Машинное обучение

Лекция 2. Линейные модели

(12.03.2024)

Общие сведения

План

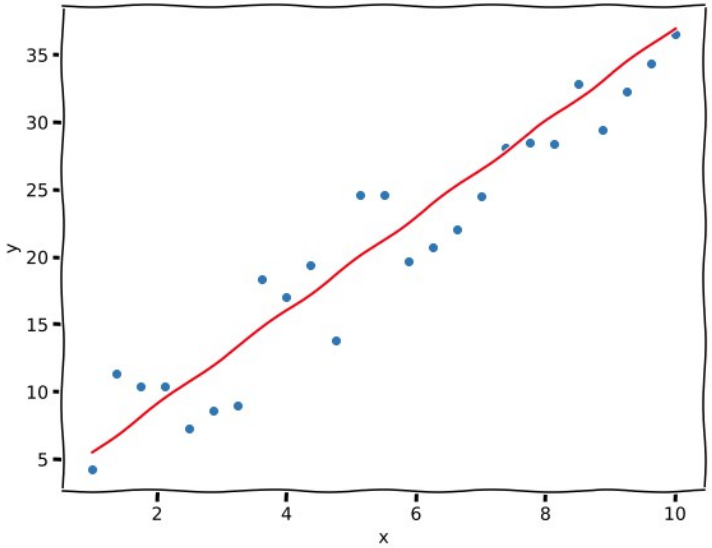
1. Линейная модель регрессии
2. Как линейные модели обучаются?
3. Линейная модель классификации

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?

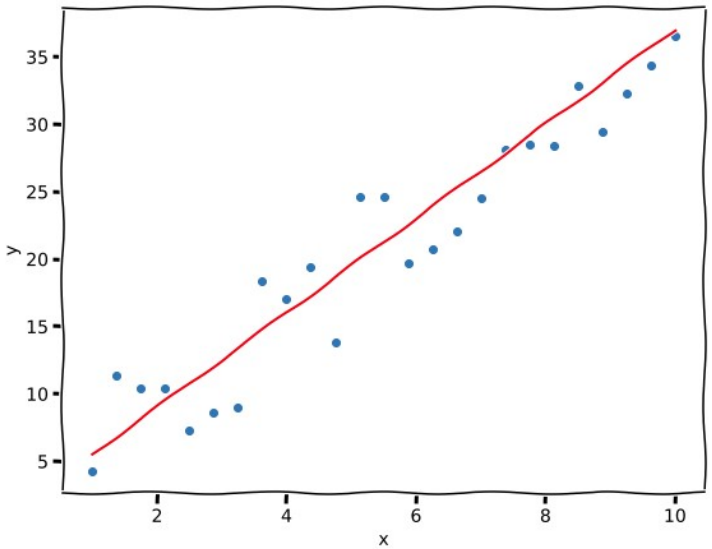
свободный член (intercept)

случайная ошибка (error)

