

12.03.2024

Машинное обучение

Лекция 2. Линейные модели

(12.03.2024)

Общие сведения

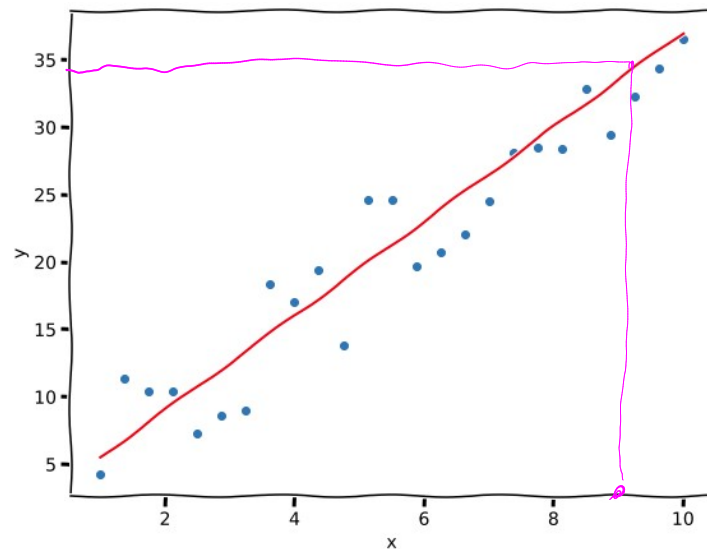
План

1. Линейная модель регрессии
2. Как линейные модели обучаются?
3. Линейная модель классификации

Что это такое?

x — баллы за экзамен по английскому 1
 y — баллы за экзамен по английскому 2

x	y
1	5
3	11
9	35
10	33



Что это такое?

$$\hat{y} = w_0 + w_1 x$$

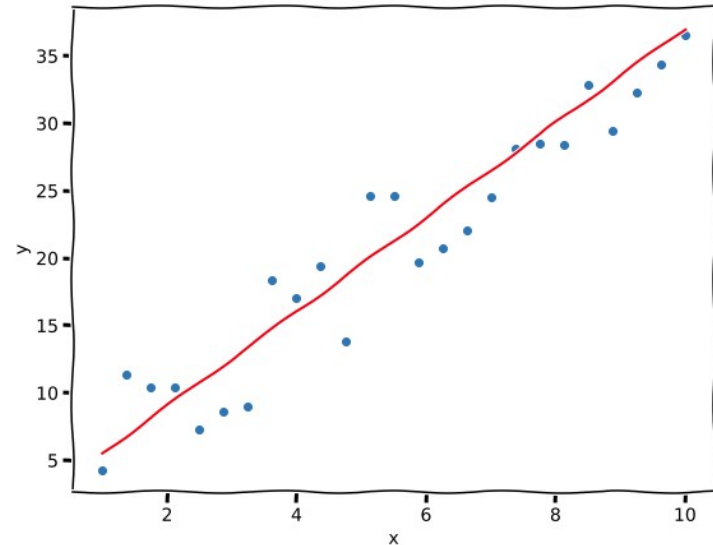
свободный член
(intercept)

случайная ошибка
(error)

$$y = w_0 + w_1 x + \epsilon$$

целевая переменная
(aka target)

независимая переменная
(slope, predictor)



А какая модель нам нужна?

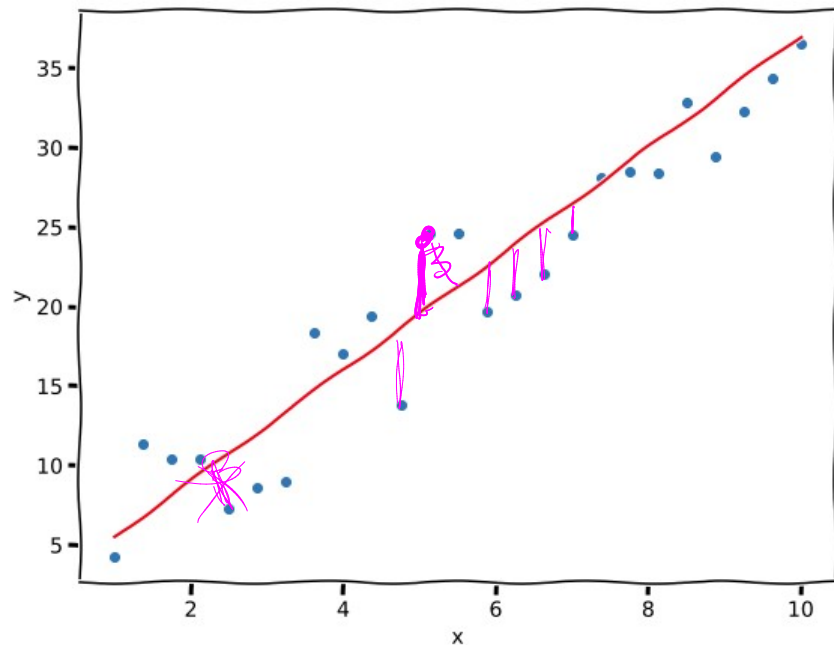
$$\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \rightarrow \min$$

