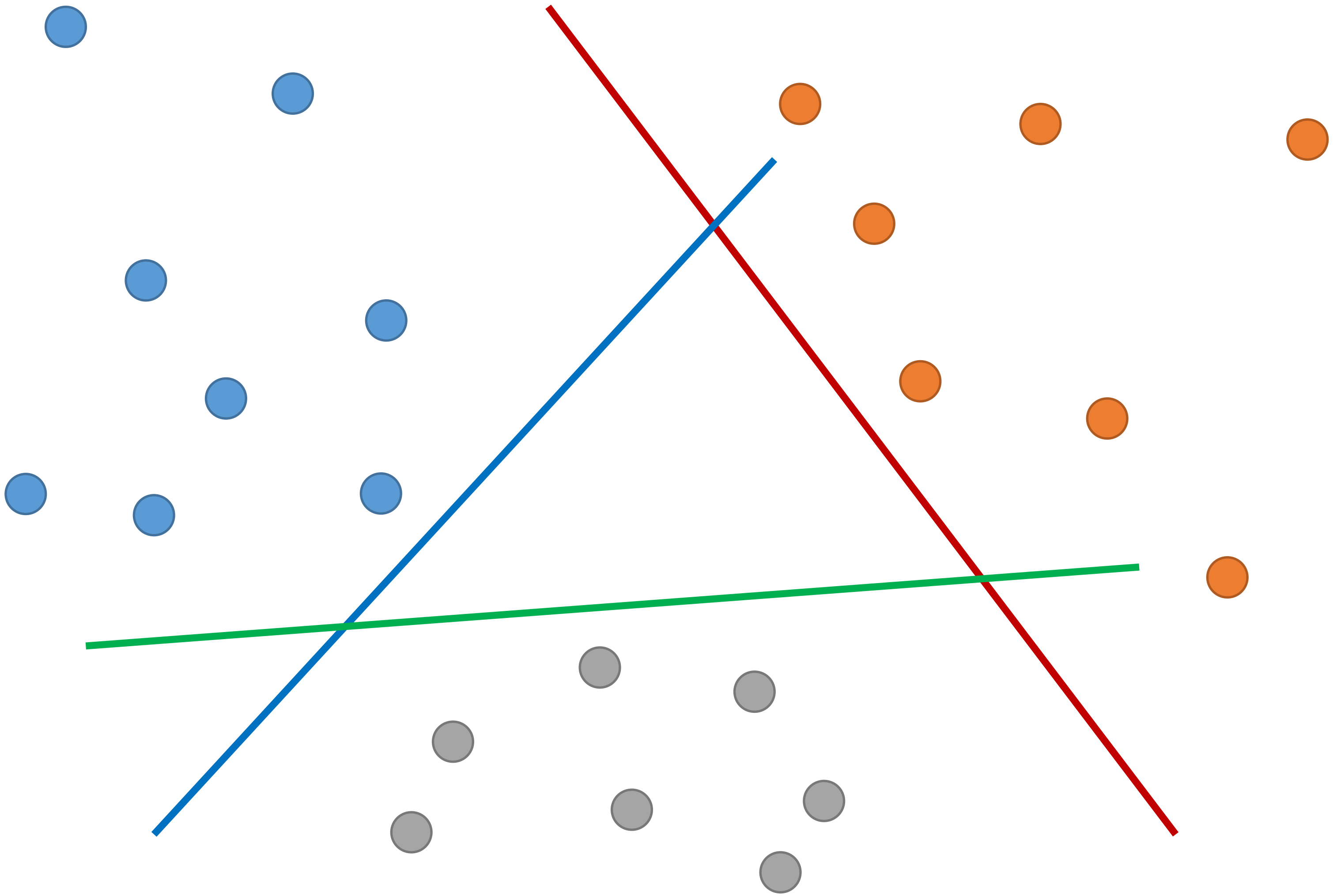
Бинарная классификация



Многоклассовая классификация



Многоклассовая классификация



One-vs-all

- K классов: $\mathbb{Y} = \{1, \dots, K\}$
- $X_k = (x_i, [y_i = k])_{i=1}^{\ell}$
- Обучаем $a_k(x)$ на X_k, $k = 1, \dots, K$
- $a_k(x)$ должен выдавать оценки принадлежности классу (например, $\langle w, x \rangle$ или $\sigma(\langle w, x \rangle)$)
- Итоговая модель:

$$a(x) = \arg \max_{\mathrm{k=1,\dots K}} a_k(x)$$

One-vs-all

- Модель $a_k(x)$ при обучении не знает, что её выходы будут сравнивать с выходами других моделей