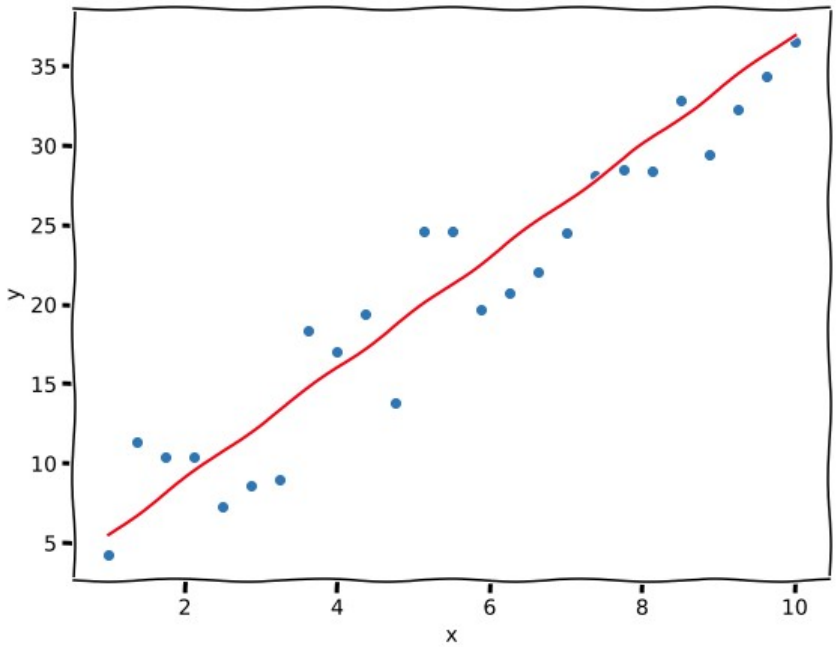
$$y = w_0 + w_1 x + \epsilon$$

целевая переменная (aka target)

независимая переменная (slope, predictor)



А какая модель нам нужна?



Интерпретация коэффициентов

Зачем нужны линейные модели?

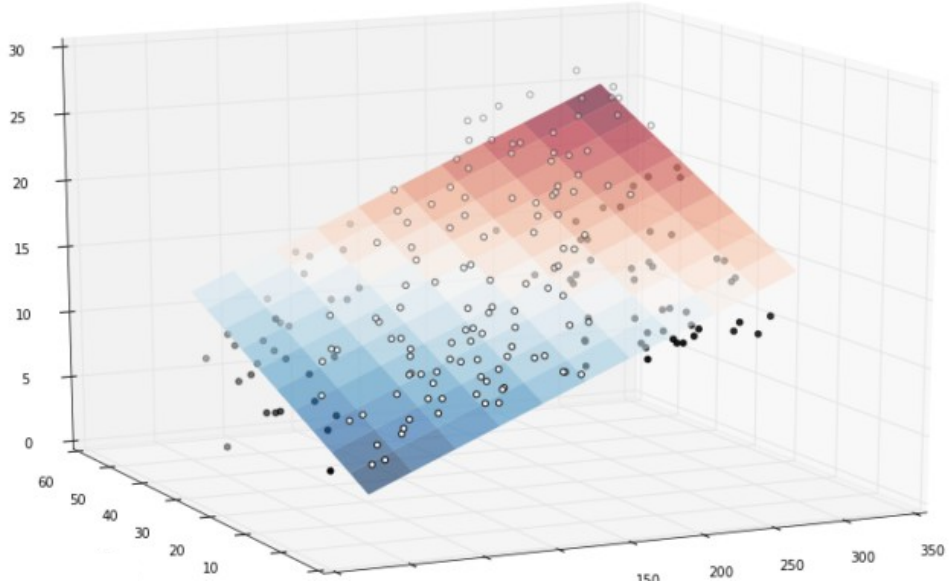
1. Предсказание интересующей нас величины
2. Оценка влияния различных факторов на нашу целевую переменную
3. Линейные модели очень легко использовать и интерпретировать
4. Линейные модели могут восстанавливать даже **нелинейные зависимости**

А если у нас много независимых переменных?

$$y = w_0 + w_1 x + w_2 z + \ldots + w_n t + \epsilon$$

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

$y = f(x)$	$\frac{\mathrm{d}y}{\mathrm{d}x} = f'(x)$
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}	$k\mathrm{e}^{kx}$
$\ln x = \log_{\mathrm{e}} x$	$\frac{1}{x}$
$\sin x$	$\cos x$
$\sin kx$	$k \cos kx$
$\cos x$	$-\sin x$
$\cos kx$	$-k \sin kx$