//
MSE

$$\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \rightarrow \min$$

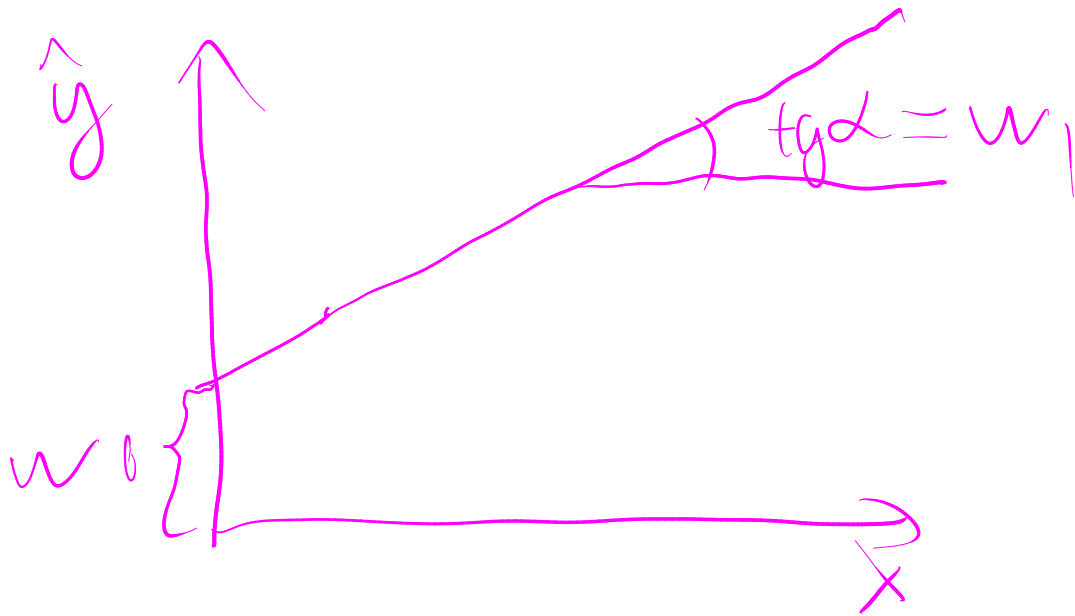
//
MAE

$$y = w_0 + w_1 x + \varepsilon$$
$$\hat{y} = w_0 + w_1 x$$



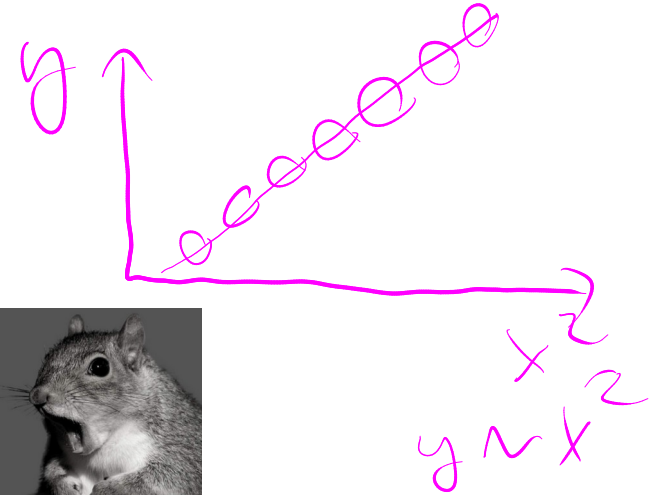
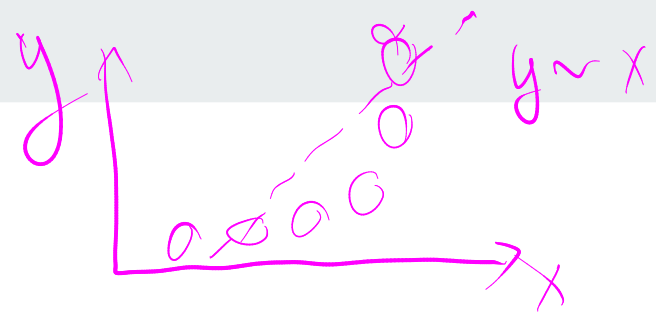
Интерпретация коэффициентов

$$\hat{y} = w_0 + w_1 x$$



Зачем нужны линейные модели?