- Нужно обучать K моделей

All-vs-all

- $X_{km} = \{(x_i, y_i) \in X \mid y_i = k \text{ или } y_i = m\}$
- Обучаем $a_{km}(x)$ на X_{km}
- Итоговая модель:

$$a(x) = \arg \max_{k \in \{1,\dots,K\}} \sum_{m=1}^{K} [a_{km}(x) = k]$$

All-vs-all

- Нужно обучать порядка K^2 моделей
- Зато каждую обучаем на небольшой выборке

Доля ошибок

- Функционал ошибки — доля ошибок (error rate)

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i]$$

- Нередко измеряют долю верных ответов (accuracy):

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i]$$

- Подходит для многоклассового случая!

Общие подходы

Микро-усреднение	Макро-усреднение
Вычисляем $\mathrm{TP}_k, \mathrm{FP}_k, \mathrm{FN}_k, \mathrm{TN}_k$ для каждого класса	Вычисляем нужную метрику для каждого класса (например, $\mathrm{precision}_1, \dots, \mathrm{precision}_K$)
Суммируем по всем классам, получаем $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}, \mathrm{TN}$	Усредняем по всем классам
Подставляем их в формулу для precision/recall/…	
Крупные классы вносят больший вклад	Игнорирует размеры классов

Как делать нелинейные модели

Предсказание стоимости квартиры

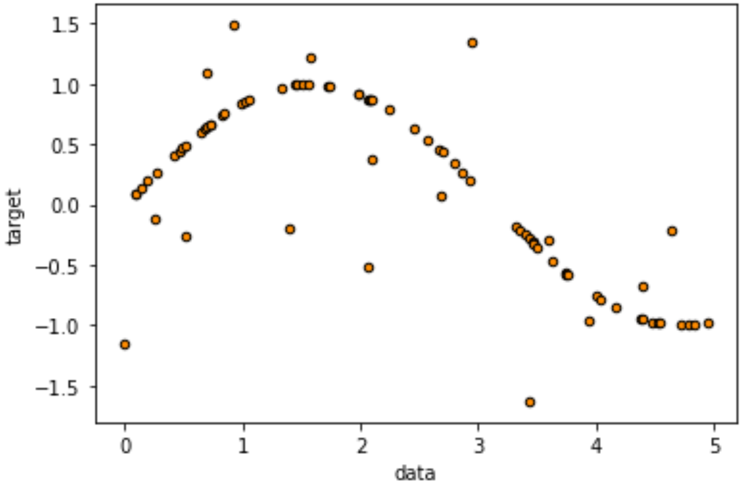
- Признаки: площадь, этаж, расстояние до метро и т.д.
- Целевая переменная: рыночная стоимость квартиры

Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$
$$+ w_3 * (\text{расстояние до метро}) + \cdots$$

- Вряд ли признаки линейно связаны с целевой переменной



Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$
$$+ w_3 * (\text{расстояние до метро}) + \cdots$$

- Вряд ли признаки не связаны между собой

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$

$$+w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2$$

$$+w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2$$

$$+w_7 * (\text{площадь}) * (\text{этаж}) + \cdots$$

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$
$$+ w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2$$
$$+ w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2$$
$$+ w_7 * (\text{площадь}) * (\text{этаж}) + \cdots$$