Производные

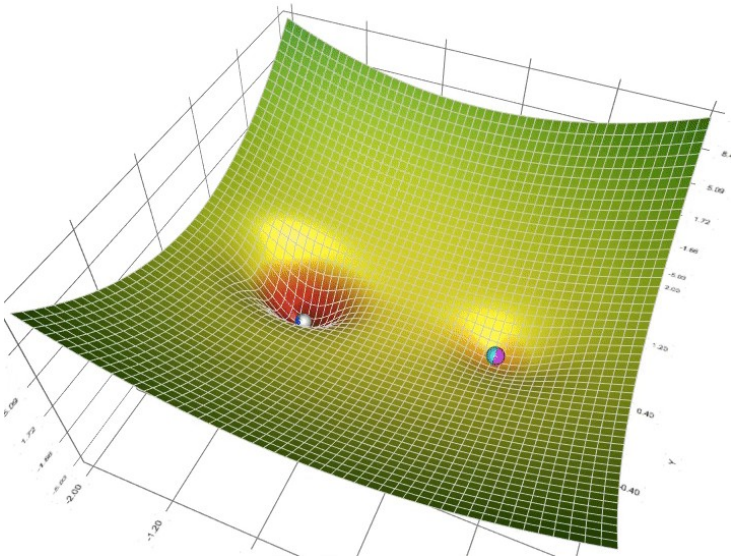
Производные

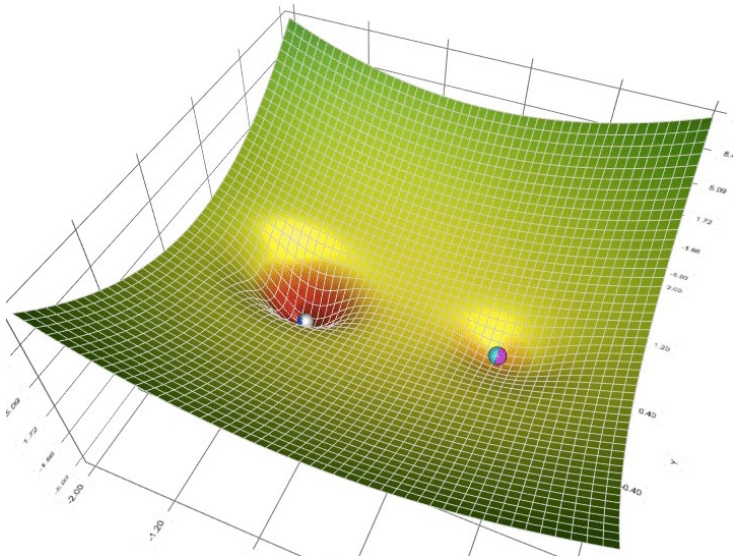
Производные

Как оценивать коэффициенты модели?

Как оценивать коэффициенты модели?