1. Предсказание интересующей нас величины
2. Оценка влияния различных факторов на нашу целевую переменную
3. Линейные модели очень легко использовать и интерпретировать
4. Линейные модели могут восстанавливать даже **нелинейные зависимости**

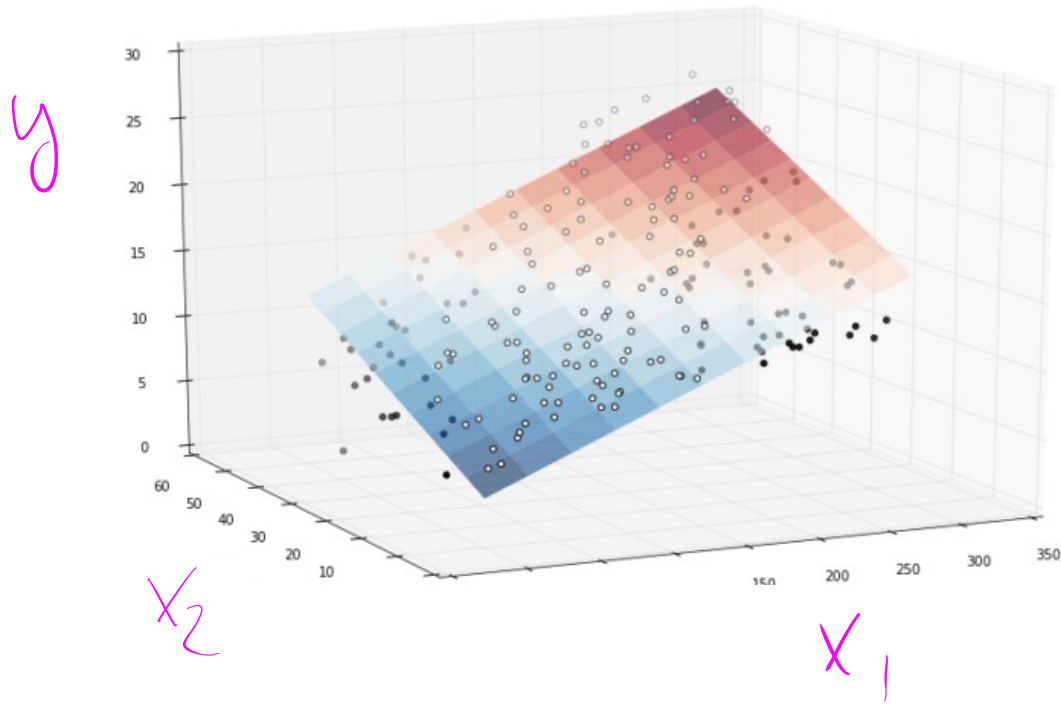


А если у нас много независимых переменных?

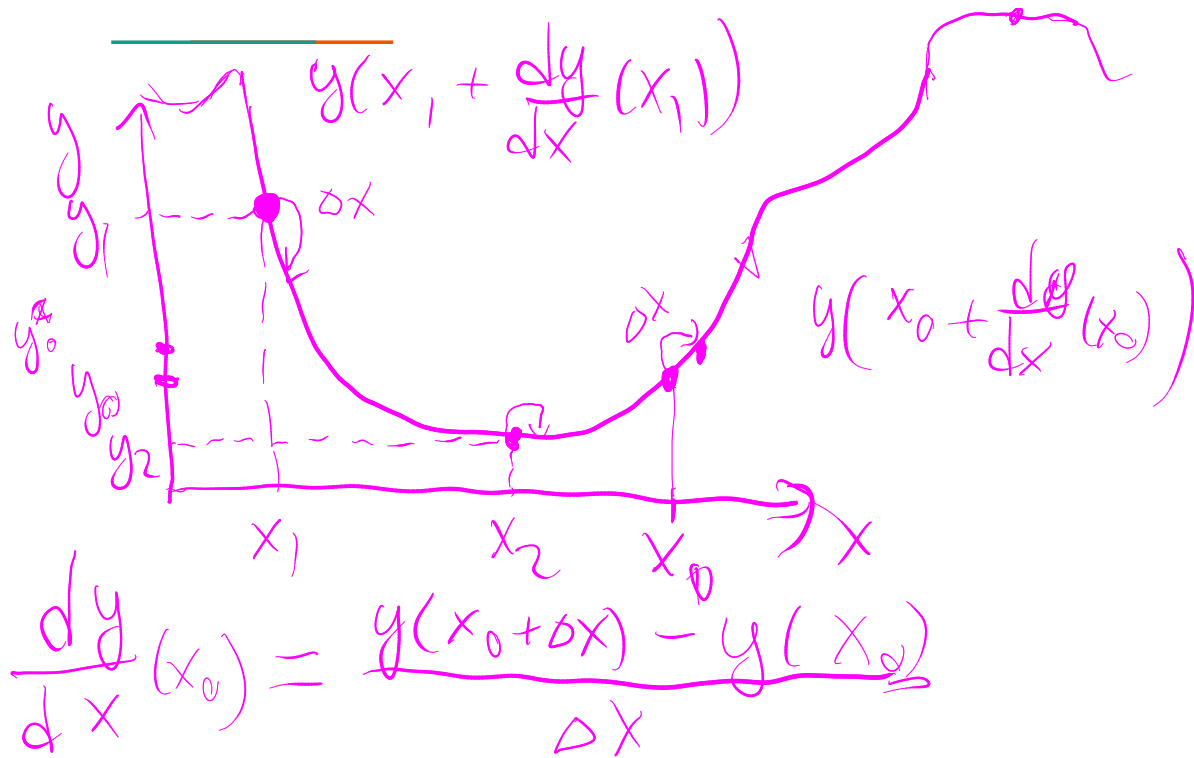
$$y = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n + \epsilon$$