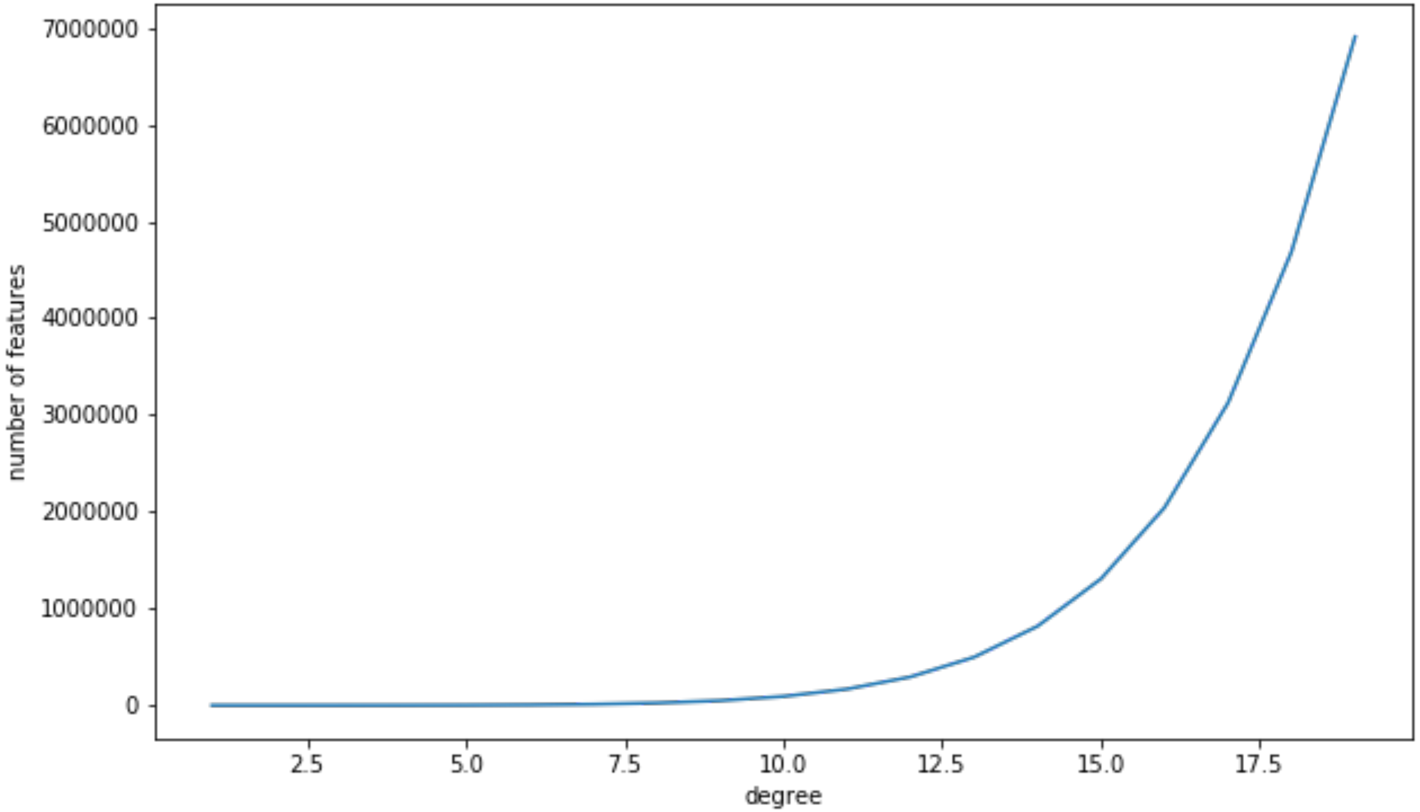
- Может быть сложно интерпретировать модель
- Что такое $(\text{расстояние до метро}) * (\text{этаж})^2$?

Предсказание стоимости квартиры

- Допустим, изначально имеем 10 признаков
- Полиномиальных степени 2: 55
- Полиномиальных степени 3: 220
- Полиномиальных степени 4: 715

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:



Предсказание стоимости квартиры

- Линейная модель с полиномиальными бинаризованными признаками:

$$a(x) = w_0 + w_1 * [30 < \text{площадь} < 50]$$

$$+w_2 * [50 < \text{площадь} < 80] + \cdots$$

$$+w_{20} * [2 < \text{этаж} < 5] + \cdots$$

$$+w_{100} * [30 < \text{площадь} < 50][2 < \text{этаж} < 5] + \cdots$$

- Признаки интерпретируются куда лучше: $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][100 < \text{расстояние до метро} < 500]$
- Но их станет ещё больше!