Градиентный спуск





Формулы

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

$$y = Xw$$

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

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

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

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 -= η *
grad(Loss)
```

Отдых -> логистическая регрессия

Связь событий и признаков

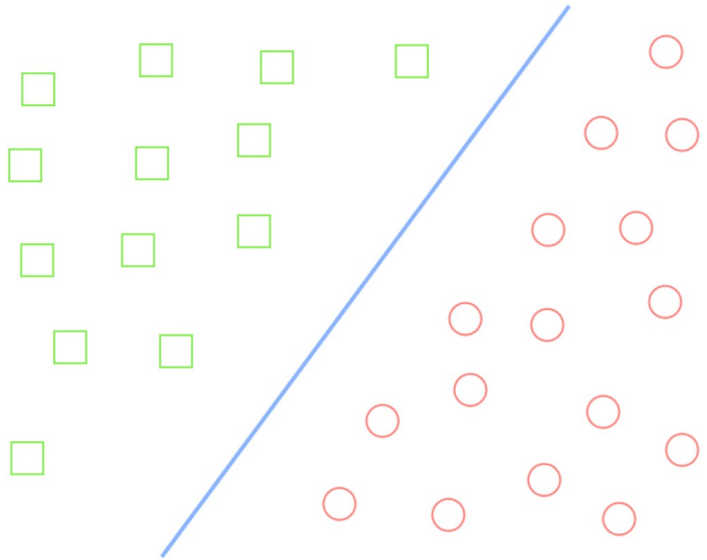
В зависимости от предикторов события могут происходить чаще или реже – логика, совпадающая с логикой связи количественной переменной отклика с набором предикторов.

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



геометрия ML

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 +∞.

Попробуем сами

Логиты

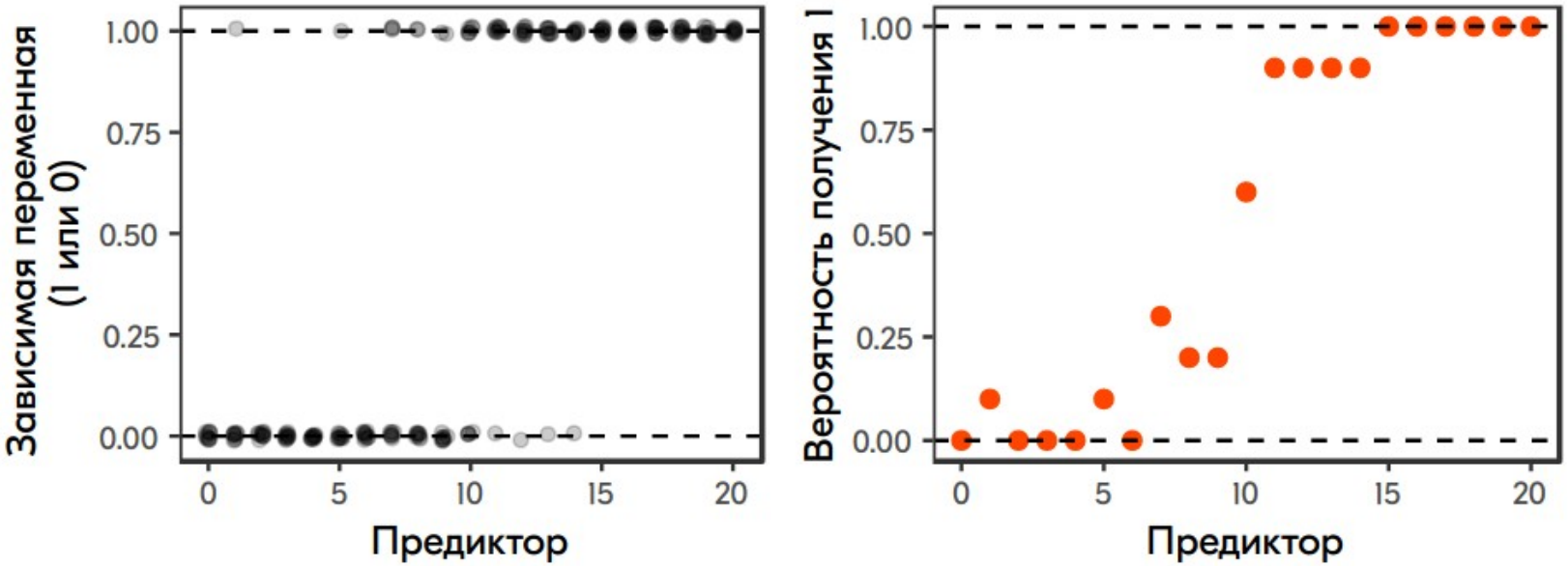
Отношение шансов можно преобразовать в логиты(logit):