площадь	число комнат	школа близко	цена квартиры
50	2	нет	5000
1000	7	да	11000
30	1	нет	3500
100	4	нет	33333

Множественная линейная регрессия дает нам плоскость



Производные



$y = f(x)$	$\frac{dy}{dx} = f'(x)$
k , any constant	0
x	1
x^2	$2x$
x^3	$3x^2$
x^n , any constant n	nx^{n-1}
e^x	e^x
e^{kx}	ke^{kx}
$\ln x = \log_e x$	$\frac{1}{x}$
$\sin x$	$\cos x$
$\sin kx$	$k \cos kx$
$\cos x$	$-\sin x$
$\cos kx$	$-k \sin kx$

Производные

$$\varphi(x, y, z) = 2x^2 + 3y^2 - \sin z$$

$$\frac{\partial \varphi}{\partial x} = 2 \cdot 2 \cdot x = 4x$$

$$\frac{\partial \varphi}{\partial y} = 2 \cdot 3 \cdot y = 6y$$

$$\frac{\partial \varphi}{\partial z} = -\cos z$$

$$\begin{pmatrix} 4x \\ 6y \\ -\cos z \end{pmatrix}$$

градиент φ
 $\nabla_{x,y,z} \varphi$

Производные



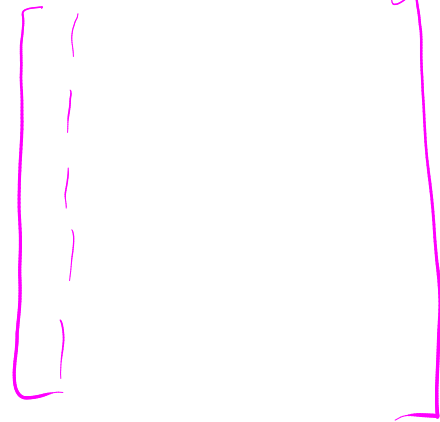
Производные



Как оценивать коэффициенты модели?

X , y , w
 $n \times 4$ $n \times 1$ 5×1

X^* x_1 x_2 x_3 x_4



$$\hat{y}_i = w_0 + w_1 x_{i1} + \dots + w_n x_{in}$$

$$y = X^* \cdot w$$

$n \times 1$ $n \times 5$ 5×1

$$[w_0 + x_{i1} w_1 + \dots]$$



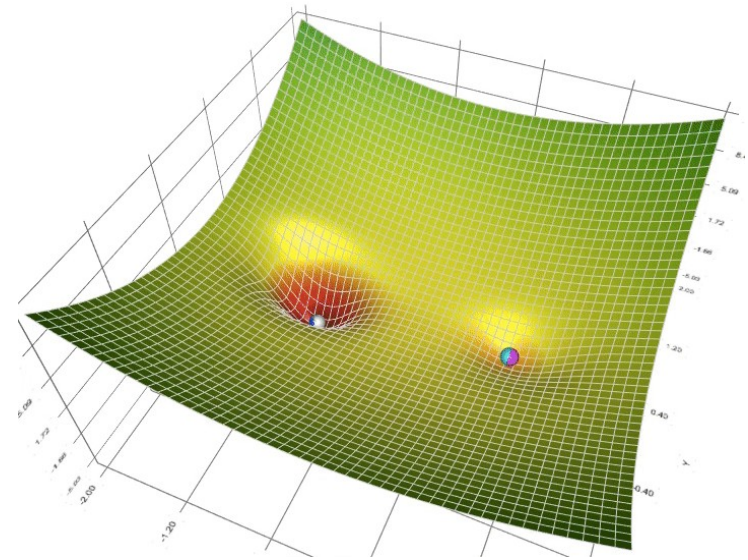
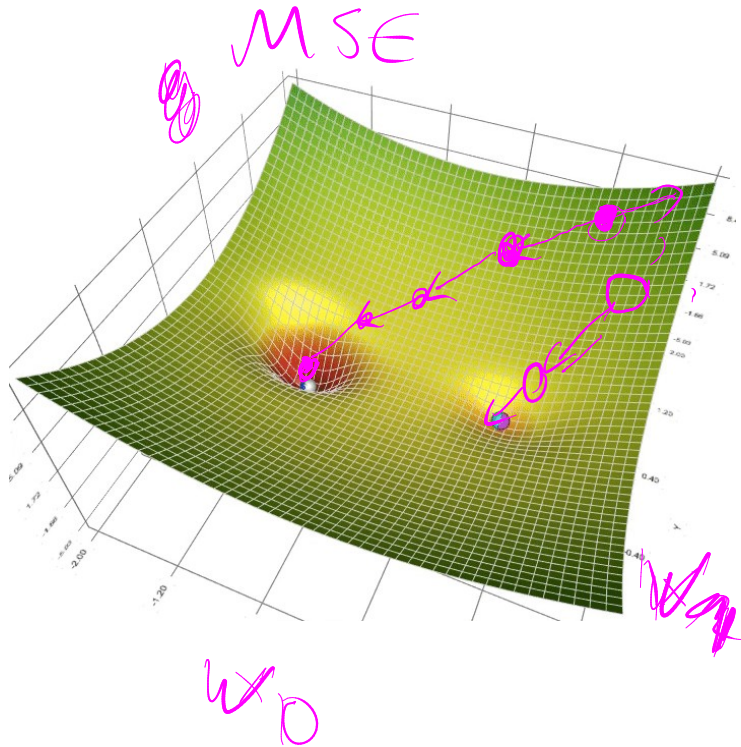
Как оценивать коэффициенты модели?

$$\underline{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \rightarrow \min$$

w

$$\hat{y}_i = w_0 + w_1 x_{1i} + \dots + w_n x_{ni}$$

Градиентный спуск



Формулы

$$y = w_0 + w_1 x + \epsilon$$

$$y = Xw$$

$$\frac{dLoss}{dw} = \nabla Loss = 2X^T(Xw - y)$$

$\nabla = 0$

$$Loss = (y - Xw)^T (y - Xw)$$

MSE

$$w = (X^T X)^{-1} X^T y$$

Градиентный спуск

$$Loss = (y - Xw)^T (y - Xw) \quad \frac{dLoss}{dw} = \nabla Loss = 2X^T (Xw - y)$$

```
w = np.random.randn(m + 1)
Пока grad(Loss) != 0:
    w -=  $\eta$  *
    grad(Loss)
```

$$w -= \eta \cdot \text{grad}(Loss)$$

learning rate
[$1e^{-5}$; $1e^{-1}$]

Отдых -> логистическая регрессия

Связь событий и признаков

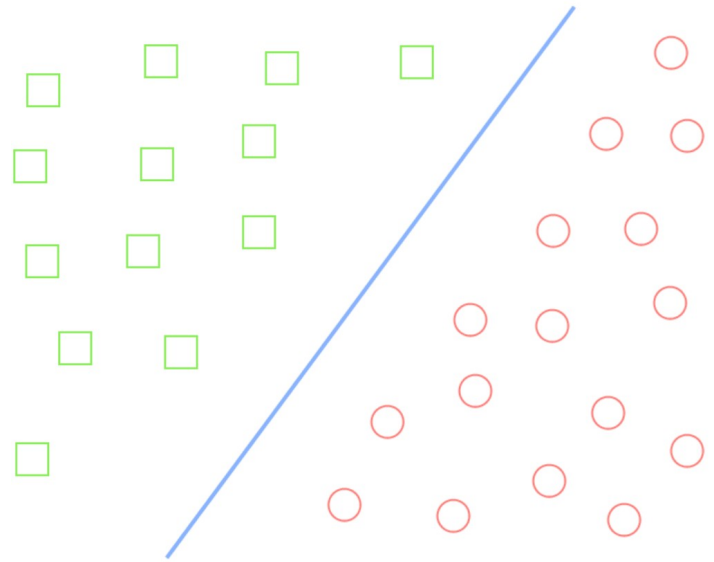
В зависимости от предикторов события могут происходить чаще или реже – логика, совпадающая с логикой связи количественной переменной отклика с набором предикторов.

Например, по мере роста температуры воздуха летом чаще будут встречаться люди в шортах: событие “встретился человек в шортах” положительно связано с температурой воздуха.

Событие “проведение исследования” явно связана с предиктором “объем полученного финансирования”, однако эта связь может быть совсем непростой.

А что если хотим классификацию?

Допустим бинарная классификация



Отношение шансов

Шансы (odds) часто представляют в виде отношения шансов (odds ratio)

Если отношение шансов > 1 , то вероятность наступления события выше, чем вероятность того, что оно не произойдет.

Если отношение шансов < 1 , то наоборот.

Если можно оценить вероятность положительного события, то отношение шансов выглядит так:

$$odds = \frac{\pi}{1-\pi}$$

Отношение шансов варьируется от 0 до $+\infty$.