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

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

Считаем вероятность

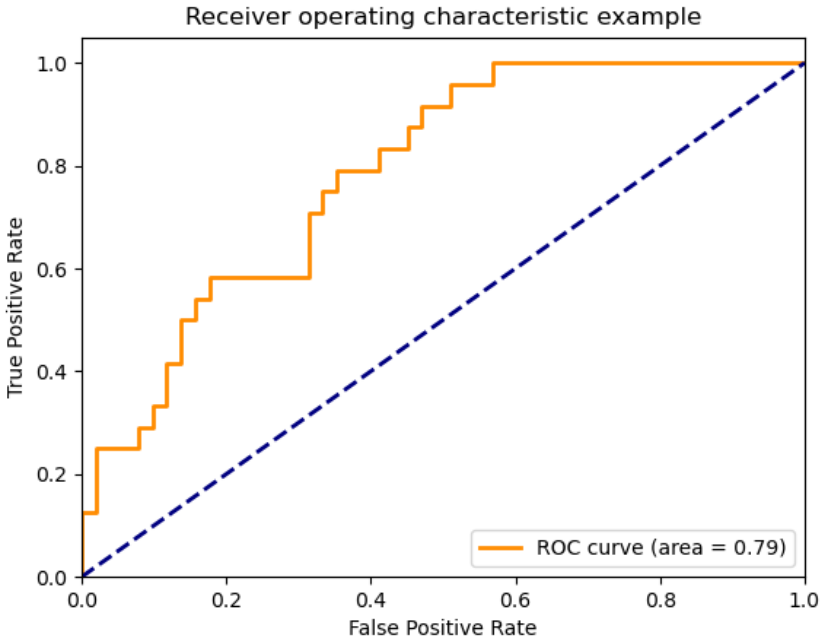
Дискретные значения *vs* вероятности



Как такое учить? BCE Loss

Качество классификации

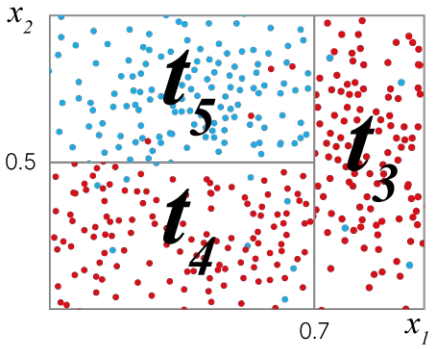
Качество классификации. ROC кривая

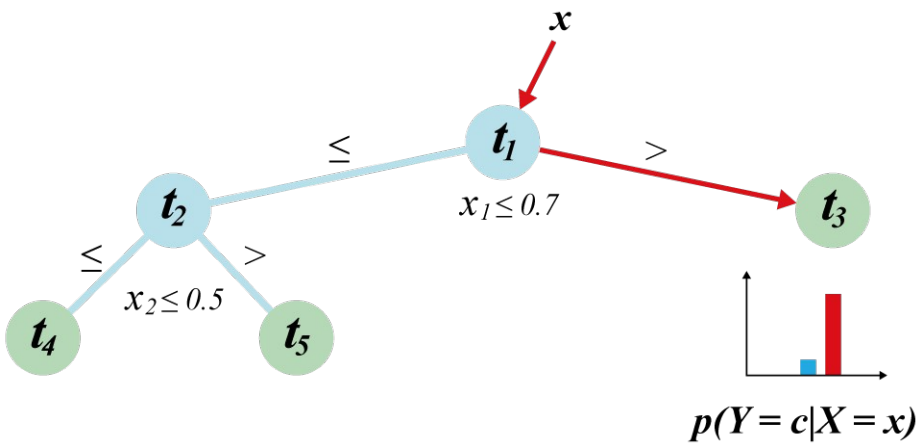


рисуем свою ROC кривую

Построение ROC кривой

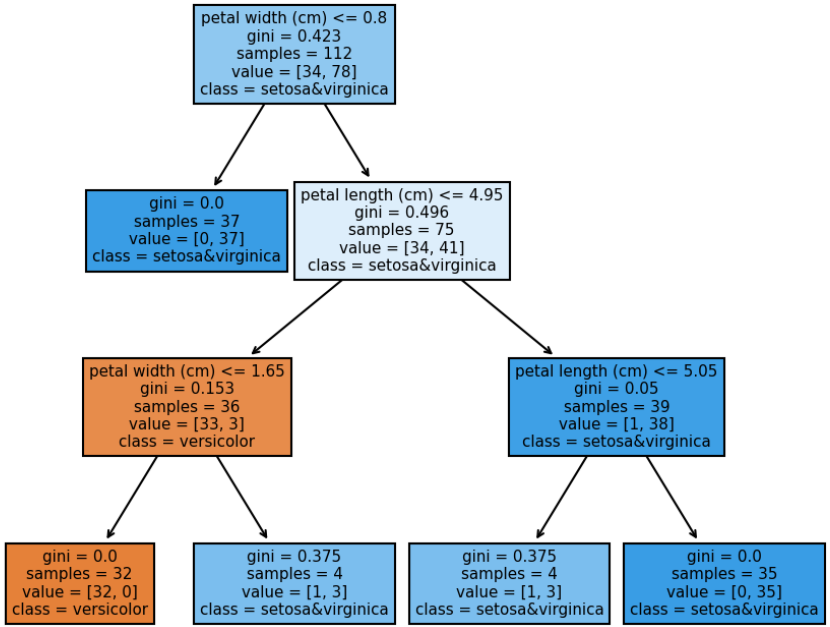
Деревья решений

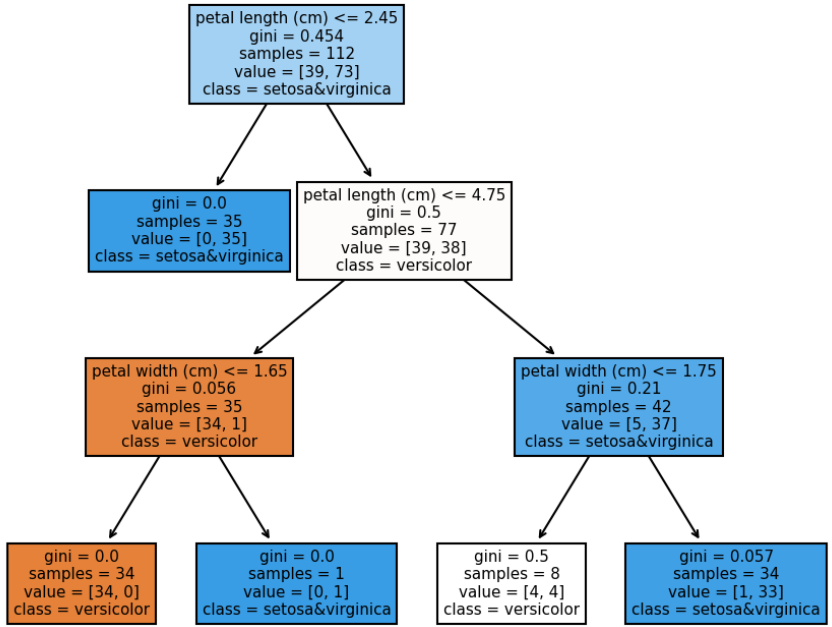




Split mode

Leaf mode


petal width (cm) <= 0.8
gini = 0.423
samples = 112
value = [34, 78]
class = setosa&virginica
gini = 0.0
samples = 37
value = [0, 37]
class = setosa&virginica
petal length (cm) <= 4.95
gini = 0.496
samples = 75
value = [34, 41]
class = setosa&virginica
petal width (cm) <= 1.65
gini = 0.153
samples = 36