Попробуем сами



Логиты

Отношение шансов можно преобразовать в логиты(logit):

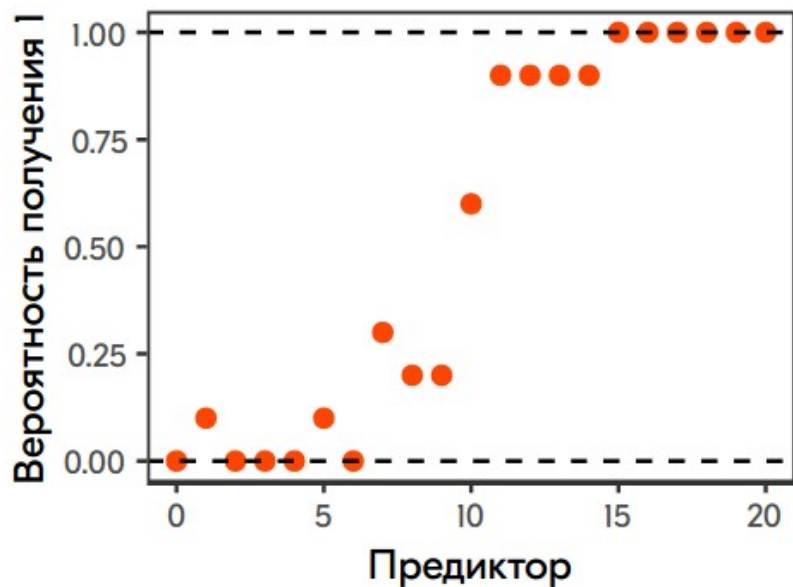
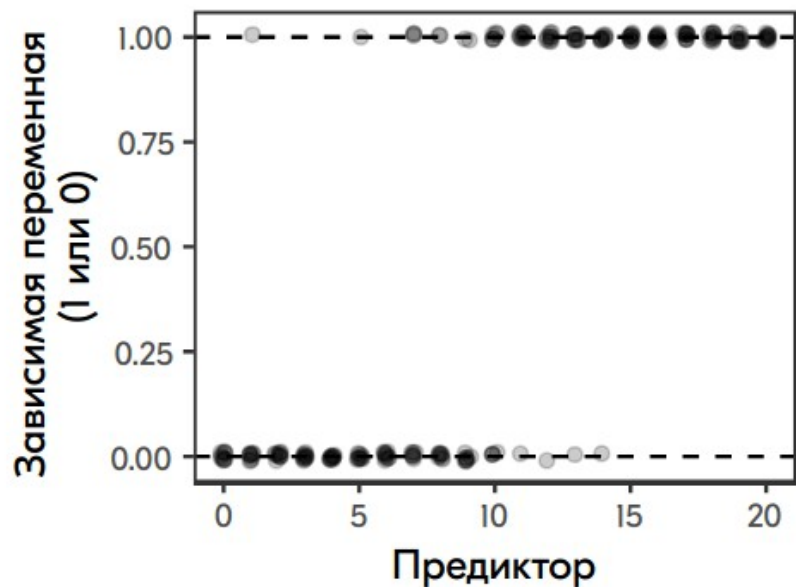
$$\ln(odds) = \ln\left(\frac{\pi}{1 - \pi}\right)$$

- Значения логитов – это трансформированные оценки вероятности события.
- Логиты варьируют от $-\infty$ до $+\infty$.
- Логиты симметричны относительно 0, т.е. $\ln(1)$.
- Для построения моделей в качестве зависимой переменной удобнее брать логиты.

Считаем вероятность



Дискретные значения vs вероятности



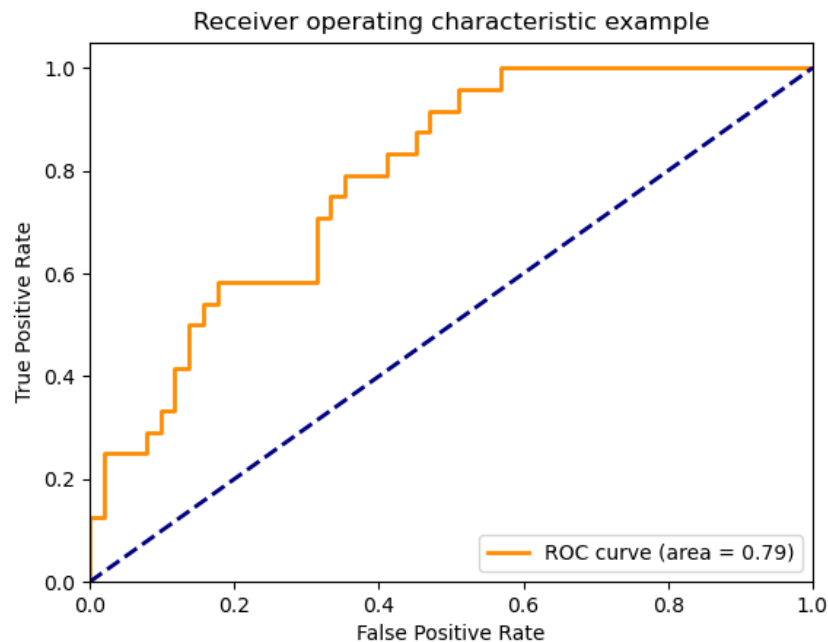
Как такое учить? BCE Loss



Качество классификации



Качество классификации. ROC кривая

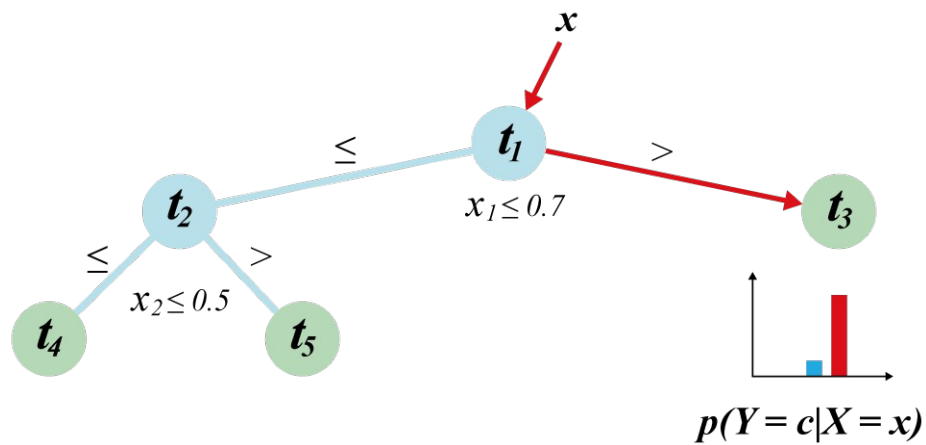
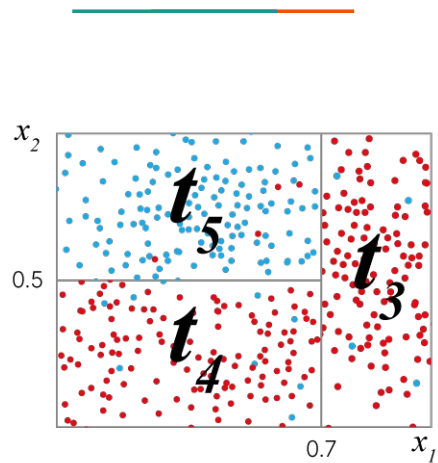


[рисуем свою ROC кривую](#)

Построение ROC кривой

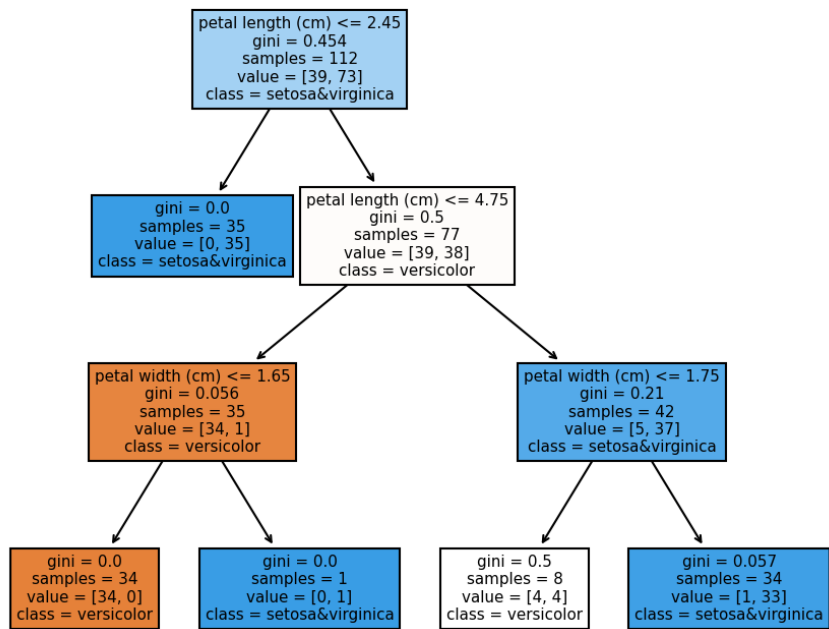
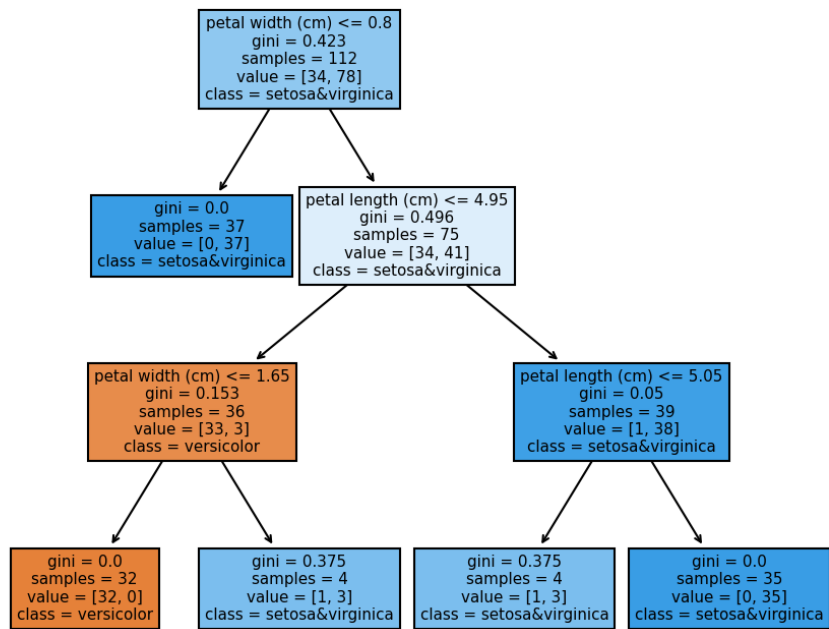