value = [33, 3]
class = versicolor
petal length (cm) <= 5.05
gini = 0.05
samples = 39
value = [1, 38]
class = setosa&virginica
gini = 0.0
samples = 32
value = [32, 0]
class = versicolor
gini = 0.375
samples = 4
value = [1, 3]
class = setosa&virginica
gini = 0.375
samples = 4
value = [1, 3]
class = setosa&virginica
gini = 0.0
samples = 35
value = [0, 35]
class = setosa&virginica


petal length (cm) <= 2.45
gini = 0.454
samples = 112
value = [39, 73]
class = setosa&virginica
gini = 0.0
samples = 35
value = [0, 35]
class = setosa&virginica
petal length (cm) <= 4.75
gini = 0.5
samples = 77
value = [39, 38]
class = versicolor
petal width (cm) <= 1.65
gini = 0.056
samples = 35
value = [34, 1]
class = versicolor
petal width (cm) <= 1.75
gini = 0.21
samples = 42
value = [5, 37]
class = setosa&virginica
gini = 0.0
samples = 34
value = [34, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 1]
class = setosa&virginica
gini = 0.5
samples = 8
value = [4, 4]
class = versicolor
gini = 0.057
samples = 34
value = [1, 33]
class = setosa&virginica

Недостатки:

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

Алгоритм

1. s0 = вычисляем энтропию исходного множества
2. Если s0 == 0 значит:
    a. Все объекты исходного набора, принадлежат к одному классу
    b. Сохраняем этот класс в качестве листа дерева
3. Если s0 != 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. 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

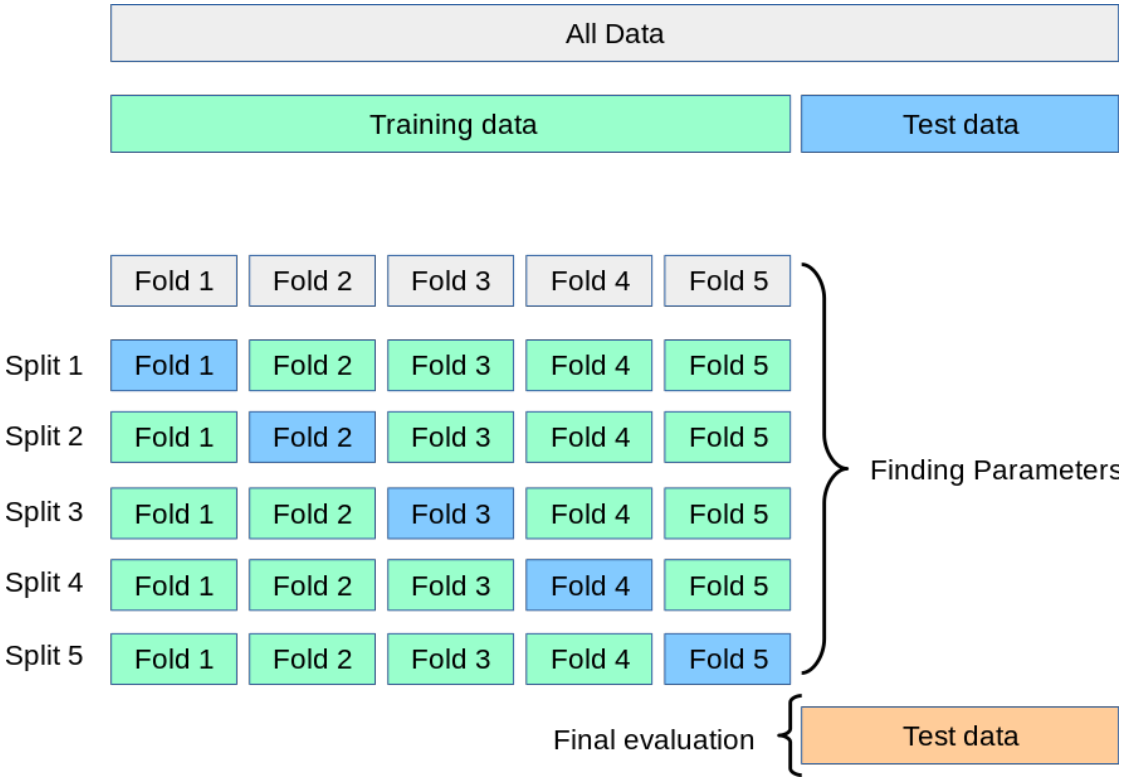
Также для классификации можно в модель передать **веса**

Есть ещё такая штука как **ccp_alpha**

Model Validation

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

K - Fold


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

Отдыхайте