Деревья решений



Split mode

Leaf mode





Недостатки:

1. Переобучение
2. Не подходит для данных с большой размерностью
3. Беззащитны перед шумными данными

Алгоритм

1. s_0 = вычисляем энтропию исходного множества
2. Если $s_0 == 0$ значит:
 - a. Все объекты исходного набора, принадлежат к одному классу
 - b. Сохраняем этот класс в качестве листа дерева
3. Если $s_0 \neq 0$ значит:
 - a. Перебираем все элементы исходного множества:
 - b. Для каждого элемента перебираем все его атрибуты:
 - c. На основе каждого атрибута генерируем предикат, который разбивает исходное множество на два подмножества
 - d. Рассчитываем среднее значение энтропии. Вычисляем ΔS



Как будем останавливаться и формировать листья?

1. Стрижка
2. Использование остановок



Остановки

1. $\text{Impurity} = 0$
2. В лист попадает число объектов меньше заданного
3. Ограничение на количество листьев
4. Максимальная глубина
5. Вероятность классификации объекта больше заданной величины



Как подготовить данные?

1. Imputation
2. OneHotEncoder, OrdinalEncoder, custom и т.д.



Гиперпараметры

1. `max_depth = None`
2. `min_samples_split = 2`
3. `min_samples_leaf = 1`
4. `max_features = None`

Также для классификации можно в модель передать **веса**

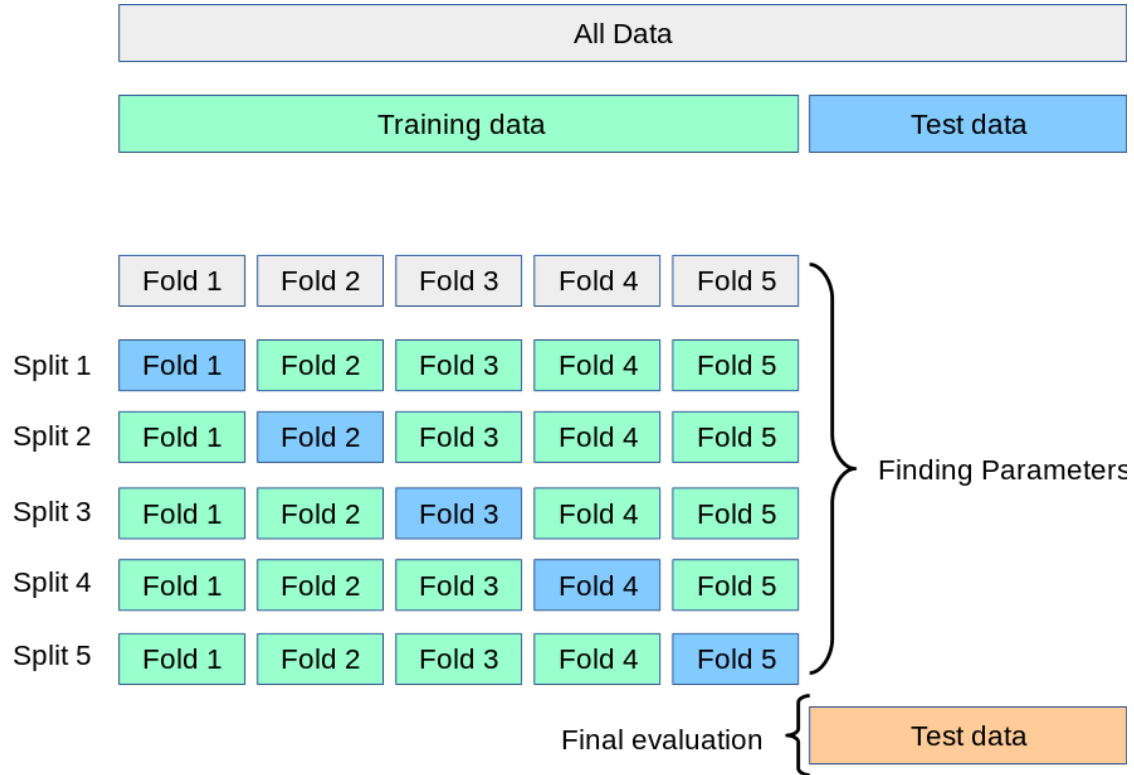
Есть ещё такая штука как **`ccp_alpha`**



Model Validation

1. Train - Test
2. Train - Valid - Test
3. Cross Validation

K - Fold



е
ОТДЫХАЙТ

